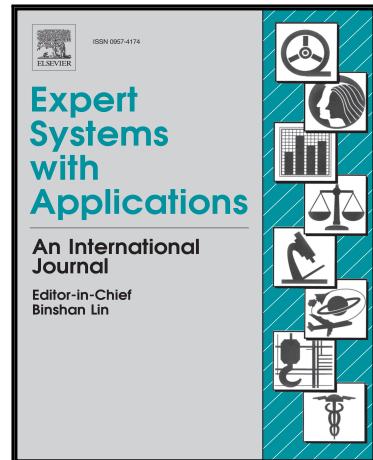


Journal Pre-proof

Geometric Probabilistic Evolutionary Algorithm

Ignacio Segovia-Domínguez, Rafael Herrera-Guzmán,
Juan Pablo Serrano-Rubio, Arturo Hernández-Aguirre

PII: S0957-4174(19)30797-3
DOI: <https://doi.org/10.1016/j.eswa.2019.113080>
Reference: ESWA 113080



To appear in: *Expert Systems With Applications*

Received date: 1 August 2019
Revised date: 18 October 2019
Accepted date: 7 November 2019

Please cite this article as: Ignacio Segovia-Domínguez, Rafael Herrera-Guzmán, Juan Pablo Serrano-Rubio, Arturo Hernández-Aguirre, Geometric Probabilistic Evolutionary Algorithm, *Expert Systems With Applications* (2019), doi: <https://doi.org/10.1016/j.eswa.2019.113080>

This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2019 Published by Elsevier Ltd.

Highlights

- Probabilistic analysis of geometric transformations leads to new search operators
- The nonlinearity of inversions w.r.t. hyperspheres is imitated stochastically
- Probabilistic search mechanisms inherit geometric transformation properties
- Uniformly distributed reflections add exploration capabilities to an EA
- Scientific contributions show competitive performance in benchmark problems

Geometric Probabilistic Evolutionary Algorithm

Ignacio Segovia-Domínguez^{a,*}, Rafael Herrera-Guzmán^a, Juan Pablo Serrano-Rubio^c, Arturo Hernández-Aguirre^b

^aMathematics Department, Center for Research in Mathematics, Guanajuato, México.

^bComputer Science Department, Center for Research in Mathematics, Guanajuato, México.

^cInformation Technologies Laboratory, Technological Institute of Irapuato (ITESI), Guanajuato, México.

Abstract

In this paper we introduce a crossover operator and a mutation operator, called Bernoulli Reflection Search Operator (BRSO) and Cauchy Distributed Inversion Search Operator (CDISO) respectively, in order to define the search mechanism of a new evolutionary algorithm for global continuous optimisation, namely the Geometric Probabilistic Evolutionary Algorithm (GPEA). Both operators have been motivated by geometric transformations, namely inversions with respect to hyperspheres and reflections with respect to a hyperplanes, but are implemented stochastically. The design of the new operators follows statistical analyses of the search mechanisms (Inversion Search Operator (ISO) and Reflection Search Operator (RSO)) of the Spherical Evolutionary Algorithm (SEA). From the statistical analyses, we concluded that the non-linearity of the ISO can be imitated stochastically, avoiding the calculation of several parameters such as the radius of hypersphere and acceptable regions of application. In addition, a new mutation based on a normal distribution is included in CDISO in order to guide the exploration. On the other hand, the BRSO imitates the mutation of individuals using reflections with respect to hyperplanes and complements the CDISO. In order to evaluate the proposed method, we use the benchmark functions of the special session on real-parameter optimisation of the CEC 2013 competition. We compare GPEA against 12 state-of-the-art methods, and present a statistical analysis using the Wilcoxon signed rank and the Friedman tests. According to the numerical experiments, GPEA exhibits a competitive performance against a variety of sophisticated contemporary algorithms, particularly in higher dimensions.

Keywords: Real Parameter Optimisation, Spherical Inversion, Evolutionary Algorithm, Multivariate Distribution

*Corresponding author

Email addresses: ijsegoviad@cimat.mx (Ignacio Segovia-Domínguez), rherrera@cimat.mx (Rafael Herrera-Guzmán), juserrano@itesi.edu.mx (Juan Pablo Serrano-Rubio), artha@cimat.mx (Arturo Hernández-Aguirre)

1. Introduction and state of the art

Evolutionary Algorithms (EAs) are methods for solving optimisation problems inspired by the natural selection process. Recently, these algorithms have become increasingly popular due to their adaptability for real-world applications in areas such as medicine (Smith & Cagnoni, 2011), environmental sciences (Haupt, 2009), music (Miranda & Biles, 2007), food sciences (Lutton et al., 2016), engineering (Kartite & Cherkaoui, 2016), etc.

The evolutionary process of the population in an EA is given by the selection, reproduction and mutation of new individuals through operators which define the search mechanism for the global optimum solution (Back et al., 1997; De Jong, 2016; Yang, 2010). The selection, reproduction and mutation processes are performed iteratively in order to obtain better solutions in each generation. The selection process replaces the individuals by candidate solutions when the candidate solutions have a better fitness value; otherwise, the same individuals are kept in the population. This process helps maintain the quality of the solutions which survive and share their information through the reproduction process to other individuals, with the purpose of exploiting the search space. The mutation process perturbs the position of the solutions in the search space in order to get to unexplored regions where the global optimum may be located (Rothlauf, 2006; Jansen, 2013).

In general, the reproduction and mutation operators define the search mechanism for the global optimum of an EA. There exist very many proposals of different approaches to implementing new reproduction and mutation operators. These approaches include linear combinations of vectors (PSO) (Mavrovouniotis et al., 2017; Kennedy & Eberhart, 1995), affine transformations (DE) (Li et al., 2018), non linear transformations (SEA) (Serrano-Rubio et al., 2018), sampling of points using probabilistic models (EDAS, CMA-ES) (Larrañaga & Lozano, 2001; Hansen & Ostermeier, 1996; Serrano-Rubio et al., 2014, 2013). Our main interest, however, lies in the study and design of evolutionary algorithms from a geometric viewpoint, whose geometric parameters are fully and thoroughly examined in order to define distribution functions to be used in efficient exploration and exploitation mechanisms.

During last two decades various authors have proposed leading approaches to design and analyse evolutionary algorithms from a geometric perspective in order to describe the reproduction and mutation operators as well as to generalise search algorithms for continuous and discrete optimisation problems (Moraglio & Johnson, 2010; Moraglio et al., 2007, 2006). These approaches have encouraged the development of new algorithms and the improvement of search mechanisms of existing algorithms, e.g. Moraglio (2007); Moraglio & Poli (2004); Serrano-Rubio (2015).

The great majority of search operators have been implemented using spatial-based operations and some of them have been designed within various geometric frameworks. For instance, the search operators of the Particle Swarm Optimisation (PSO) produce new individuals by performing linear combinations of vectors/particles of the swarm Kennedy & Eberhart (1995). The search oper-

50 ators of the Differential Evolution (DE) algorithm create new individuals via addition of a mutation vector to a third vector Price (2013). The operators of PSO and DE are affine geometric transformations. On the other hand, the design and analysis of search strategies for EAs can be carried out by using geometric properties of the space itself. From this perspective, the search space has been described as a spatial structure on which geometric operations are carried out taking into account the distance between points (e.g. candidate solutions in the search space, parameters, vector fields, etc.).

55 In this context, a prominent geometric framework, introduced by Moraglio (2007), describes the dynamism of search operators by using only geometric terms, arguing that this geometric framework allows the generalisation of evolutionary search mechanisms (see Moraglio & Togelius (2009b), Moraglio & Togelius (2009a)), and using geometric shapes to delimit regions of the search space where to sample offspring solutions with respect to the parents' positions.

60 Another leading geometric framework, in the EC community, is based on the Information Geometry approach, which uses the Riemannian structure of the parameter space of a statistical model to update a search distribution and to increase the probability of sampling in a promising search region, see Ollivier et al. (2017), Segovia-Dominguez & Hernandez-Aguirre (2015) and Amari (2016).

65 Similarly, Serrano-Rubio et al. (2014) introduced a new EA whose operators were designed taking advantage of the Geometric Algebra (GA) and the Conformal Geometric Algebra (CGA) frameworks, which facilitate modelling and manipulating geometric objects (such as vectors, n-spheres and multi-vectors) and transformations via compact algebraic representations using the geometric product, see Dorst et al. (2010). The Spherical Evolutionary Algorithm (SEA) and its operators, The Inversion Search Operator (ISO) and the Reflection Search Operator (RSO) (Serrano-Rubio et al., 2018). The ISO performs inversions with respect to a hypersphere in order to explore and exploit the search space in a non-linear and non-affine fashion, while the RSO mutates an individual using Euclidean reflections to redistribute the population on the surface of hypersphere. The ISO and RSO operators were constructed as a result of the analysis of EA search mechanisms via Geometric Algebra with the reproduction and mutation of particles defined via algebraic operations on geometric objects, Serrano-Rubio et al. (2013, 2018).

70 80 This paper aims at continuing the research area of search operator design inspired by Geometric Algebra by adding a novel approach of analysing operators from a probabilistic/statistical point of view. The introduction of appropriate random variables and density functions into the geometric framework allows us to maintain the non-linear properties of the original geometric framework.

85 From this study we determined various improvements could be carried out for the ISO and RSO operators. Our statistical analysis led us to consider the use of Cauchy, Normal and Bernoulli distributions in the design of the new operators. We consider that the idea of using statistical analyses of operators to design new ones is one of our main contributions.

90 Thus, our study shows that compact and competitive EAs can be built from GA-operators and implemented stochastically, i.e. we adopt a geometric per-

spective combined with a statistical approach for introducing two new operators: the Cauchy Distributed Inversion Search Operator (CDISO) and the Bernoulli Reflection Search Operator (BRSO). The resulting search mechanism aims to
 95 keep diversity and avoid premature convergence, speeds up the search and reduces the computational time. The CDISO and the BRSO operate jointly in the evolutionary algorithm called Geometric Probabilistic Evolutionary Algorithm (GPEA), exploring and exploiting the search space using few parameters from a Cauchy distribution, which guides the reproduction and mutation of the
 100 individuals efficiently.

The CDISO, in analogy to the ISO, provides an aggressive search mechanism in the exploration and exploitation stages of the evolutionary algorithm. The combination of CDISO and BRSO is effective in avoiding stagnation of the population since they redistribute the individuals near the best individuals,
 105 and ensuring diversity by using a mutation in the best regions found by the algorithm. Our definition of the CDISO replaces the calculation of two parameters used in the ISO (the radius of hypersphere and the applicability region for the inversion) by parameters included in one random variable obtained from the Cauchy distribution. Thus, our proposal takes advantage of the geometric insights of SEA, its implementation is shorter, and turns out to be more competitive.
 110

Unlike highly sophisticated contemporary algorithms, the implementation of GPEA requires no extra effort to control its parameters. The GPEA is (statistically) compared with the SEA and other state of the art algorithms
 115 on the 28 real parameter optimisation problems from the CEC2013 benchmark set. From the statistical comparisons, we conclude that the operators of GPEA explore and exploit the search space efficiently.

The paper is organised as follows. In Section 2, we introduce notation on inversion and reflection maps, and recall the description of the SEA. In Section 3,
 120 we present the statistical analysis of the geometric operators ISO and RSO, and determine the relevant probability distributions. The details of the proposed GPEA are given in Section 4. In Section 5, we present statistical comparisons of the performance of the GPEA versus a variety of Evolutionary Algorithms using the benchmark functions of the CEC 2013 competition on real parameter
 125 optimisation. Finally, in Section 6 we summarise our results and identify various paths for future work.

2. Preliminaries

In this section we explain the geometric transformations included in the search mechanism of the SEA. We also present a statistical analysis of the
 130 overall behaviour of the ISO and the RSO within the SEA.

2.1. Inversion with respect to a hypersphere

An inversion with respect to a hypersphere is a non-linear transformation which maps a point inside the hypersphere to a point outside the hypersphere

and vice versa (Coxeter, 1971). Let ϑ be a hypersphere in the d -dimensional Euclidean space with centre \vec{c} and radius r . The inversion of a point \vec{x} with respect to the hypersphere is defined by

$$Inv_{\vec{c}, r}(\vec{x}) = \frac{r^2}{||\vec{x} - \vec{c}||^2}(\vec{x} - \vec{c}) + \vec{c}, \quad (1)$$

Figure 1 shows the effect of inverting various objects with respect to a circle, some of which clearly exhibit its non-linear nature (lines are mapped to curves and vice versa).

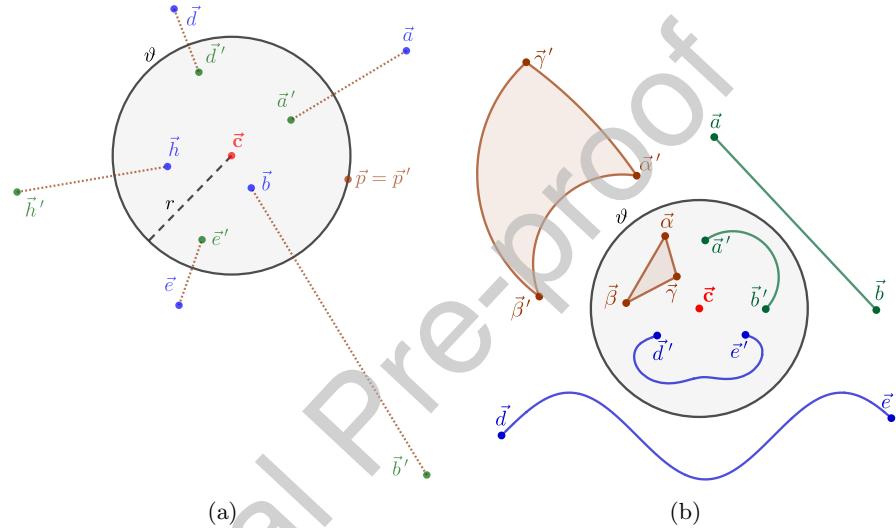


Figure 1: Inversion with respect to a circle. (a) Figure shows the inversion of six points with respect to the circle ϑ with centre \vec{c} and radius r . (b) Inversions of three geometric spaces with respect to the circle ϑ which display the non-linear nature of the transformation.

¹³⁵ 2.2. Reflection with respect to a hyperplane

The reflection of a point \vec{x} with respect to a hyperplane with equation

$$\vec{x} \cdot \vec{a} = \alpha$$

in d -dimensional space, where \vec{a} is a vector normal to the hyperplane and $\alpha \in \mathbb{R}$, is given by

$$\vec{x} \mapsto \vec{x} - 2 \frac{\vec{x} \cdot \vec{a} - \alpha}{\vec{a} \cdot \vec{a}} \vec{a}, \quad (2)$$

where “.” denotes the usual dot product. We shall, in fact, only use reflections with respect to canonical planes translated to the point \vec{c} (see Figure 2). More precisely, for $\vec{x} \in \mathbb{R}^d$, $\vec{\lambda} \in \{1, -1\}^{d \times d}$,

$$Ref_{\vec{c}, \vec{\lambda}}(\vec{x}) = (\lambda_1(x_1 - c_1), \dots, \lambda_d(x_d - c_d)) + (c_1, \dots, c_d). \quad (3)$$

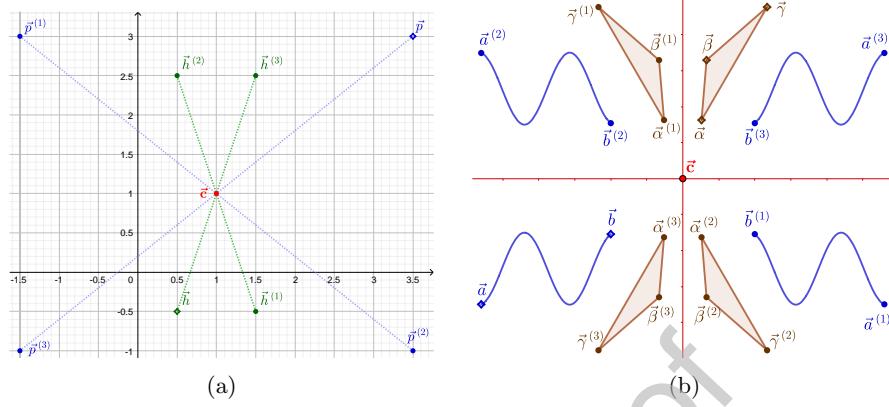


Figure 2: Reflections of vectors and geometric shapes with respect to the canonical basis subspaces.

2.3. Evolutionary algorithm based on Spherical Inversions (SEA)

The Evolutionary Algorithm based on Spherical Inversions (SEA) includes a non linear geometric transformation in the design of its operators. This approach differs from the geometric perspective of Differential Evolution (DE) and Particle Swarm Optimisation (PSO) whose geometric operations are affine. The reproduction mechanisms of the SEA involve inversions with respect to hyper-spheres and reflections with respect to hyperplanes (Serrano-Rubio et al., 2018), whose use was first motivated by Geometric Algebra (Serrano-Rubio et al., 2014).

The search mechanism of the SEA is based on two geometric search operators: the Reflection Search Operator (RSO) and the Inversion Search Operator (ISO). The pseudo-code for SEA is shown in Algorithm 1, in which each vector $\vec{x}^{(i)}$ is an individual of the population $\mathcal{P}^{(t)}$. The goal of RSO is to redistribute the individuals on the surface of a hyper-sphere by using *reflections*, lines 18-25. The ISO produces a new point by applying an *inversion map* with respect to a hypersphere, line 16. SEA performs either the ISO or the RSO with probability 0.5. These operators need an individual as a centre of a hyper-sphere, which is randomly chosen from the set η of the best individuals in the population. The geometric search operator produces a candidate individual $\vec{y}^{(i)}$. The current individual $\vec{x}^{(i)}$ is replaced by $\vec{y}^{(i)}$ when $\vec{y}^{(i)}$ has a better fitness value than $\vec{x}^{(i)}$.

Let us consider a hypersphere ϑ centred on one of the best individuals of $\mathcal{P}^{(t)}$ and the random radius r . According to the RSO, every component of $\vec{x}^{(i)}$ has a chance of being reflected w.r.t the centre c and the canonical hyperplanes, see Fig. 3a. On the other hand, according to the ISO, an individual $\vec{x}^{(i)}$ is either mapped inside ϑ whenever $\vec{x}^{(i)}$ is outside the hypersphere, or mapped outside ϑ whenever $\vec{x}^{(i)}$ is inside the hypersphere, see Fig. 3b.

The ISO and the RSO are search operators inspired by the inversion and reflection maps, respectively. Their effectiveness has been tested empirically in

Algorithm 1 The SEA algorithm

```

1:  $t \leftarrow 0, N \leftarrow 129, \eta \leftarrow 9$ 
2:  $\mathcal{P}^{(t)} \leftarrow \mathcal{U}(\text{Domain}),$  where  $\vec{x}^{(i)} \in \mathcal{P}^{(t)}$ 
3: while (Stop condition is not reached) do
4:    $\mathcal{P}_{best}^{(t)} \leftarrow$  The best  $\eta$  individuals among  $\mathcal{P}^{(t)}$ 
5:    $\mathcal{P}^{(t+1)} \leftarrow \emptyset$ 
6:   for all individuals  $\vec{x}^{(i)} \in \mathcal{P}^{(t)}$  do
7:      $\vec{c} \leftarrow$  randomly chosen from  $\mathcal{P}_{best}^{(t)}$ , such that  $\vec{x}^{(i)} \neq \vec{c}$ 
8:      $\kappa, u, v \leftarrow$  three observed values of  $U \sim \mathcal{U}(0, 1)$ 
9:     if  $\kappa \geq 0.5$  then                                ▷ ISO (start)
10:      if  $v \geq 0.5$  then                         ▷ Mutation
11:         $b \leftarrow$  an observed value of  $B \sim \mathcal{N}(0, 1)$ 
12:         $j$  is randomly taken from  $\{1, \dots, d\}$ 
13:         $c_j \leftarrow c_j + b$ 
14:      end if
15:       $r \leftarrow \sqrt{2u\|\vec{x} - \vec{c}\|^2}$ 
16:       $\vec{y}^{(i)} = T_{inv}(\vec{x}^{(i)}, \vec{c}, r)$           ▷ Inversion map
17:    else                                         ▷ ISO (end)
18:      for all  $k \in \{1, 2, \dots, d\}$  do           ▷ RSO (start)
19:         $u_k \leftarrow$  an observed value of  $U \sim \mathcal{U}(0, 1)$ 
20:        if  $u_k > 0.5$  then                      ▷ Reflection map
21:           $y_k^{(i)} \leftarrow -(x_k^{(i)} - c_k) + c_k$ 
22:        else
23:           $y_k^{(i)} \leftarrow (x_k^{(i)} - c_k) + c_k$ 
24:        end if
25:      end for                                     ▷ RSO (end)
26:    end if
27:     $\vec{y}^{(i)} \leftarrow \text{Reinsertion}(\vec{y}^{(i)})$ 
28:    if  $\mathcal{F}(\vec{y}^{(i)}) < \mathcal{F}(\vec{x}^{(i)})$  then
29:       $\mathcal{P}^{(t+1)} \leftarrow \mathcal{P}^{(t+1)} \cup \vec{y}^{(i)}$ 
30:    else
31:       $\mathcal{P}^{(t+1)} \leftarrow \mathcal{P}^{(t+1)} \cup \vec{x}^{(i)}$ 
32:    end if
33:  end for
34:   $t \leftarrow t + 1$ 
35: end while
36: Return the elite individual in  $\mathcal{P}^{(t)}$ 

```

a variety of classical optimisation problems (Serrano-Rubio et al., 2018, 2014).

3. Statistical Analysis of ISO and RSO

In this section, we examine statistically the performance and effectiveness of the ISO and RSO operators in order to improve them based on a thorough

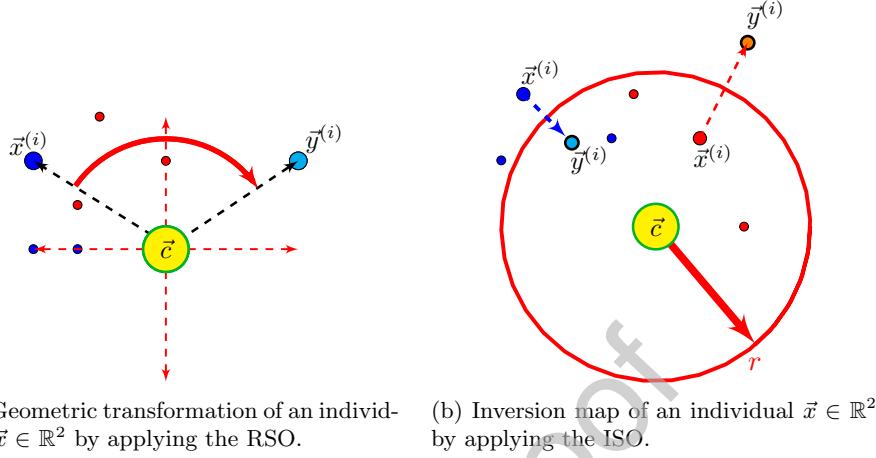


Figure 3: Examples in which geometric operators RSO and ISO are applied to produce new candidate solutions.

170 understanding of their stochastic parameters. We believe this approach will lead to further enhancements and proposals.

3.1. Statistical Analysis of ISO

175 The ISO inverts points with respect to a hypersphere and it has the capability of producing new individuals in locations not explored previously. It is implemented in lines 9 - 17 of Algorithm 1. The purpose of the ISO is to map points to new regions in a non-linear fashion. Our statistical study focuses on two important steps implemented in the ISO and whose stochastic parameters are decisive in its working: a) the calculation of the inverse individual (points) and b) the mutation of the centre of the hypersphere.

180 3.1.1. The inverse individual

The radius of the inversion hypersphere is an important stochastic parameter which determines the location of the new individual. For the sake of the reader, we will carry out the calculations in great detail. The radius of the hypersphere is determined by the estimation of its center and the Euclidean distance between the center and the current individual, as well as a random variable u to be sampled from an uniform distribution $U \sim \mathcal{U}(0, 1)$, i.e.

$$r = \sqrt{2u\|\vec{x} - \vec{c}\|}. \quad (4)$$

Let us substitute this expression in 1

$$\begin{aligned}
 \vec{y} &= \frac{r^2}{\|\vec{x} - \vec{c}\|} (\vec{x} - \vec{c}) + \vec{c} \\
 &= \frac{(\sqrt{2u}\|\vec{x} - \vec{c}\|)^2}{\|\vec{x} - \vec{c}\|} (\vec{x} - \vec{c}) + \vec{c} \\
 &= 2u(\vec{x} - \vec{c}) + \vec{c}
 \end{aligned} \tag{5}$$

where \vec{x} is the current individual and \vec{c} is the centre of hypersphere. This equation has been implemented in line 15 of the pseudo-code of SEA. In general, the inverse point \vec{y} lies on the line passing through \vec{c} with direction $(\vec{x} - \vec{c})$, and it is equally likely to be any point over the segment between \vec{c} and $\vec{c} + 2(\vec{x} - \vec{c})$.

In order to obtain a probability distribution, let us assume that the Euclidean distance between the current individual and the centre of the hypersphere is constant, so that we rewrite the radius (4) using a random variable r_u

$$\begin{aligned}
 r &= \sqrt{2u}\|\vec{x} - \vec{c}\|^2 \\
 &= \sqrt{2u}\|\vec{x} - \vec{c}\| \\
 &= r_u\|\vec{x} - \vec{c}\|,
 \end{aligned} \tag{6}$$

where the coefficient $2u$ has been written in terms of the random variable r_u . The random variable r_u represents the probability of getting a radius below or above of the Euclidean distance between the current individual and the centre of hypersphere. According Eq. 6 when $r_u < 1$ the inverse operations produce a point inside the hypersphere, on the contrary, when $r_u > 1$ the inverse operation produce a point outside the hypersphere. Therefore the value of $r_u < 1$ is crucial since determines the position of the new point. Equations (7) and (8) present the distribution function and density function associated to the random variable r_u respectively, i.e.

$$\begin{aligned}
 F(r_u) &= P(R_u \leq r_u) \\
 &= P(g(U) \leq r_u) \\
 &= P(U \leq g^{-1}(r_u)) \\
 &= P\left(U < \frac{r_u^2}{2}\right) \\
 &= \int_0^{\frac{r_u^2}{2}} du \\
 &= \frac{r_u^2}{2},
 \end{aligned} \tag{7}$$

$$\begin{aligned}
 f(r_u) &= \frac{d}{dr_u} \frac{r_u^2}{2} \\
 &= r_u,
 \end{aligned} \tag{8}$$

185 so that $f(r_u)$ is defined by $\sqrt{2u}$ in the interval $0 \leq r_u \leq \sqrt{2}$. According to the plots, the minimum and maximum value which can be assigned to the radius of hypersphere is 0 and $\sqrt{2}$ since r_u follows a triangular density function $f(r_u) = r_u \mathbb{1}_{[0,\sqrt{2}]}$.

190 Figures 4a and 4b show the triangular density functions associated to the random variable r_u and the radius r of the hypersphere respectively.

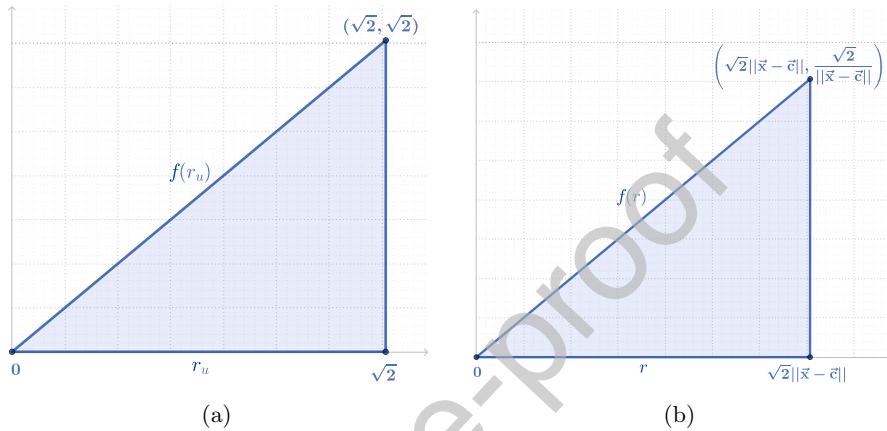


Figure 4: Triangular density functions. (a) r_u follows a triangular density function $f(r_u)$. (b) r follows a triangular density function $f(r)$.

Our statistical analysis is based on the calculation of the probability to obtain a value lower than 1 for r_u since it may be useful to determine the number of times that an inverse individual is produced outside and inside the hypersphere. We consider a domain for r_u between $2 - \sqrt{2}$ and $\sqrt{2}$.

According to our analysis, there is a probability value of 0.828427 to obtain a radius value close to $|\vec{x} - \vec{c}|$,

$$\begin{aligned} \int_{2-\sqrt{2}}^{\sqrt{2}} r_u dr_u &= \frac{r_u^2}{2} \Big|_{2-\sqrt{2}}^{\sqrt{2}} \\ &= 2\sqrt{2} - 2 \\ &\approx 0.828427. \end{aligned} \tag{9}$$

195 On the other hand, there is low probability 0.171573 of getting a radius close to zero, i.e. of getting a radius value below the 30% of the maximum length. It is worth noticing that the final inverse position \vec{y} depends on a adjustment, via square, of the current radius sampling scheme, see Eq. (5).

200 From previous observations, we conclude that the inversion map adapts the radius distribution in order to produce a uniformly distributed inverse position \vec{y} over the segment between \vec{c} and $\vec{c} + 2(\vec{x} - \vec{c})$.

3.1.2. Mutation of the center

The mutation process is implemented adding a normal distributed random number to a randomly selected component of the Euclidean vector \vec{c} . It is described in the lines 10 - 14 of Algorithm 1, and is computed as the sum of \vec{c} and a vector with only one non-zero entry given by a random number $b \sim \mathcal{N}(0, 1)$:

$$\begin{aligned}\vec{c}_m &= [c_1, \dots, c_{j-1}, c_j + b, c_{j+1}, \dots, c_d]^T \\ &= [c_1, \dots, c_{j-1}, c_j, c_{j+1}, \dots, c_d]^T + [0, \dots, 0, z_j = b, 0, \dots, 0]^T \\ &= \vec{c} + \vec{z}\end{aligned}\quad (10)$$

where the component j is uniformly chosen. Therefore, a new individual produced by ISO is computed as follows:

$$\begin{aligned}\vec{y}_m &= 2u(\vec{x} - \vec{c}_m) + \vec{c}_m \\ &= 2u(\vec{x} - \vec{c}) + \vec{c} + (1 - 2u)\vec{z} \\ &= \vec{y} + (1 - 2u)\vec{z} \\ &= \vec{y} + \beta.\end{aligned}\quad (11)$$

Now, we will analyse the density function for the random variable β .

Figure 5 shows a numerical approximation of $f(\beta)$ using the product of two independent random variables $(1 - 2u)\vec{z}$. The probability mass function of $f(\beta)$ in the interval $[-1, 1]$ is approximately equal to 0.906. Furthermore, the density function decays rapidly and therefore β is more likely to take small random values in (11). Additionally, notice that there is no variable in (11) controlling the size of the random mutation. As a consequence, a bounded random perturbation is performed regardless of the population variance. However, this is a desirable feature.

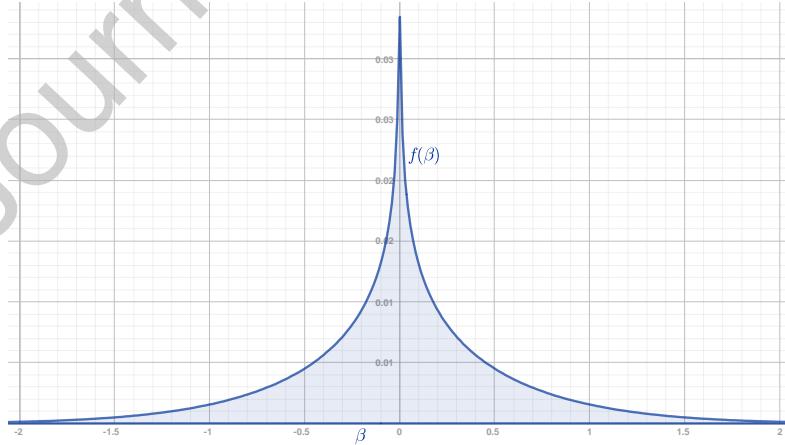


Figure 5: Numerical approximation of the density function $f(\beta)$, one-component mutation on $\vec{y}^{(i)}$.

The mutation strategy of ISO perturbs the position of individuals by using the product between two random variables which are obtained from an uniform distribution and a standard normal distribution. According to our previous analysis, the mutation represents a change of position in only one canonical direction.

3.2. Statistical Analysis of RSO

Randomly distributed reflection maps can be generated using a simulation of random variables as in (Serrano-Rubio et al., 2014). Recall that for $\vec{x} \in \mathbb{R}^d$, $\lambda \in \{1, -1\}^{\times d}$,

$$Ref_{\vec{c}, \vec{\lambda}}(\vec{x}) = (\lambda_1(x_1 - c_1), \dots, \lambda_d(x_d - c_d)) + (c_1, \dots, c_d), \quad (12)$$

so that for $z = T_{ref}(\vec{x}, \vec{c}, \vec{\lambda})$, $z_i = x_i$ when $\lambda_i = 1$ and $z_i = 2c_i - x_i$ when $\lambda_i = -1$. Thus, the computational implementation of this transformation in lines 18–25 of Algorithm 1 depends on a Bernoulli distributed random variable, i.e.

$$z_i = (1 - 2s_i)(x_i - c_i) + c_i, \quad (13)$$

where s_i is an observed value of $S \sim Ber(p)$ with $p = 0.5$. It is worth noting that the random variable $1 - 2S$ follows a Rademacher distribution, which is widely used in several research areas and engineering (Montgomery-Smith, 1990)(Hitczenko & Kwapień, 1994).

Although there is an apparent fairness in the RSO, we are going to show that there is a bias. When all outcomes from S are equal to 0, $\vec{z} = \vec{x}$, which is not desirable because as we wish the population to evolve. On the other hand, the number of entries of \vec{x} which are modified, equals the number of positive outcomes form sampling a Bernoulli variable. Thus, the probability of getting a given number of differences between \vec{z} and \vec{x} is modelled by a Binomial distribution. In other words,

$$\sum_{i=0}^d S_i \sim Bin(d, p), \quad (14)$$

where S_1, \dots, S_d are i.i.d. random variables, all Bernoulli trials with success probability $p = 0.5$ (Hayslett, 1967).

The number of entries that get reflected is statistically modelled by a sequence of d independent Bernoulli experiments and their sum as a Binomial distribution with probability value $p = 0.5$ (see the graph of corresponding density function $Bin(d, 0.5)$ for $d = 20$ in Fig. 6a). Note that most of the probability mass is located around the centre of the domain $[0, d]$ and, as a consequence, there is very small probability of obtaining values either close to 0 or d . A similar behaviour can be observed regardless of the value of d (see Fig. 6b), which produces partial exploration of the search space.

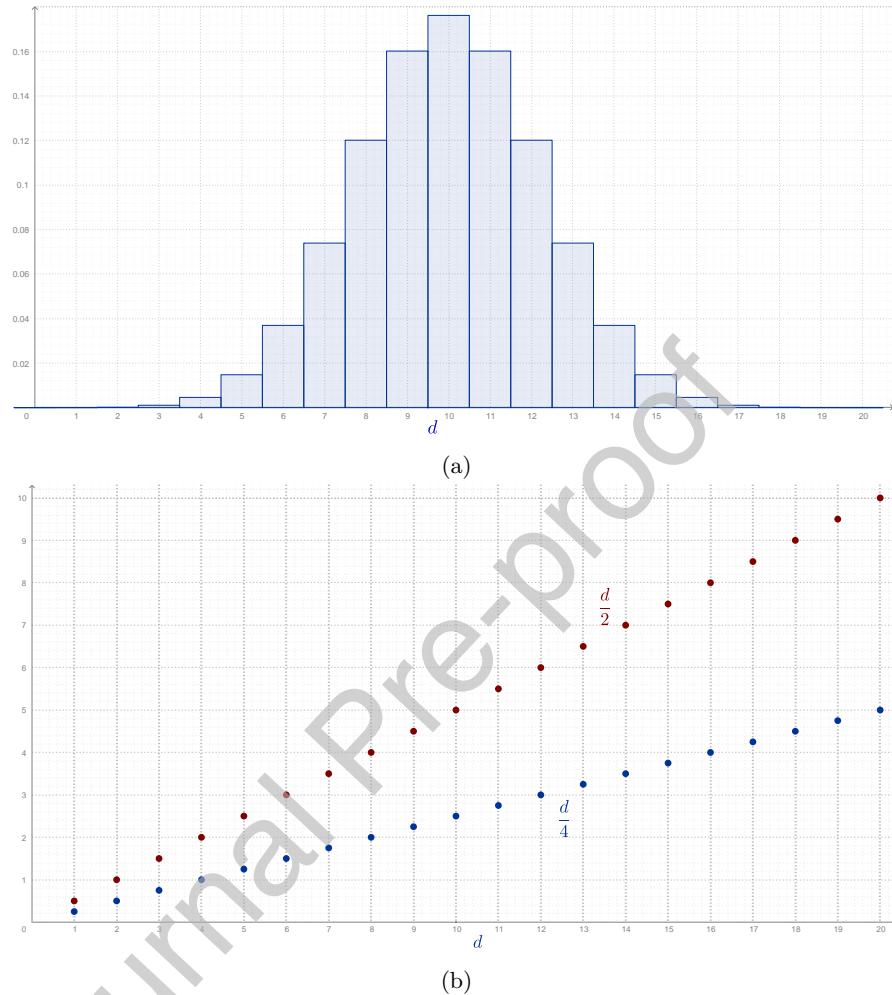


Figure 6: Number of entries that are taken from \vec{z} in the Reflection Search Operator (Serrano-Rubio et al., 2018). (a) Binomial density function $Bin(d, 0.5)$ when $d = 20$ (b) Mean and variance of $Bin(d, 0.5)$, $\frac{d}{2}$ and $\frac{d}{4}$ respectively.

In order to tackle the main issues studied above, in next section we introduce a modification capable of creating evenly distributed reflections of any particle in the search space.

²³⁵ 3.3. New Search Operators CDISO and BRSO

3.3.1. Cauchy Distributed Inversion Search Operator (CDISO)

Recall that the inversion of a point \vec{x} with respect to the hypersphere with center \vec{c} and radius r is defined by

$$Inv_{\vec{c},r}(\vec{x}) = \frac{r^2}{\|\vec{x} - \vec{c}\|^2}(\vec{x} - \vec{c}) + \vec{c},$$

The scalar value

$$\tau = \frac{r^2}{\|\vec{x} - \vec{c}\|^2} \quad (15)$$

controls the distance between \vec{c} and the point produced by the inversion map. Hence, the importance of modelling the factor τ .

In order to be able to draw some conclusions, let us assume that the current population is distributed according to the multivariate normal density $\mathcal{N}(\vec{x}; \vec{\mu}, \Sigma)$, and that the set of best individuals follows a multivariate normal density $\mathcal{N}(\vec{c}; \vec{\mu}_c, \Sigma_c)$, for some given parameters $\vec{\mu}, \Sigma, \vec{\mu}_c, \Sigma_c$. Replace \vec{x} by an individual of the population and \vec{c} by an individual from the set of best individuals in (15) by random variables, i.e.

$$T = \frac{r^2}{\|\mathbf{X} - \mathbf{C}\|^2}, \quad (16)$$

where $\mathbf{X} \sim \mathcal{N}(\vec{x}; \vec{\mu}, \Sigma)$ and $\mathbf{C} \sim \mathcal{N}(\vec{c}; \vec{\mu}_c, \Sigma_c)$ are two random vectors, and T is the random variable for the scalar factor τ . In order to find the distribution law for T , we need to calculate the density function of $\|\mathbf{X} - \mathbf{C}\|^2$. Since \mathbf{X} and \mathbf{C} follow two different multivariate density functions, the subtraction of both variables follows a multivariate normal density $\mathcal{N}(\vec{x}_\xi; \vec{\mu}_\xi, \Sigma_\xi)$, which can be rewritten as follows

$$\mathbf{X} - \mathbf{C} = \vec{\mu}_\xi + L_\xi \mathbf{Z}, \quad (17)$$

where $\vec{x}_\xi = \vec{x} - \vec{c}$; $\vec{\mu}_\xi = \vec{\mu} - \vec{\mu}_c$, $L_\xi L_\xi^\top = \Sigma_\xi = \Sigma + \Sigma_c$, and \mathbf{Z} is a random vector from a multivariate standard normal density such that $\mathbf{Z} \sim \mathcal{N}(\vec{0}; \mathbf{I})$. Therefore,

$$\begin{aligned} \|\mathbf{X} - \mathbf{C}\|^2 &= (\vec{\mu}_\xi + L_\xi \mathbf{Z})^\top (\vec{\mu}_\xi + L_\xi \mathbf{Z}) \\ &= \vec{\mu}_\xi^\top \vec{\mu}_\xi + 2\vec{\mu}_\xi^\top L_\xi \mathbf{Z} + \mathbf{Z}^\top L_\xi^\top L_\xi \mathbf{Z}, \end{aligned} \quad (18)$$

where each term follows different distributions. The first term produces a constant value. The second term is a sum of d random variables distributed as normal densities with different variances, so that it follows an univariate normal distribution (Patel & Read, 1996). The last term represents the norm $\|L_\xi \mathbf{Z}\|^2$, which is distributed as a sum of d independent gamma variables with different parameters (Moschopoulos, 1985)(Moschopoulos & Canada, 1984). In summary, although there is a closed form for each term, the summation produces a composite distribution from which straightforward calculations are rather cumbersome.

In order to circumvent the difficulties found in (18), we are going to simplify our model by forcing $\|\mathbf{X} - \mathbf{C}\|^2 \sim \mathcal{N}(\vec{x}; \vec{\mu}_\gamma, \Sigma_\gamma)$. This means

$$\begin{aligned}\vec{\mu}_\gamma &= \mathbb{E} [\|\mathbf{X} - \mathbf{C}\|^2] \\ &= \mathbb{E} [\vec{\mu}_\xi^\top \mu_\xi] + 2\vec{\mu}_\xi^\top L_\xi \mathbb{E} [\mathbf{Z}] + \mathbb{E} [\mathbf{Z}^\top L_\xi^\top L_\xi \mathbf{Z}] \\ &= \vec{\mu}_\xi^\top \mu_\xi + \text{Tr} [L_\xi^\top L_\xi],\end{aligned}\quad (19)$$

and

$$\begin{aligned}\Sigma_\gamma &= \text{Var} [\|\mathbf{X} - \mathbf{C}\|^2] \\ &= \text{Var} [\vec{\mu}_\xi^\top \vec{\mu}_\xi] + \text{Var} [2\vec{\mu}_\xi^\top L_\xi \mathbf{Z}] + \text{Var} [\mathbf{Z}^\top L_\xi^\top L_\xi \mathbf{Z}] \\ &= 4 (\vec{\mu}_\xi^\top L_\xi) (\vec{\mu}_\xi^\top L_\xi)^\top + \text{Tr} [L_\xi^\top L_\xi (L_\xi^\top L_\xi + (L_\xi^\top L_\xi)^\top)] \\ &= 4 (\vec{\mu}_\xi^\top L_\xi) (\vec{\mu}_\xi^\top L_\xi)^\top + 2\text{Tr} [L_\xi^\top \Sigma_\xi L_\xi].\end{aligned}\quad (20)$$

At this point, we have calculated the necessary quantities so that $\|\mathbf{X} - \mathbf{C}\|^2 \sim \mathcal{N}(\vec{x}; \vec{\mu}_\gamma, \Sigma_\gamma)$. However, further assumptions are needed to compute τ .

Let us assume, for the sake of computer implementation, that the distribution of $\|\mathbf{X} - \mathbf{C}\|^2$ is centered at the origin, i.e. $\vec{\mu}_\gamma = \mathbb{E} [\|\mathbf{X} - \mathbf{C}\|^2] = 0$. Recall (15)

$$\tau = \frac{r^2}{\|\vec{x} - \vec{c}\|^2}$$

and assume that r^2 is also an outcome from $\mathcal{N}(\vec{x}; \vec{\mu}_\gamma, \Sigma_\gamma)$, i.e. $R^2 \sim \mathcal{N}(\vec{x}; \vec{\mu}_\gamma, \Sigma_\gamma)$. Since $\|\mathbf{X} - \mathbf{C}\|^2$ and R^2 are independent normal random variables centered at 0, we conclude that

$$T = \frac{R^2}{\|\mathbf{X} - \mathbf{C}\|^2} \sim \mathcal{C}(0, 1), \quad (21)$$

where $\mathcal{C}(0, 1)$ stands for the standard Cauchy density function (Balakrishnan & Nevzorov, 2003). This reasoning results in positive and negative outcomes of T as opposed to what actually happened in the SEA (see section 3.1 and Fig. 7a). Our proposal, produces values mostly in positions around 0 since the Cauchy distribution has half of its mass of probability in $[-1, 1]$ (see Fig. 7b). Additionally, the heavy tails of $\mathcal{C}(0, 1)$ allow us to occasionally generate τ -values in distant positions, which contributes to the exploration in the search domain.

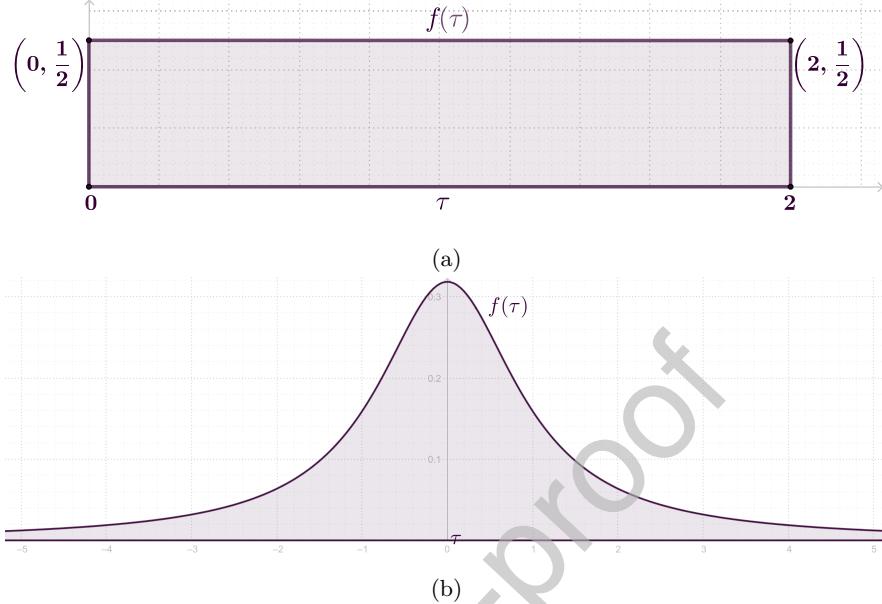


Figure 7: Possible values for τ according to the density function $f(\tau)$. (a) Previous work, T follows a uniform density function. (b) Our proposal, via statistical analysis, T follows a Cauchy density function.

Furthermore, recall that the ISO included a mutation, which we are now going to analyse. Recall (1), substitute τ into it and assume a constant centre \vec{c}_0 such that $Inv_{\vec{c},r}$ is expressed as follows

$$Inv_{\vec{c},r}(\vec{x}) = \tau(\vec{x} - \vec{c}) + \vec{c}_0. \quad (22)$$

Since \vec{x} and \vec{c} are two individuals from the population during the evolutionary cycle, we may assume that both are outcomes from random variables \mathbf{X} and \mathbf{C} , which follow a unique distribution function. In this context, we are going to assume, for the sake of simplicity, that $\mathbf{X}, \mathbf{C} \sim \mathcal{N}(\vec{\mu}, \Sigma)$ and $\vec{c}_0 = \vec{0}$. As a consequence, (22) is written in terms of random variables as

$$\boldsymbol{\Lambda} = \tau(\mathbf{X} - \mathbf{C}), \quad (23)$$

where $\boldsymbol{\Lambda}$ is a random variable which generates random vectors to mutate any particle in the search space. It would be good to have a closed form for the density function associated with $\boldsymbol{\Lambda}$. Let us express \mathbf{X} and \mathbf{C} as $\mathbf{X} = \vec{\mu} + L\mathbf{Z}_1$ and $\mathbf{C} = \vec{\mu} + L\mathbf{Z}_2$, respectively, where $\mathbf{Z}_1, \mathbf{Z}_2 \sim \mathcal{N}(\vec{0}, \mathbf{I})$ and $\Sigma = LL^T$. In this case (23) is rewritten as

$$\begin{aligned} \boldsymbol{\Lambda} &= \tau(\vec{\mu} + L\mathbf{Z}_1 - (\vec{\mu} + L\mathbf{Z}_2)) \\ &= \tau L(\mathbf{Z}_1 - \mathbf{Z}_2). \end{aligned} \quad (24)$$

It is known that $\mathbf{Z}_2 \sim -\mathbf{Z}_2$ since its density function is symmetric around zero (Balakrishnan & Nevzorov, 2003) and, therefore,

$$\begin{aligned} \mathbf{\Lambda} &\sim \tau L (\mathbf{Z}_1 + \mathbf{Z}_2) \\ &\sim \tau L \mathcal{N}(\vec{0}, 2\mathbf{I}) \\ &\sim \mathcal{N}(\vec{0}, 2\tau^2 LL^\top) \\ &\sim \mathcal{N}(\vec{0}, 2\tau^2 \Sigma). \end{aligned} \tag{25}$$

so that $\mathbf{\Lambda}$ follows a multivariate density function. Notice that the covariance matrix $2\tau^2 \Sigma$ depends on τ and the variance-covariance matrix Σ characterizing the dispersion of the population. Figure 8 exemplifies the extent of normally-distributed mutations according to different values for τ .
260

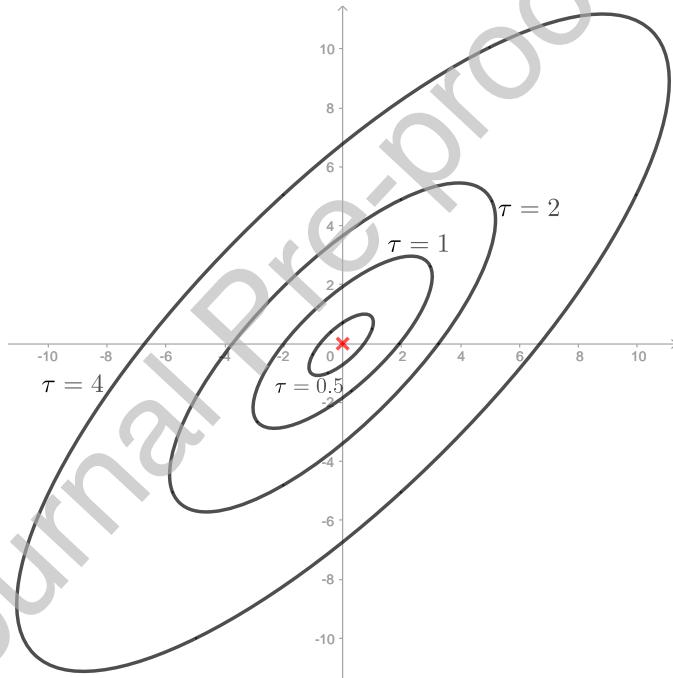


Figure 8: Normally-distributed mutations according to different values for τ .

The mutation strategy developed in this section produces perturbations in all entries of a vector position according to the variance of the population. The length of mutations varies according to the distribution of individuals in the search space. Bigger random perturbations are introduced in the direction where elongated dispersion is found, whilst smaller mutations are produced in other directions. The factor τ allow the resizing of the length of the mutation and produce even bigger or smaller perturbations in order to encourage exploration of the search space.
265

3.3.2. Bernoulli Search Operator (BRSO)

A procedure to create a vector \vec{z} from \vec{x} using random reflections is given by

$$\begin{aligned}\vec{z} &= \text{Ref}_{\vec{c}, 1-2\vec{s}}(\vec{x}) \\ &= ((1 - 2s_1)(x_1 - c_1) + c_1, \dots, (1 - 2s_d)(x_d - c_d) + c_d),\end{aligned}\tag{26}$$

where s_k is an observed value of $S_k \sim \text{Ber}(p)$ with probability value p .

In subsection 3.2, we showed that every outcome \vec{z} of (26) has a high probability of sharing half of the components of \vec{x} when a fixed probability value $p = 0.5$ is selected. Below, we introduce an idea about a fairness value for p and show that our approach produces evenly distributed random reflections.

Let us assume $p = u$, where u is an observed value from the uniform distribution $\mathcal{U}(0, 1)$. In such a case, the number of differences q between the output \vec{z} and the input \vec{x} is modelled by a Binomial distribution $Q \sim \text{Bin}(d, p)$. The corresponding density function $f(q)$ depends on a random probability value p , so that $f(q)$ can be seen as a conditional density function. Additionally, the uniform distribution is a particular case of the Beta distribution (Balakrishnan & Nevzorov, 2003). Therefore, a general case is calculated below in order to find the density function $f(q)$.

From a Bayesian perspective $f(q, p) := f(q|p)f(p)$, so that $f(q)$ can be calculated by integrating p away. It is a feasible approach by considering the density functions involved in our problem at hand: the prior Beta density

$$f(p) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1} (1-p)^{\beta-1},\tag{27}$$

where $p \in [0, 1]$, $\alpha \in \mathbb{R}^+$ and $\beta \in \mathbb{R}^+$, and the conditional Binomial density

$$f(q|p) = \binom{d}{q} p^q (1-p)^{d-q},\tag{28}$$

where $d \in \mathbb{N}$ and $q \in \{0, 1, 2, \dots, d\}$. Therefore the joint distribution $f(q, p)$ is expressed by

$$f(q, p) = \binom{d}{q} \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{q+\alpha-1} (1-p)^{d-q+\beta-1},\tag{29}$$

and

$$\begin{aligned}f(q) &= \binom{d}{q} \frac{\Gamma(\alpha + \beta)\Gamma(q + \alpha)\Gamma(d - q + \beta)}{\Gamma(\alpha)\Gamma(\beta)\Gamma(q + \alpha + d - q + \beta)} \\ &= \frac{\Gamma(d + 1)\Gamma(\alpha + \beta)\Gamma(q + \alpha)\Gamma(d - q + \beta)}{\Gamma(q + 1)\Gamma(d - q + 1)\Gamma(\alpha)\Gamma(\beta)\Gamma(q + \alpha + d - q + \beta)},\end{aligned}\tag{30}$$

which is calculated using $f(q) = \int_0^1 f(q, p) dp$ and completing the Beta density $p^{q+\alpha-1}(1-p)^{d-q+\beta-1}$.

The previous calculations allow us to get a general version of $f(q)$ using a Beta distribution as the model for the probability of generating a reflection. However, as we stated at the beginning of this section, we aim for $f(q)$ to be modelled by a uniform distribution. Since such goal is reached by setting $\alpha = 1$ and $\beta = 1$, we reduce (30) to

$$f(q) = \frac{1}{d+1}. \quad (31)$$

- 285 which is the discrete uniform density function in the domain $[0, d]$ (see Fig 9). It lead us to conclude that Q follows a discrete uniform density when p is chosen from the continuous uniform distribution. As a consequence, it is equally probable to obtain any number of reflections for a given individual in the search space, as opposed to what happened in the SEA (see Section 3.2).
 290 Therefore, our approach generates evenly distributed reflections of any particle in the search space.

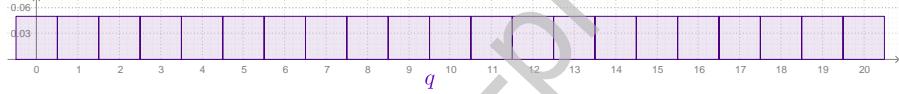


Figure 9: Example of the discrete uniform density, as stated in Eq. (31), when $d = 20$. Notice that such density produces evenly distributed number of reflections.

4. Geometric Probabilistic Evolutionary Algorithm (GPEA)

In this section, we provide the details of our proposal, the Geometric Probabilistic Evolutionary Algorithm (GPEA) whose search mechanism is built on the CDISO and the BRSO.
 295

The pseudo-code of GPEA is given in Algorithm 2. Our proposal consists in applying the formulae of section 3:

- Firstly, a Cauchy-distributed inversion map is applied (see Section 3.3.1, line 11).
- Secondly, perturbations are added to each component following a normal distribution (see Section 3.3.1), where the covariance matrix is computed via maximum likelihood (see lines 7 and 13). Note that the procedure *Reinsertion* in line 14 ensures every sample generated is inside the domain search. Such a procedure consists in applying $\lambda_k^{(i)} = x_k^{(i)} + \frac{2}{3}(l_k^{up/low} - x_k^{(i)})$ whenever a component k in $\vec{\lambda}^{(i)}$ is outside the domain, where $l_k^{up/low}$ is the upper/lower limit of the violated variable x_k .
- Finally, a new individual is generated by mixing components between the original selected individual and the previously mutated particle using the evenly distributed Reflection Map of Section 3.3.2 (see line 19). Since it is

310

not desirable to obtain the original individual as the final outcome, we enforce a random component-reflection (see line 18). As a consequence, our proposal ensures the creation of different individuals in each generation.

Algorithm 2 Geometric Probabilistic Evolutionary Algorithm (GPEA)

```

1:  $t \leftarrow 0, N \leftarrow 6d + 120, N_S \leftarrow \frac{N}{3}, \eta \leftarrow \lceil \frac{N_S}{5} \rceil$ 
2:  $\mathcal{P}^{(t)} \leftarrow \mathcal{U}(\text{Domain})$ , where  $\vec{x}^{(i)} \in \mathcal{P}^{(t)}$  ▷ First population
3: while (Stop condition is not reached) do
4:    $\mathcal{P}_{\text{sample}}^{(t)} \leftarrow N_S$  randomly chosen individuals from  $\mathcal{P}^{(t)}$  ▷ Sample
5:    $\mathcal{P}_{\text{best}}^{(t)} \leftarrow$  The best  $\eta$  individuals from  $\mathcal{P}_{\text{sample}}^{(t)}$  ▷ Best individuals
6:    $\mathcal{P}_{\text{new}}^{(t)} \leftarrow \emptyset$ 
7:    $\widehat{\Sigma} \leftarrow$  Covariance matrix of  $\mathcal{P}_{\text{sample}}^{(t)}$ 
8:   for all individuals  $\vec{x}^{(i)} \in \mathcal{P}_{\text{sample}}^{(t)}$  do
9:      $\vec{c}^{(i)} \leftarrow$  randomly chosen from  $\mathcal{P}_{\text{best}}^{(t)}$ , such that  $\vec{x}^{(i)} \neq \vec{c}^{(i)}$  ▷ Centre
10:     $\tau_i \leftarrow$  an observed value of  $T \sim \text{Cauchy}(0, 1)$ 
11:     $\vec{y}^{(i)} \leftarrow \tau_i(\vec{x}^{(i)} - \vec{c}^{(i)}) + \vec{c}^{(i)}$  ▷ Cauchy-distributed Inversion Map
12:     $\vec{\epsilon}^{(i)} \leftarrow$  an observed vector of  $\Lambda \sim \mathcal{N}(\vec{0}, 2\tau_i^2 \widehat{\Sigma})$ 
13:     $\vec{\lambda}^{(i)} \leftarrow \vec{y}^{(i)} + \vec{\epsilon}^{(i)}$  ▷ Mutation
14:     $\vec{\lambda}^{(i)} \leftarrow \text{Reinsertion}(\vec{\lambda}^{(i)})$  ▷ To ensure box-constraints
15:     $\vec{x}_m^{(i)} = (\vec{\lambda}^{(i)} + \vec{x}^{(i)}) / 2$ 
16:     $u_i$  is an observed value of  $U \sim \mathcal{U}(0, 1)$ 
17:     $\vec{s} = [s_1, \dots, s_k, \dots, s_d]^\top$ , where  $s_k$  is an observed value of  $S_k \sim \text{Ber}(u_i)$ 
18:    if  $s_k = 0 \forall k \in \{1, \dots, d\}$  then  $s_{r_a} = 1$ , where  $r_a$  is randomly chosen
19:     $\vec{z}^{(i)} \leftarrow T_{\text{ref}}(\vec{x}^{(i)}, \vec{x}_m^{(i)}, 1 - 2\vec{s})$  ▷ Reflection Map
20:     $\mathcal{P}_{\text{new}}^{(t)} \leftarrow \mathcal{P}_{\text{new}}^{(t)} \cup \vec{z}^{(i)}$ 
21:  end for
22:   $\mathcal{P}^{(t+1)} \leftarrow N$  best individuals of  $\mathcal{P}^{(t)} \cup \mathcal{P}_{\text{new}}^{(t)}$ 
23:   $t \leftarrow t + 1$ 
24: end while
25: Return the elite individual in  $\mathcal{P}^{(t)}$ 

```

315

It is worth noting that the previous procedure is applied on a random sample of the population (see line 4). Such a set of individuals is representative of the current population and allows us to diversify the selected best η individuals in line 5. Thus $N - \eta$ individuals have the chance of being part of the elite set. This is relevant since we seek to enrich the creation of particles in unexplored regions and to make an impacts in the diversity of individuals during the search procedure. Since the inversion maps depend on the location of the hyperspheres, our approach assumes that the individuals with the best fitness values are the centres of such hyperspheres (see lines 5 and 9). In this manner, we are producing new particles in positions relatively close to promising regions. Notice that we are promoting exploitation while random samples from the population encourage exploration of the search space. Finally, in line 22

320

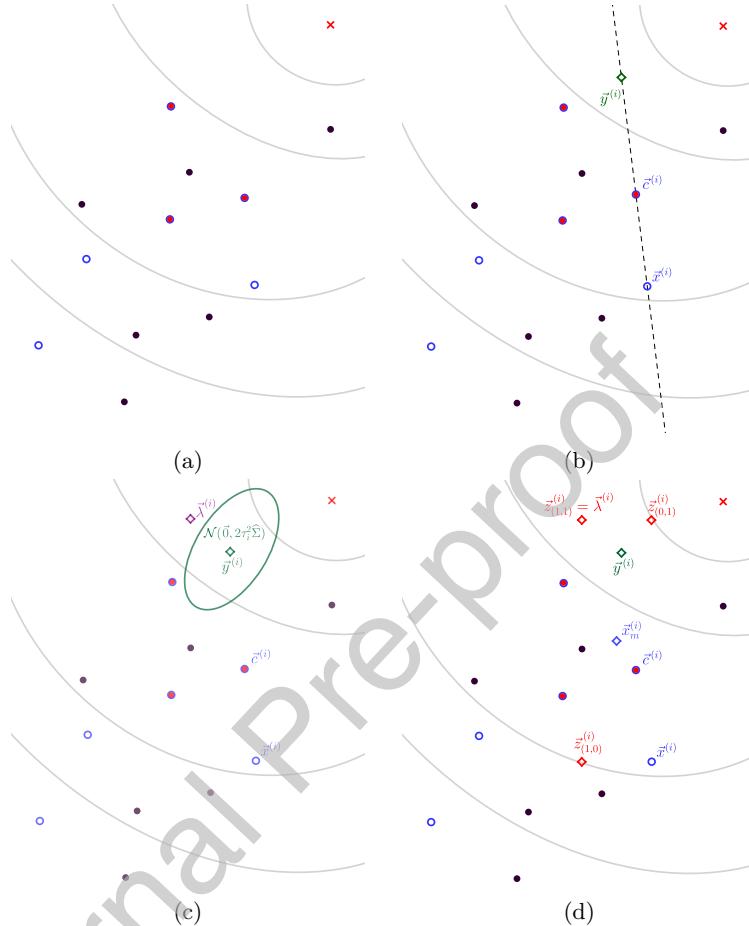


Figure 10: Illustration of possible samples obtained for an individual $\vec{x}^{(i)}$ by the GPEA. (a) From a population of 12 particles, 6 individuals are chosen as the sample set (blue and red points) and its best 3 individuals as the elite set (red points). b) A position $\vec{y}^{(i)}$ is sampled from a Cauchy-distributed Inversion Map. c) The mutation $\vec{\lambda}^{(i)}$ is obtained by adding a random vector from the density function $\mathcal{N}(\vec{0}, 2\tau_i^2 \hat{\Sigma})$. d) An evenly distributed Inversion map, with respect to individual $\vec{x}^{(i)}$ and midpoint $\vec{x}_m^{(i)}$, produces three possible outcomes, diamond-red symbol, from which only one is uniformly chosen as the child $\vec{z}^{(i)}$.

325 our method replaces worst individuals of the current population whenever new individuals with better fitness values are encountered.

In general, EAs use parameter tuning in order to find a set of optimal parameters in order to have the best performance in different scenarios. Our method, the Geometric Probabilistic Evolutionary Algorithm (GPEA), operates with a minimal parameter set, i.e. only three parameters are required: the population size N , the sample size N_S and the number of hyperspheres η to be used,

which satisfy $N \geq N_S \geq \eta$. Note that these three parameters could actually be equal. However, we suggest using smaller values for N_S and η due to empirical observations in our testing stage.

In most EAs, the number of individuals or population size N is specified by the user and remains constant during all generations, as discussed by Guan et al. (2017). Weise et al. (2016) has argued that the assigned value N is crucial in ensuring the efficiency of EAs in exploration and exploitation stages during the evolutionary process. In the EC community it is a common practice to initialise the population size using a linear equation whose input parameter depends on the dimensionality of the optimisation problem, see Bujok (2019). Other authors, such as Tanabe & Fukunaga (2014), have implemented efficient methods to control the diversity of the population by the use of a big population at the beginning of the iterations, but with (linear or non-linear) reduction of N as generations go by.

We conducted an empirical investigation, using a set of selected optimisation problems/functions, to obtain a population size which delivers average performance in most test cases. By means of the empirical observations and linear regression we found a linear equation to set the population size N , which only depends on the dimension value d of the optimisation problem. Algorithm 2 shows the recommended values for each parameter in line 1. In the current version, these values remain unchanged throughout all the generations.

Furthermore, we introduce an efficient sampling mechanism, based on sampling with replacement, to select the parents from which offspring will be generated. GPEA performs a sampling of N_S individuals representing a third of the current members of the population. The subset of N_S individuals is uniformly chosen at random in order to preserve the diversity of the population and to avoid an elitist mechanism which promotes premature convergence. The number of hyperspheres at the reproduction step is inspired by the $\mu + \lambda$ technique, where μ is the parent population size and λ is the offspring size, see Back et al. (1997); Dang & Lehre (2016); Campos et al. (2017); Ter-Sarkisov & Marsland (2011); De Jong (2016). Note that all the individuals have the same probability of being recombined in order to generate new offspring, so that GPEA is based on a non-elitist selection scheme which furnishes an efficient evolutionary search.

Additionally, inspired by the “ $\frac{1}{5}$ rule” (one every five mutations is expected to improve the parent), commonly used in standard $(\mu + \lambda)$ -Evolution Strategy, the number of hyperspheres η used in the GPEA is a fifth of N_S . Thus, it is expected that every hypersphere produces at least one new good candidate, see Schwefel (1993); Back et al. (1997); De Jong (2016). Note that the η value increases linearly with the dimension, see line 1 of algorithm 2. According to the numerical results, our search operators CDISO and BRSO behave in a complementary manner with the chosen number of selected hyperspheres.

Figure 10a exemplifies the way our proposal constructs samples from an individual $\bar{x}^{(i)}$. In Fig. 10a, a population $\mathcal{P}^{(t)}$ of 12 individuals is shown, where a subset $\mathcal{P}_{sample}^{(t)}$ of 6 individuals is chosen as the current sample from which the best $\eta = 3$ individuals (red dots) form the elite set $\mathcal{P}_{best}^{(t)}$. Note that such elite

individuals are the centers of three different hyperspheres. Next, one such center is randomly chosen for particle $\vec{x}^{(i)}$ (see Fig. 10b). According to the Cauchy-distributed inversion map a new position $\vec{y}^{(i)}$ is simulated on the straight line determined by $\vec{c}^{(i)}$ and $\vec{x}^{(i)}$. Figure 10c illustrates the mutation added to $\vec{y}^{(i)}$ using the multivariate Gaussian distribution obtained in Eq. (25). Note that the orientation of such density depends on $\mathcal{P}_{sample}^{(t)}$, while the spread is inversely proportional to the distance between $\vec{c}^{(i)}$ and $\vec{y}^{(i)}$. Let us assume that the mutation of $\vec{y}^{(i)}$ produces the position $\vec{\lambda}^{(i)}$ as shown in Fig. 10c. Finally, the random Reflection Map with respect to $\vec{x}_m^{(i)}$ and the individual $\vec{x}^{(i)}$ could generate three possible outcomes: $\vec{z}_{(1,1)}^{(i)}$ when both components are reflected, $\vec{z}_{(0,1)}^{(i)}$ or $\vec{z}_{(1,0)}^{(i)}$ when only one random component-reflection is made. Only one of such outcomes is called $\vec{z}^{(i)}$ and added as a child in $\mathcal{P}_{new}^{(t)}$ (see line 20 in Algorithm 2).

390 5. Experiments and numerical results

The experimental setup and numerical results presented in this section aim to provide a greater understanding of the performance of the GPEA. We use the benchmark functions proposed in the special session on real-parameter optimisation of the CEC 2013 competition. We also provide statistical comparisons between the GPEA and a set of 12 state-of-the-art methods based on various metaheuristics. The main characteristics of such state-of-the-art methods are summarised below:

1. iCMAESils (Liao & Sttzle, 2013). It is a hybrid algorithm based on the CMA-ES that consists of an advanced evolution strategy with covariance matrix adaptation integrated with an occasional restart strategy and increasing population size. Additionally an iterated local search algorithm is incorporated.
2. CMAES-RIS (Caraffini et al., 2013). This algorithm employs a super-fit scheme, where the super-fit individual is obtained by means of the CMA-ES. Such individual is injected to a single solution local search which perturbs each variable. It also includes a resampling mechanism and adaptive strategies to avoid premature convergence.
3. SHADE (Tanabe & Fukunaga, 2013). It is an adaptive Differential Evolution algorithm which uses a historical memory in order to adapt the control parameters of crossover and mutation operators. It is a well-established and adaptive approach.
4. JANDE (Brest et al., 2013). This Differential Evolution algorithm includes self-adaptive strategies from the jDE algorithm, population reduction, and multiple mutation strategies using a structured population to save the number of function evaluations.
5. xNES (Glasmachers et al., 2010). It is an improved version of the Natural Evolution Strategy. The algorithm is completely invariant under linear

transformations of the underlying search space, it computes all updates in the natural coordinate system, and introduce auto-adaptive strategies for each parameter.

- 420 6. NAGEDA (Segovia-Dominguez & Hernandez-Aguirre, 2015). Authors propose an Estimation of Distribution algorithm in which the parameters of the search distribution is update by the natural gradient technique. Parameter updating is guided via the Kullback-Leibler divergence between the multivariate Normal and Boltzmann density.
- 425 7. GA-TPC (Elsayed et al., 2013). Authors propose a three-parent crossover operator which is incorporated in a Genetic Algorithm.
- 430 8. PLES (Papa & ilc, 2013). Authors propose an algorithm which use a lower number of parameters and without the need for setting the control parameters to found the global optimum. This approach offers the advantage of solving any problem without any human intervention to set suitable control parameters and is based on the Genetic Algorithm approach.
- 435 9. CDASA (Korošec & Šilc, 2013). Its search mechanism is based on the Ant Colony Optimisation Algorithm and consists of transforming a real-parameter optimisation problem into a graph-search problem. This proposal is adapted for solving high-dimensional real-parameter optimisation problems.
- 440 10. HPSO (Nepomuceno & Engelbrecht, 2013). It is a modified Particle Swarm Optimisation algorithm which incorporates different update equations, referred to as behaviours, therefore any particle could change its behaviour at any iteration.
- 445 11. SPSOABC (El-Abd, 2013). Authors propose a hybrid Particle Swarm Optimisation (PSO) and Artificial Bee Colony (ABC) algorithm. This approach uses an ABC approach to improve the personal bests of the PSO.
- 450 12. SEA (Serrano-Rubio et al., 2018). This algorithm is inspired by Conformal transformations, in particular spherical inversions and aleatory reflections, see 2.3.

450 Notice that the previous methods belong to six different families: 1) Covariance Matrix Adaptation Evolution Strategy (Liao & Sttzle, 2013; Caraffini et al., 2013), 2) Adaptive Differential Evolution (Tanabe & Fukunaga, 2013; Brest et al., 2013), 3) Natural Evolution Strategy (Glasmachers et al., 2010; Segovia-Dominguez & Hernandez-Aguirre, 2015), 4) Genetic Algorithm (Elsayed et al., 2013; Papa & ilc, 2013), 5) Ant Colony and Swarm Optimisation (Korošec & Šilc, 2013; Nepomuceno & Engelbrecht, 2013; El-Abd, 2013), and 455 6) An Evolutionary Algorithm based on Spherical Inversions (SEA) (Serrano-Rubio et al., 2018). The search mechanism of the last algorithm is based on reproduction operators which perform inversions with respect to a hypersphere and reflections with respect to a hyperplane. This last method is a reference for the geometric transformations employed in the GPEA method. As can be seen we have selected several evolutionary algorithm approaches which achieve

a competitive performance for benchmark functions of the CEC 2013 competition. The numerical performance of these methods were obtained from the results provided in the web site of authors and the official website of the CEC 2013 special session.

The test suite proposed in the special session on real-parameter optimisation of the CEC 2013 consist of 28 scalable real parameter single objective benchmark functions which have been used to evaluate the performance of algorithms. The functions are grouped into three classes: a) $f_1 - f_5$: are uni-modal functions, b) $f_6 - f_{20}$: are basic multimodal functions and c) $f_{21} - f_{28}$: are composition functions.

These benchmark functions are treated as black-box problems. In (Liang et al., 2013) can be found detailed information about these benchmark problems. The details of the experimental setup are similarly to those used in the competition, and are listed below.

- The functions are evaluated on 10, 30 and 50 dimensions.
- The solution error measure is given by the next equation:

$$f(x) - f(x^*) \quad (32)$$

where $f(x)$ is the best fitness value obtained by algorithms in one run and x is the best solution. $f(x^*)$ is the optimal function value on the well-known global optimal solution x^* .

- The variable bounds for all functions are fixed as $[-100, 100]^D$
- The stop criteria is given true when either: a maximum number of function evaluations (FEs) is achieved or an error $f(x) - f(x^*) < 1 \times 10^{-8}$ is reached. The FEs is set to $10000 \times D$.
- Experiments are lead for each function on 51 independently runs.

For a more rigorous comparison, the performance of algorithms is compared using two non parametric statistical hypothesis test: a) Wilcoxon signed-rank test (for evaluating the pair-differences between the GPEA and competitor methods) and b) Friedman test.

The Wilcoxon signed rank test is adopted to verify if the performance of GPEA is better than the performance of other methods. The Wilcoxon signed-rank test is a non parametric statistical hypothesis test which is used when the average of the optimal fitness values can not be assumed to be normally distributed. The null hypothesis in the Wilcoxon signed-rank test represents that there is not significant difference between the performance of GPEA with the performance of its competitor. The alternative hypothesis represents that there is a significant difference in the performance of the GPEA against its competitor. Equation 33 presents the null hypothesis H_0 and alternative hypothesis H_1 when GPEA is compared with its competitor.

$$\begin{aligned} H_0 : \mu^{GPEA} &= \mu^{\text{competitor}} \\ H_1 : \mu^{GPEA} &\neq \mu^{\text{competitor}} \end{aligned} \quad (33)$$

500 where μ^{GPEA} and $\mu^{\text{competitor}}$ represent the average of the optimal function values of the GPEA and its competitor respectively. The p value is evaluated and compared with the significance level α on the 28 functions, the null hypothesis is rejected if the p value is less than the significance level.

505 Table 4 shows the test results for 10, 30 and 50 dimensions with two significance levels of $\alpha = 0.05$ and $\alpha = 0.1$. Column with the label R^+ represents the sum of ranks for the functions in which GPEA outperforms its competitor. R^- represents the sum of ranks for the functions in which GPEA achieves a worse or similar performance than its competitor. Higher values indicate larger performance discrepancy between algorithms. In addition a label with the legend Yes is added when the p value of the comparison between algorithms is less than 510 the significance level, otherwise the legend is No.

515 For obtaining the final rankings for all methods, we employ the Friedman test. Friedman test is a non-parametric multiple comparison technique whose purpose is to rank separately the algorithms on each problem. The final rank for each algorithm is defined by the averaging these ranks over all the problems. The p -value among all algorithms according to Friedman Test is 0.001 for 10, 30 and 50 dimensions. The analysis of results are discussed below.

- 520 • Results for functions at 10 dimensions. Table 1 shows that GPEA achieves a better performance above 50% of functions than xNES, TPC-GA, PLES, CDASA and SEA methods. For convenience of the reader, Figures 11a, 11b and 11c illustrate the comparison between GPEA and other methods at 10, 30 and 50 dimensions respectively. In accordance with the Wilcoxon signed rank test at $\alpha = 0.05$ and $\alpha = 0.1$ confirms that the performance of GPEA is better than PLES, CDASA and SEA methods. On the other hand, there is no statistical evidence to establish that GPEA presents a better performance than xNES and TPC-GA methods. Table 5 shows that GPEA achieves the seventh place in the ranking of all algorithms over functions at 10 dimensions.
- 525 • Results for functions at 30 dimensions. Table 2 shows that GPEA achieves a better performance above 50% of functions than CMAES-RIS, xNES, NAGEDA, TPC-GA, PLES, CDASA, fk-PSO, SPSOABC and SEA methods. GPEA and jande present a similar performance. In accordance with the Wilcoxon signed rank test at $\alpha = 0.05$ and $\alpha = 0.1$ confirms that the performance of GPEA is better than TPC-GA, PLES, CDASA, fk-PSO and SEA methods. When $\alpha = 0.1$ there is a significance difference to establish that the performance of GPEA is better than NAGEDA and SPSOABC methods. On the other hand, there is no statistical evidence to establish that GPEA presents a better performance than CMAES-RIS and jande methods. Table 5 shows that GPEA achieves the third place in the ranking of all algorithms over functions at 30 dimensions.
- 530 • Results for functions at 50 dimensions. Table 3 shows that GPEA achieves a better performance above 50% of functions than CMA-RIS, jande, NAGEDA,

545

550

TPC-GA, PLES, CDASA, fk-PSO, SPSOABC and SEA methods. In accordance with Table 4 confirms that the performance of GPEA is better than TPC-GA, PLES, CDASA, fk-PSO and SEA methods. When $\alpha = 0.1$ there is a significance difference to establish that the performance of GPEA is better than NAGEDA method. On the other hand, there is no statistical evidence to establish that GPEA presents a better performance than CMAES-RIS, jande and SPSOABC methods. Table 5 shows that GPEA achieves the third place in the ranking of all algorithms over functions at 50 dimensions.

555

560

565

GPEA shows an improvement in its performance whenever the dimension size is higher than 10, which means that our proposed method reaches better results as the dimension size growths. The Cauchy-distributed inversion map allows the algorithm to explore potential regions where the global optimum may be located. The parameter τ randomly perturbs the locations of the parents. The length of mutations depends on the distribution of population. Bigger random perturbations are added to the parents over directions where there are higher dispersion directions. Random reflections introduce entries not previously considered in the mutation. In this way, our proposal diversifies the possible directions in which unexplored regions can be found. Additionally, the probabilistic framework of the proposed method provides a comprehensive way to implement the algorithm and to understand its behaviour. GPEA needs a minimum number of parameters in comparison with other methods. It has a better performance than PLES, which is designed to use a minimum number of parameters without any human intervention. In conclusion, in spite of GPEA using fixed values in its only three parameters, it performs very well against other state-of-the-art algorithms with highly sophisticated adaptive techniques.

Problem	GPEA	icmaefsls	CMAFS-RIS	SHADE	jaunde	xNES	NAGEDA	TPC-GA	PLES	CDASA	Re-PSO	SPOABC	SEA
\mathcal{F}_1	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
\mathcal{F}_2	1.6503e+03	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.7180e+03	9.7704e+01	0.000e+00	0.000e+00	2.4152e+06	0.000e+00	1.4422e+05	1.4864e+05
\mathcal{F}_3	7.5955e+03	0.000e+00	7.0425e+01	1.2663e+01	1.6071e+00	1.3271e+03	3.6049e+06	0.000e+00	2.2127e+08	1.9775e+11	6.7491e+05	1.2764e+05	9.2619e+06
\mathcal{F}_4	1.5553e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	1.2429e+01	1.66548e+01	0.000e+00	0.000e+00	1.3423e+04	4.1566e+02	2.3427e+06	2.3953e+03
\mathcal{F}_5	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00	0.000e+00
\mathcal{F}_6	8.6844e+00	0.000e+00	1.0978e+00	7.8854e+00	8.4982e+00	7.1487e+00	7.6960e+01	0.000e+00	1.6399e+01	5.04743e+00	2.6364e+00	4.8635e+00	5.8064e+00
\mathcal{F}_7	4.5897e-01	4.9026e-06	5.3332e+01	3.2593e-03	9.4791e-01	1.5049e+00	5.9081e+00	4.2450e-02	4.7003e-01	1.1497e+02	1.9215e+00	4.8742e-01	1.6256e-01
\mathcal{F}_8	2.0363e+01	2.0358e+01	3.3835e+00	2.0353e+01	2.0348e+01	2.0342e+01	2.0369e+01	2.0419e+01	2.0318e+01	2.0316e+01	2.0317e+01	2.0316e+01	2.0317e+01
\mathcal{F}_9	2.1013e+00	2.8622e-01	3.5881e+00	1.1987e-02	7.0906e-02	4.2746e+00	4.1140e+00	1.9812e+00	3.3931e+00	6.8125e+00	5.0506e+00	2.7478e+00	4.4806e+00
\mathcal{F}_{10}	1.3395e-01	0.000e+00	1.2360e-02	1.1987e-02	7.0906e-02	7.6758e-03	2.1154e-03	7.6332e-00	2.7313e-01	1.7070e-01	2.6660e-01	5.1306e-01	4.4080e-02
\mathcal{F}_{11}	9.3643e-01	4.7717e-01	3.5702e+00	7.0906e-02	7.0906e-02	1.0730e-01	1.7630e-01	2.7313e-00	4.6822e-01	1.7594e-01	2.6660e-01	5.1306e-01	4.9149e-01
\mathcal{F}_{12}	8.9351e+00	2.3412e-01	1.2856e+01	3.1413e+00	6.1144e+00	9.5594e+00	8.7856e+00	6.0347e+00	2.3412e+00	2.1889e+01	7.0428e+00	6.2891e+00	1.1549e-01
\mathcal{F}_{13}	1.4635e+01	3.3347e-01	3.7708e+00	5.0296e-01	7.8102e+00	1.8718e+01	4.6667e+00	9.8088e+00	4.3246e+01	3.4535e+01	5.2419e+01	1.1479e+01	2.5359e-01
\mathcal{F}_{14}	1.9088e+01	5.0751e+01	1.0193e+02	4.8984e-03	5.0296e-02	4.2974e-02	1.0965e-03	1.0965e-03	2.4452e+01	3.7844e+01	4.5404e+02	4.9562e+02	8.4237e-02
\mathcal{F}_{15}	5.8744e+02	6.1744e+02	4.2075e+02	8.4017e-02	4.2918e+02	4.2918e+02	9.9129e+02	7.3421e+02	1.0581e+03	1.0801e+03	2.6186e+01	4.0673e+01	4.9720e+01
\mathcal{F}_{16}	1.0840e+00	3.7298e-01	1.6427e-01	7.0799e-01	1.0991e+00	1.1523e+00	1.1818e+00	1.2535e+00	7.2925e-01	2.6186e+01	4.0673e+01	6.6334e-01	6.6334e-01
\mathcal{F}_{17}	1.0552e+01	1.1183e+01	1.0377e+01	1.0122e+01	9.9240e+00	9.9240e+00	9.9173e+01	2.7756e+01	1.1237e+01	2.7660e+01	1.0978e+01	9.6002e+00	1.2636e-01
\mathcal{F}_{18}	1.0657e+01	2.9770e+01	1.6878e+01	2.77116e+01	1.9271e+01	1.9271e+01	1.8041e+01	3.1255e+01	1.8041e+01	3.1255e+01	1.8041e+01	3.1255e+01	2.16570e-01
\mathcal{F}_{19}	4.0621e-01	6.9807e-01	8.1433e-01	3.4352e-01	3.1993e-01	1.0504e+00	1.53638e+00	5.0119e-01	1.5891e+00	5.0119e-01	1.5891e+00	5.00533e-01	2.8889e-01
\mathcal{F}_{20}	2.6424e+00	2.7223e+00	4.1600e+00	2.1572e+00	2.7178e+00	3.2743e+00	3.40191e+02	3.1746e+00	3.5870e+00	4.0235e+00	2.50250e+00	3.1050e-00	3.1050e-00
\mathcal{F}_{21}	3.8892e+02	4.0184e+02	1.6075e+02	4.8396e+00	3.5113e+02	3.6879e+02	4.01191e+02	3.9234e+02	3.4328e+02	3.4328e+02	3.4328e+02	3.4328e+02	4.00194e-02
\mathcal{F}_{22}	1.0636e+02	1.6553e+02	2.4386e+02	4.8396e+00	9.1879e+01	6.3636e+02	6.7324e+02	9.0726e+01	6.5045e+02	1.32236e+02	1.32236e+02	5.49323e+02	1.06323e-03
\mathcal{F}_{23}	6.3898e-02	4.0768e+01	8.3481e+02	4.6061e+02	8.1116e+02	7.7649e+02	5.09074e+02	8.3963e+02	5.2699e+03	5.1483e+02	5.1483e+02	5.1483e+02	5.1483e+02
\mathcal{F}_{24}	2.0751e+02	1.3162e+02	1.1851e+02	1.9314e+02	2.0851e+02	1.86356e+02	2.0715e+02	2.1286e+02	2.11112e+02	2.11112e+02	2.00556e+02	2.00556e+02	2.1233e-02
\mathcal{F}_{25}	2.0149e+02	1.9206e+02	1.9206e+02	2.0015e+02	2.09553e+02	2.0682e+02	2.03134e+02	2.1496e+02	2.2089e+02	2.1496e+02	2.0469e+02	2.0469e+02	2.06144e+02
\mathcal{F}_{26}	1.7188e+02	1.1750e+02	1.6110e+02	1.3333e+02	1.3301e+02	2.30120e+02	1.24106e+02	1.7066e+02	1.9641e+02	2.14179e+02	1.8880e+02	1.3130e+02	1.4544e+02
\mathcal{F}_{27}	3.7755e+02	3.2541e+02	3.1281e+02	3.0000e+02	4.9412e+02	3.6168e+02	3.9785e+02	3.1281e+02	4.6470e+02	5.7526e+02	3.6961e+02	3.5063e+02	3.4930e+02
\mathcal{F}_{28}	2.8431e+02	2.2396e+02	2.0588e+02	3.0000e+02	2.8824e+02	3.0000e+02	3.5311e+02	2.9216e+02	4.9611e+02	3.1950e+02	3.2584e+02	3.2942e+02	3.7203e+02

Table 1: Average of fitness values reached for each algorithm in 10 dimension problems. The best average values are typed in boldface. Average fitness values above achieved average values by GPEA appears as underlined. These values indicate the algorithm that corresponds this value presents a worse performance than GPEA.

Problem	GPEA	CMAES-RIS	SHADE	xNES	jaude	NAGEADA	TPC-GA	PLES	CDASA	fl-PSO	SPSO-ABC	SEA
\mathcal{F}_1	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00	0.0000e+00
\mathcal{F}_2	1.5441e-06	0.0000e+00	0.0000e+00	9.0022e-03	1.2914e-05	5.02119e-01	2.3871e-09	3.7962e+07	2.4104e-05	9.5210e-05	8.7504e-05	4.7624e+05
\mathcal{F}_3	1.3782e+07	0.0000e+00	2.2375e+03	4.0206e+01	1.98414e+06	1.2105e+04	2.3871e-09	1.39353e-09	1.3448e+07	9.5210e-05	8.7504e-05	5.1589e-07
\mathcal{F}_4	3.2333e+03	0.0000e+00	0.0000e+00	1.9216e-04	1.9720e+04	4.4668e-01	0.0000e+00	1.3839e+01	4.4675e+04	4.4675e+04	4.4675e+04	2.3951e+08
\mathcal{F}_5	0.0000e+00	0.0000e+00	0.0000e+00	1.2606e-08	1.5580e-06	1.5580e-06	1.2606e-08	1.8296e-01	4.7760e+02	4.7760e+02	4.7760e+02	7.2820e+03
\mathcal{F}_6	3.3950e+01	6.9353e-04	5.9506e-01	7.9292e+00	5.8521e+00	0.0000e+00	0.0000e+00	2.4251e+01	7.7653e+01	2.9920e+01	2.8977e+01	2.8939e+01
\mathcal{F}_7	1.0720e+01	7.0064e-02	4.4831e-01	4.6016e-00	9.8167e+00	6.1520e-02	1.3829e+01	2.9107e+01	1.1766e+02	6.9451e+01	5.1155e+01	9.3835e+01
\mathcal{F}_8	2.0942e+01	2.0885e-01	2.0728e+01	2.0946e+01	2.0946e+01	2.0952e+01	2.0946e+01	2.0919e+01	2.0925e+01	2.0925e+01	2.0925e+01	2.0949e+01
\mathcal{F}_9	1.4735e+01	4.3384e+00	2.36539e+01	2.7166e+01	1.0148e+01	1.0148e+01	1.0145e+01	3.6055e+01	3.3099e+01	2.3452e+01	1.8478e+01	2.7255e+01
\mathcal{F}_{10}	2.2937e-01	0.0000e+00	3.3123e-03	7.6855e-02	7.9055e-02	0.0000e+00	0.0000e+00	8.6810e-02	1.1799e-01	3.5455e-02	2.2944e-01	1.3216e-01
\mathcal{F}_{11}	1.1686e+01	2.2522e+00	2.5362e+01	3.1409e+01	3.1409e+01	1.4466e+02	1.4466e+02	2.39118e+01	1.6477e+02	2.3552e+01	1.0945e+00	1.0945e+02
\mathcal{F}_{12}	3.5272e+01	1.7224e+00	7.9423e+01	2.3039e+01	2.9127e+01	1.4625e+01	1.4625e+01	4.1140e+01	1.17066e+00	1.17066e+00	1.17066e+00	1.2733e+02
\mathcal{F}_{13}	8.4119e+01	2.1570e+00	1.75535e+02	5.0357e+01	7.0740e-01	6.5400e+01	1.4671e+02	8.4143e+01	1.28583e+02	1.28583e+02	1.28583e+02	1.30595e-02
\mathcal{F}_{14}	2.6795e+02	7.0778e-02	7.9180e+02	3.1841e-02	1.0832e+03	6.9788e-03	9.2475e+02	6.6378e+02	7.0361e-02	6.6378e+02	6.6378e+02	3.0595e-03
\mathcal{F}_{15}	3.2687e+03	2.5950e+02	3.1296e+03	3.2179e-03	4.8310e-03	1.7236e-03	6.7105e+03	3.9702e+03	3.8719e+03	3.4170e+03	3.5496e+03	4.0063e+03
\mathcal{F}_{16}	2.3316e+00	3.7456e-01	1.0704e-01	9.1315e-01	2.2791e+00	2.4388e+00	2.3922e+00	2.4987e+00	3.1518e+00	3.2566e+01	3.4825e+01	4.0963e+00
\mathcal{F}_{17}	3.9704e+01	3.4255e+01	5.4998e+01	3.0434e+01	3.0434e+01	6.3644e+01	6.3644e+01	5.4377e+01	5.4262e+02	5.6120e+01	6.4406e+01	7.2780e+01
\mathcal{F}_{18}	5.9133e+01	4.0091e+01	1.8919e+02	1.2311e+01	1.2311e+01	1.4625e+01	1.4625e+01	1.1738e+02	1.1738e+02	1.1738e+02	1.1738e+02	1.1738e+02
\mathcal{F}_{19}	2.1755e+00	2.2358e+00	2.8039e+00	1.3557e+00	1.3557e+00	1.2218e+00	1.2218e+00	3.2823e+00	2.4106e+01	3.2109e+00	3.12118e+00	6.0780e+00
\mathcal{F}_{20}	1.0816e+01	1.4350e+01	1.0473e+01	1.4252e+01	1.1639e+01	1.3932e+01	1.3932e+01	1.3737e+01	1.4257e+01	1.4755e+01	1.1994e+01	1.1979e+01
\mathcal{F}_{21}	2.8884e+02	1.8824e+02	1.8627e+02	1.71744e+03	9.8095e+01	5.11621e+01	1.2193e+03	2.9386e+02	2.98046e+02	2.7682e+02	3.1135e+02	3.1848e+02
\mathcal{F}_{22}	3.5392e+02	5.3346e+02	4.0296e+03	3.5078e+03	4.60061e+03	1.5687e-03	6.5922e+03	3.2493e+03	4.8921e+02	5.8585e+02	8.4117e+01	3.2937e+03
\mathcal{F}_{23}	3.3974e+03	2.69111e+02	4.0296e+03	2.17744e+03	2.0525e+02	2.25810e+02	2.25810e+02	4.3289e+03	5.0000e+03	5.4094e+03	5.3753e+03	4.1843e+03
\mathcal{F}_{24}	2.2470e+02	2.0000e+02	2.59130e+02	2.61944e+02	2.61944e+02	2.48118e+02	2.48118e+02	2.74466e+02	2.9823e+02	2.48120e+02	2.5089e+02	4.8347e+03
\mathcal{F}_{25}	2.5507e+02	2.3975e+02	2.8206e+02	2.15717e+02	2.0208e+02	2.57578e+02	2.57578e+02	3.2697e+02	3.2697e+02	2.4874e+02	2.7532e+02	2.6561e+02
\mathcal{F}_{26}	2.00094e+02	3.0000e+02	1.9717e+02	7.4869e+02	3.8763e+02	7.2161e+02	6.1815e+02	4.8929e+02	1.0312e+03	1.1485e+03	1.0840e+03	9.1034e+02
\mathcal{F}_{27}	6.4396e+02	3.0000e+02	2.45101e+02	5.3873e+02	3.0000e+02	4.5462e+02	3.0000e+02	3.0000e+02	2.0756e+03	3.8666e+02	4.0651e+02	3.3317e+02
\mathcal{F}_{28}	3.0000e+02											

Table 2: Average of fitness values reached for each algorithm in 30 dimension problems. The best average values are typed in boldface. Average fitness values above achieved average values by GPEA appears as underlined. These values indicate the algorithm that corresponds this value presents a worse performance than GPEA.

Problem	GPEA	CMAES-RIS	SHADE	xNES	NAGEDA	TPC-GA	PLES	CDAS _A	R-PSO	SPOABC	SEA
\mathcal{F}_1	0.0000e+00										
\mathcal{F}_2	0.0435e+06	0.0000e+00	2.6565e+04	6.7646e-08	1.7457e+05	0.0000e+00	4.7607e+05	1.9294e+06	2.7565e+06	4.9512e+05	8.7108e+05
\mathcal{F}_3	7.2035e+07	2.0109e+02	2.8265e+05	8.7397e+00	4.7824e+05	1.7304e+03	1.8138e+10	1.0571e+08	2.1813e+08	9.6805e+08	3.0945e+08
\mathcal{F}_4	1.1588e+04	0.0000e+00	1.6131e+03	8.3433e+04	2.6391e+01	0.0000e+00	3.3306e+00	5.1286e+04	5.2513e+02	4.8751e+13	6.8720e+03
\mathcal{F}_5	0.0000e+00	1.1705e+08	0.0000e+00	2.4273e+06	1.8071e+06	0.0000e+00	3.1906e+10	8.3988e-06	0.0000e+00	0.0000e+00	0.0000e+00
\mathcal{F}_6	4.6436e+01	4.1887e+01	9.5134e+00	4.2706e+01	4.2973e+01	4.3447e+01	4.7228e+01	9.7037e+01	4.8015e+01	4.0512e+01	6.8813e+01
\mathcal{F}_7	2.6429e+01	5.4399e-01	4.8110e+01	2.2966e+01	2.9422e+01	2.8597e-02	1.9331e+01	4.1650e+01	1.2783e+02	1.0431e+02	7.8054e+01
\mathcal{F}_8	2.1136e+01	2.1107e+01	2.1045e+01	2.1128e+01	2.1130e+01	2.1130e+01	2.1165e+01	2.1165e+01	2.1165e+01	2.1165e+01	2.1115e+01
\mathcal{F}_9	2.9500e+01	8.1775e+00	4.7198e+01	5.5124e+01	5.3298e+01	1.3751e+01	1.8156e+01	7.2966e+01	6.1487e+01	4.6663e+01	5.8409e+01
\mathcal{F}_{10}	4.4797e+01	0.0000e+00	8.5592e+03	7.3658e+02	1.4733e+01	0.0000e+00	1.0474e-01	2.9942e+01	4.6565e-02	2.1816e-01	1.5548e-01
\mathcal{F}_{11}	2.5518e+01	5.9373e+00	35.68e+01	1.9509e+02	5.9015e+01	3.1215e+01	5.5677e+01	2.1463e+00	8.6094e+01	6.6161e-02	2.7621e+02
\mathcal{F}_{12}	7.9110e-01	5.7747e+00	2.6561e+02	5.8632e+01	9.7158e+01	6.3933e+01	3.0798e+02	9.6213e+01	4.0509e+02	2.6691e+02	1.7271e+02
\mathcal{F}_{13}	1.6893e+02	5.7293e+00	4.5629e+02	1.4538e+02	1.7587e+02	1.4026e+02	3.0542e+02	1.9118e+02	6.3778e+02	2.7378e+02	4.5113e+02
\mathcal{F}_{14}	6.1808e+02	8.5920e+02	1.4939e+03	3.4536e+02	8.0144e+00	4.1908e+03	1.3232e+04	2.5498e+03	5.0582e+03	1.0777e+03	5.6963e+03
\mathcal{F}_{15}	6.85530e+03	6.4199e+02	6.3043e+03	9.1809e+03	9.1809e+03	1.3876e+04	1.3559e+04	9.4046e+03	8.5138e+03	7.3316e+03	7.4239e+03
\mathcal{F}_{16}	3.2500e+00	6.2803e+01	8.6630e+02	1.2821e+00	3.1319e+00	3.3833e+00	3.3469e+00	3.3837e+00	2.0708e+00	4.9728e+01	1.3675e+00
\mathcal{F}_{17}	7.5796e+01	5.7505e+01	1.0195e+02	5.0786e+01	1.8753e+02	3.5034e+02	1.1468e+02	6.0059e+02	5.8163e+01	1.1643e+02	5.1959e+01
\mathcal{F}_{18}	1.1601e+02	6.4278e+01	4.1789e+02	2.1822e+02	1.3679e+02	2.1822e+02	3.5163e+02	1.6788e+02	6.3251e+02	4.4265e+02	1.3249e+02
\mathcal{F}_{19}	5.1744e+00	3.6246e+00	5.0414e+00	2.6414e+00	2.2386e+00	5.0515e+00	2.6437e+00	8.9245e+00	1.2777e+02	3.6923e+00	5.1857e+00
\mathcal{F}_{20}	2.0960e+01	2.4391e+01	2.4328e+01	1.9271e+01	2.1508e+01	2.2209e+01	2.0946e+01	2.3409e+01	2.3738e+01	2.4293e+01	2.0639e+01
\mathcal{F}_{21}	7.7558e+02	2.0000e+02	2.8539e+02	8.4485e+02	8.2422e+02	8.1683e+02	1.1166e+03	7.9256e+02	7.5811e+02	8.8638e+02	8.3365e+02
\mathcal{F}_{22}	7.1106e+02	5.8713e+02	2.3934e+03	1.3263e+01	3.0965e+01	2.5170e+03	1.2871e+04	3.5100e+03	6.4966e+03	7.3186e+02	2.2196e+03
\mathcal{F}_{23}	6.5182e+03	5.5746e+02	8.3710e+03	7.6289e+03	9.4753e+03	1.3713e+04	1.2771e+04	9.9252e+03	1.0215e+04	1.0141e+04	7.4020e+03
\mathcal{F}_{24}	2.5235e+02	2.0001e+02	3.2175e+02	2.3384e+02	2.8856e+02	2.4812e+02	2.3377e+02	3.7651e+02	3.8323e+02	3.0029e+02	3.3418e+02
\mathcal{F}_{25}	3.0687e+02	2.7429e+02	3.3964e+02	3.1673e+02	3.0688e+02	3.3713e+02	3.8594e+02	4.4390e+02	4.0358e+02	3.6419e+02	3.8657e+02
\mathcal{F}_{26}	3.0982e+02	2.4123e+02	2.0003e+02	2.5751e+02	3.9708e+02	2.8230e+02	4.2202e+02	3.4361e+02	3.9037e+02	3.9866e+02	4.1840e+02
\mathcal{F}_{27}	1.0025e+03	3.0210e+02	1.2457e+03	9.3554e+02	1.1633e+03	7.8422e+02	7.0701e+02	2.0320e+03	1.5090e+03	1.3152e+03	1.7757e+03
\mathcal{F}_{28}	5.2287e+02	4.0000e+02	4.5784e+02	9.4335e+02	1.2415e+03	1.4250e+03	4.0000e+02	4.5930e+02	1.0353e+03	1.6281e+03	1.0086e+03

Table 3: Average of fitness values reached for each algorithm in 50 dimension problems. The best average values are typed in boldface. Average fitness values above achieved average values by GPEA appears as underlined. These values indicate the algorithm that corresponds this value presents a worse performance than GPEA.

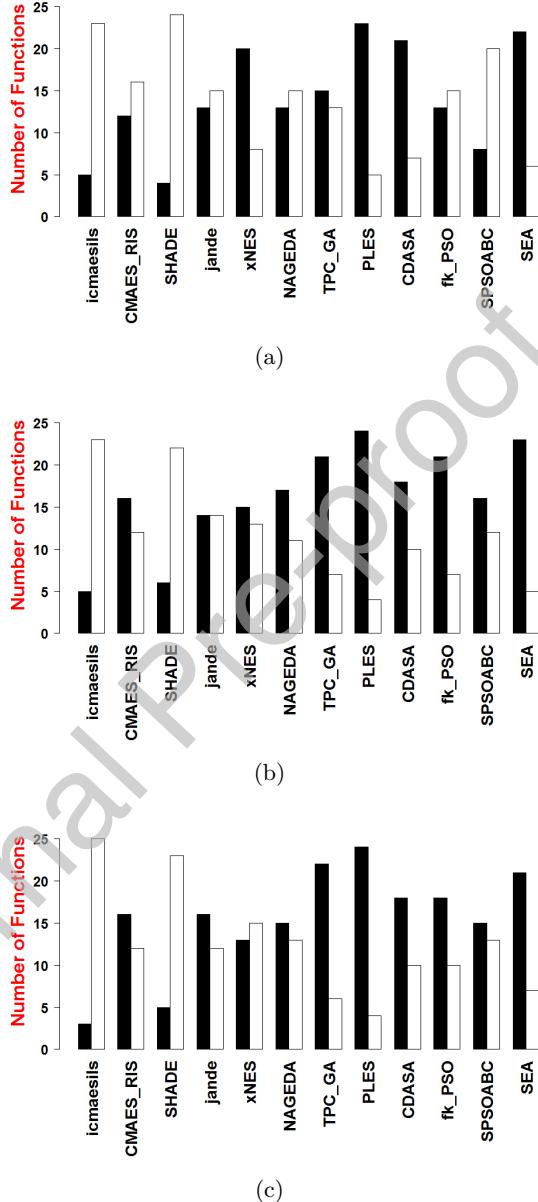


Figure 11: A comparison of the performance of GPEA with other methods over 28 benchmark functions. The dimensions of benchmark functions are $D = 10$, $D = 30$ and $D = 50$ for (a), (b) and (c) respectively. The black area indicates the number of benchmark functions where GPEA achieves a better result than the other algorithm. The white area indicates the number of benchmark functions where GPEA achieves a worse result than the other algorithm.

Algorithm	R ⁺	R ⁻	p-value	$\alpha = 0.05$	$\alpha = 0.1$
10 dimensions					
GPEA versus icmaesils	59	346	0.001203	Yes	Yes
GPEA versus CMAES-RIS	188	217	0.750883	No	No
GPEA versus SHADE	58	347	0.001314	Yes	Yes
GPEA versus jande	221	184	0.638453	No	No
GPEA versus xNES	273	132	0.124146	No	No
GPEA versus NAGEDA	232	173	0.453712	No	No
GPEA versus TPC-GA	223	182	0.675167	No	No
GPEA versus PLES	390	15	0.000021	Yes	Yes
GPEA versus CDASA	351	54	0.000705	Yes	Yes
GPEA versus fk-PSO	220	185	0.656707	No	No
GPEA versus SPSOABC	141	264	0.190875	No	No
GPEA versus SEA	350	55	0.001005	Yes	Yes
30 dimensions					
GPEA versus icmaesils	69	336	0.002843	Yes	Yes
GPEA versus CMAES-RIS	251	154	0.280403	No	No
GPEA versus SHADE	91	314	0.013275	Yes	Yes
GPEA versus jande	214	191	0.810136	No	No
GPEA versus xNES	191	214	0.754794	No	No
GPEA versus NAGEDA	287	118	0.058471	[No]	Yes
GPEA versus TPC-GA	330	75	0.004628	Yes	Yes
GPEA versus PLES	397	8	0.000012	Yes	Yes
GPEA versus CDASA	301	104	0.023924	Yes	Yes
GPEA versus fk-PSO	348	57	0.001005	Yes	Yes
GPEA versus SPSOABC	284	121	0.055168	[No]	Yes
GPEA versus SEA	364	41	0.000326	Yes	Yes
50 dimensions					
GPEA versus icmaesils	30	375	0.000099	Yes	Yes
GPEA versus CMAES-RIS	238	167	0.427880	No	No
GPEA versus SHADE	94	311	0.018808	Yes	Yes
GPEA versus jande	249	157	0.294876	No	No
GPEA versus xNES	206	199	0.923441	No	No
GPEA versus NAGEDA	272	133	0.096199	[No]	Yes
GPEA versus TPC-GA	329	76	0.005857	Yes	Yes
GPEA versus PLES	392	13	0.000017	Yes	Yes
GPEA versus CDASA	296	109	0.032498	Yes	Yes
GPEA versus fk-PSO	305	100	0.018808	Yes	Yes
GPEA versus SPSOABC	256	149	0.208681	No	No
GPEA versus SEA	340	65	0.002030	Yes	Yes

Table 4: Results of multiple Wilcoxon tests between GPEA and other algorithms in 10, 30 and 50 dimension problems. The values in boldface indicate that there is significant improvement of the performance of GPEA with respect to the performance of its competitor. In addition the legends in brackets highlight cases where there is not a statistical evidence to establish a comparison between algorithms for some α value.

R	10 dimensions		30 dimensions		50 dimensions	
	Algorithm	Mean ranking	Algorithm	Mean ranking	Algorithm	Mean ranking
1	icmaesils	3.553571	icmaesils	3.214286	icmaesils	2.910714
2	SHADE	4.035714	SHADE	3.642857	SHADE	3.732143
3	SPSOABC	5.250000	GPEA	5.553571	GPEA	6.017857
4	CMAES-RIS	5.892857	jande	6.071429	CMAES-RIS	6.053571
5	fk-PSO	6.392857	CMAES-RIS	6.232143	SPSOABC	6.696429
6	jande	6.464286	xNES	6.785714	jande	6.714286
7	GPEA	6.571429	SPSOABC	6.839286	xNES	6.839286
8	TPC-GA	6.660714	fk-PSO	7.410714	fk-PSO	7.303571
9	NAGEDA	7.035714	NAGEDA	7.857143	NAGEDA	7.625000
10	xNES	8.571429	CDASA	8.250000	CDASA	7.732143
11	SEA	9.428571	TPC-GA	8.392857	TPC-GA	8.517857
12	CDASA	9.785714	SEA	9.517857	SEA	9.625000
13	PLES	11.357143	PLES	11.232143	PLES	11.232143

Table 5: Average ranking of GPEA and other algorithms according to Friedman test. The statistical results are presented according to each experiment using the functions at 10, 30 and 50 dimensions. The p -values are: 5.8645e-20, 1.5915e-18 and 5.622e-18 for 10, 30 and 50 dimensions. The values for GPEA method are typed in boldface.

6. Conclusion and Future Scope

In this paper, we propose the Geometric Probabilistic Evolutionary Algorithm (GPEA) for solving continuous optimisation problems by means of a probabilistic implementation of geometrically motivated search operators. GPEA includes two new search operators called the Cauchy Distributed Inversion Search Operator (CDISO) and the Bernoulli Reflection Search Operator (BRSO). The designing of the CDISO and the BRSO is carried out based on a detailed statistical analysis of the operators implemented by the Spherical Evolutionary Algorithm (SEA).

The CDISO and the BRSO take the geometric features of the SEA operators (ISO and RSO) and add a significant improvement to the reproduction and mutation of the population using well defined distribution functions that lead to an efficient search mechanism. The CDISO implements the inversion with respect to a hypersphere using only one equation and incorporates a random variable from a Cauchy distribution, needing fewer calculations than the ISO. In addition, the CDISO incorporates a new mutation using the information of the best individuals. We estimate a multivariate normal distribution for mutating the new individuals in the direction where it is expected the fitness of the individuals will improve. The BRSO incorporates a new mechanism which avoids the premature convergence and speeds up the search of the global optimum. It implements a new mechanism in which the vector components of the individuals have the same probability of being modified.

590 The CDISO and the BRSO operate jointly in the GPEA exploring and exploiting the search space using few parameters. According to the experimental setup GPEA presents a competitive performance against 12 state-of-the-art methods whose search mechanisms approaches are based on various metaheuristics. In order to evaluate the performance of GPEA, we used the benchmark
 595 functions proposed in the special session on real-parameters optimisation of the CEC 2013 competition. According to the numerical results, GPEA has a better performance when the dimension of the problem increased. For instance, GPEA is ranking seventh in 10 dimensions, while it ranks third in 30 and 50 dimensions.

600 Regarding future work, we will implement additional techniques such as path prediction and modelling dependency between variables to enhance the search capability of GPEA. The population size, the sampling size and the number of hyperspheres are parameters which can be adapted according to the information gathered in each iteration in order to make better use of the available number of
 605 function evaluations. On the other hand we will introduce our proposed method into the multi-objective optimization field after further research.

7. Acknowledgment

The authors would like to thank to the Science and Technology Council of México, CONACyT, for its financial support during this research throughout
 610 the project CB-2015/256126.

References

- Amari, S.-i. (2016). *Information Geometry and Its Applications*. (1st ed.). Springer Publishing Company, Incorporated.
- Back, T., Fogel, D. B., & Michalewicz, Z. (Eds.) (1997). *Handbook of Evolutionary Computation*. Computational Intelligence Library (1st ed.). Bristol, UK: Oxford University Press.
- Balakrishnan, N., & Nevzorov, V. B. (2003). *A Primer on Statistical Distributions*. Wiley-Interscience. doi:10.1002/0471722227.
- Brest, J., Bošković, B., Zamuda, A., Fister, I., & Mezura-Montes, E. (2013).
 620 Real parameter single objective optimization using self-adaptive differential evolution algorithm with more strategies. In *2013 IEEE Congress on Evolutionary Computation* (pp. 377–383). IEEE. doi:10.1109/CEC.2013.6557594.
- Bujok, P. (2019). An evaluative study of adaptive control of population size in differential evolution. In *International Conference on Artificial Intelligence and Soft Computing* (pp. 421–431). Springer.
- Campos, J., Ge, Y., Fraser, G., Eler, M., & Arcuri, A. (2017). An empirical evaluation of evolutionary algorithms for test suite generation. In *International Symposium on Search Based Software Engineering* (pp. 33–48). Springer.

- 630 Caraffini, F., Iacca, G., Neri, F., Picinali, L., & Mininno, E. (2013). A cma-es super-fit scheme for the re-sampled inheritance search. In *2013 IEEE Congress on Evolutionary Computation* (pp. 1123–1130). IEEE. doi:10.1109/CEC.2013.6557692.
- Coxeter, H. S. M. (1971). Inversive geometry. *Educational Studies in Mathematics*, 3, 310–321. doi:10.1007/978-1-4612-5648-9_27.
- 635 Dang, D.-C., & Lehre, P. K. (2016). Self-adaptation of mutation rates in non-elitist populations. In *International Conference on Parallel Problem Solving from Nature* (pp. 803–813). Springer.
- De Jong, K. A. (2016). *Evolutionary Computation: A Unified Approach*. A Bradford Book. Cambridge, MA, USA: MIT Press.
- 640 Dorst, L., Fontijne, D., & Mann, S. (2010). *Geometric algebra for computer science: an object-oriented approach to geometry*. Elsevier.
- El-Abd, M. (2013). Testing a particle swarm optimization and artificial bee colony hybrid algorithm on the CEC13 benchmarks. In *2013 IEEE Congress on Evolutionary Computation* (pp. 2215–2220). IEEE. doi:10.1109/CEC.2013.6557832.
- 645 Elsayed, S. M., Sarker, R. A., & Essam, D. L. (2013). A genetic algorithm for solving the CEC'2013 competition problems on real-parameter optimization. In *2013 IEEE Congress on Evolutionary Computation* (pp. 356–360). IEEE. doi:10.1109/CEC.2013.6557591.
- 650 Glasmachers, T., Schaul, T., Yi, S., Wierstra, D., & Schmidhuber, J. (2010). Exponential natural evolution strategies. In *Proceedings of the 12th Annual Conference on Genetic and Evolutionary Computation* (pp. 393–400). New York, NY, USA: ACM. doi:10.1145/1830483.1830557.
- Guan, Y., Yang, L., & Sheng, W. (2017). Population control in evolutionary algorithms: Review and comparison. In *International Conference on Bio-Inspired Computing: Theories and Applications* (pp. 161–174). Springer.
- 655 Hansen, N., & Ostermeier, A. (1996). Adapting arbitrary normal mutation distributions in evolution strategies: the covariance matrix adaptation. In *Proceedings of IEEE International Conference on Evolutionary Computation* (pp. 312–317). IEEE. doi:10.1109/ICEC.1996.542381.
- Haupt, S. E. (2009). Environmental optimization: Applications of genetic algorithms. In S. E. Haupt, A. Pasini, & C. Marzban (Eds.), *Artificial Intelligence Methods in the Environmental Sciences* (pp. 379–395). Dordrecht: Springer, Netherlands. doi:10.1007/978-1-4020-9119-3_18.
- 660 Hayslett, H. (1967). *Statistics Made Simple*. Made simple books. Garden City, NY: Doubleday.

- Hitczenko, P., & Kwapień, S. (1994). On the rademacher series. In J. Hoffmann-Jørgensen, J. Kuelbs, & M. B. Marcus (Eds.), *Probability in Banach Spaces, 9* (pp. 31–36). Boston, MA: Birkhäuser Boston volume 35 of *Progress in Probability*. doi:10.1007/978-1-4612-0253-0_2.
- Jansen, T. (2013). *Analyzing Evolutionary Algorithms: The Computer Science Perspective*. Natural Computing Series. Springer Publishing Company, Incorporated. doi:10.1007/978-3-642-17339-4.
- Kartite, J., & Cherkaoui, M. (2016). Optimization of hybrid renewable energy power systems using evolutionary algorithms. In *5th International Conference on Systems and Control (ICSC)* (pp. 383–388). IEEE. doi:10.1109/ICoSC.2016.7507059.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization (PSO). In *Proceedings of ICNN'95, International Conference on Neural Networks* (pp. 1942–1948). IEEE volume 4. doi:10.1109/ICNN.1995.488968.
- Korošec, P., & Šilc, J. (2013). The continuous differential ant-stigmergy algorithm applied on real-parameter single objective optimization problems. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 1658–1663). IEEE. doi:10.1109/CEC.2013.6557760.
- Larrañaga, P., & Lozano, J. A. (2001). *Estimation of distribution algorithms: a new tool for evolutionary computation*, volume 2 of *Genetic Algorithms and Evolutionary Computation*. Springer Science & Business Media. doi:10.1007/978-1-4615-1539-5.
- Li, Y., Peng, C., Wang, Y., & Jiao, L. (2018). Optimization based on nonlinear transformation in decision space. *Soft Computing*, 23, 1–20. doi:10.1007/s00500-018-3209-7.
- Liang, J., Qu, B., Suganthan, P., & Hernández-Díaz, A. G. (2013). *Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization*. Computational Intelligence Laboratory, Nanyang Technological University, Singapore.
- Liao, T., & Sttzle, T. (2013). Benchmark results for a simple hybrid algorithm on the CEC 2013 benchmark set for real-parameter optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 1938–1944). IEEE. doi:10.1109/CEC.2013.6557796.
- Lutton, E., Perrot, N., & Tonda, A. (2016). *Evolutionary Algorithms for Food Science and Technology, Volume 7*. Computer Engineering Series, Meta-heuristics Set. Wiley. doi:10.1002/9781119136828.
- Mavrovouniotis, M., Li, C., & Yang, S. (2017). A survey of swarm intelligence for dynamic optimization: Algorithms and applications. *Swarm and Evolutionary Computation*, 33, 1 – 17. doi:10.1016/j.swevo.2016.12.005.

- Miranda, E. R., & Biles, J. A. (2007). *Evolutionary Computer Music*. Berlin, Heidelberg: Springer-Verlag London. doi:10.1007/978-1-84628-600-1.
- Montgomery-Smith, S. (1990). The distribution of rademacher sums. *Proceedings of the American Mathematical Society*, 109, 517–522. doi:10.2307/2048015.
- Moraglio, A. (2007). *Towards a Geometric Unification of Evolutionary Algorithms*. Ph.D. thesis University of Essex, United Kingdom.
- Moraglio, A., Di Chio, C., & Poli, R. (2007). Geometric particle swarm optimisation. In *European conference on genetic programming* (pp. 125–136). Springer, Berlin, Heidelberg. doi:10.1007/978-3-540-71605-1_12.
- Moraglio, A., & Johnson, C. G. (2010). Geometric generalization of the nelder-mead algorithm. In *European Conference on Evolutionary Computation in Combinatorial Optimization* (pp. 190–201). Springer, Berlin, Heidelberg. doi:10.1007/978-3-642-12139-5_17.
- Moraglio, A., & Poli, R. (2004). Topological interpretation of crossover. In *Genetic and Evolutionary Computation Conference* (pp. 1377–1388). Springer.
- Moraglio, A., & Togelius, J. (2009a). Geometric differential evolution. In *Proceedings of the 11th Annual conference on Genetic and evolutionary computation* (pp. 1705–1712). ACM.
- Moraglio, A., & Togelius, J. (2009b). Inertial geometric particle swarm optimization. In *2009 IEEE Congress on Evolutionary Computation* (pp. 1973–1980). IEEE.
- Moraglio, A., Togelius, J., & Lucas, S. (2006). Product geometric crossover for the sudoku puzzle. In *Proceedings of the IEEE International Conference on Evolutionary Computation* (pp. 470–476). IEEE. doi:10.1109/CEC.2006.1688347.
- Moschopoulos, P., & Canada, W. (1984). The distribution function of a linear combination of chi-squares. *Computers & Mathematics with Applications*, 10, 383–386. doi:10.1016/0898-1221(84)90066-X.
- Moschopoulos, P. G. (1985). The distribution of the sum of independent gamma random variables. *Annals of the Institute of Statistical Mathematics*, 37, 541–544. doi:10.1007/BF02481123.
- Nepomuceno, F. V., & Engelbrecht, A. P. (2013). A self-adaptive heterogeneous pso for real-parameter optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 361–368). IEEE. doi:10.1109/CEC.2013.6557592.

- Ollivier, Y., Arnold, L., Auger, A., & Hansen, N. (2017). Information-geometric optimization algorithms: A unifying picture via invariance principles. *J. Mach. Learn. Res.*, 18, 564–628. URL: <http://dl.acm.org/citation.cfm?id=3122009.3122027>.
- Papa, G., & ilc, J. (2013). The parameter-less evolutionary search for real-parameter single objective optimization. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 1131–1137). IEEE. doi:10.1109/CEC.2013.6557693.
- Patel, J. K., & Read, C. B. (1996). *Handbook of the normal distribution*. Statistics: A Series of Textbooks and Monographs (2nd ed.). CRC Press.
- Price, K. V. (2013). Differential evolution. In *Handbook of Optimization* (pp. 187–214). Springer.
- Rothlauf, F. (2006). *Representations for Genetic and Evolutionary Algorithms*. (2nd ed.). Springer-Verlag Berlin Heidelberg. doi:10.1007/3-540-32444-5.
- Schwefel, H.-P. P. (1993). *Evolution and Optimum Seeking: The Sixth Generation*. New York, NY, USA: John Wiley & Sons, Inc.
- Segovia-Dominguez, I., & Hernandez-Aguirre, A. (2015). An estimation of distribution algorithm based on the natural gradient and the boltzmann distribution. In *Proceedings of the Conference on Genetic and Evolutionary Computation* (pp. 527–534). New York, NY, USA: ACM. doi:10.1145/2739480.2754803.
- Serrano-Rubio, J. P. (2015). *Pattern Recognition and Global Optimization with Geometric Algebra*. Ph.D. thesis Center for Research in Mathematics, Mexico.
- Serrano-Rubio, J. P., Hernández-Aguirre, A., & Herrera-Guzmán, R. (2013). Training multilayer perceptron by conformal geometric evolutionary algorithm. In M. Emmerich, A. Deutz, O. Schuetze, T. Bäck, E. Tantar, A.-A. Tantar, P. D. Moral, P. Legrand, P. Bouvry, & C. A. Coello (Eds.), *A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation IV* (pp. 31–45). Springer, Heidelberg. doi:10.1007/978-3-319-01128-8_3.
- Serrano-Rubio, J. P., Hernández-Aguirre, A., & Herrera-Guzmán, R. (2014). Function optimization in conformal space by using spherical inversions and reflections. In A. L. Bazzan, & K. Pichara (Eds.), *Advances in Artificial Intelligence – IBERAMIA 2014* (pp. 418–429). Springer International Publishing. doi:10.1007/978-3-319-12027-0_34.
- Serrano-Rubio, J. P., Hernández-Aguirre, A., & Herrera-Guzmán, R. (2018). An evolutionary algorithm using spherical inversions. *Soft Computing*, 22, 1993–2014. doi:10.1007/s00500-016-2461-y.
- Smith, S., & Cagnoni, S. (2011). *Genetic and Evolutionary Computation: Medical Applications*. Wiley Publishing.

- Tanabe, R., & Fukunaga, A. (2013). Evaluating the performance of shade on cec 2013 benchmark problems. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 1952–1959). IEEE. doi:10.1109/CEC.2013.6557798.
- 785 Tanabe, R., & Fukunaga, A. S. (2014). Improving the search performance of shade using linear population size reduction. In *2014 IEEE congress on evolutionary computation (CEC)* (pp. 1658–1665). IEEE.
- Ter-Sarkisov, A., & Marsland, S. (2011). Convergence properties of $(\mu + \lambda)$ evolutionary algorithms. In *Twenty-Fifth AAAI Conference on Artificial Intelligence*.
- 790 Weise, T., Wu, Y., Chiong, R., Tang, K., & Lässig, J. (2016). Global versus local search: the impact of population sizes on evolutionary algorithm performance. *Journal of Global Optimization*, 66, 511–534.
- Yang, X.-S. (2010). *Engineering Optimization: An Introduction with Meta-heuristic Applications*. (1st ed.). Wiley Publishing.
- 795

Declaration of interests**July 3, 2019**

Dear Editor,

In order to ensure compliance with the Journal of Expert System with Applications policy I, as corresponding author on behalf of all the authors, state the following:

- There are no known conflicts of interest associated with the article entitled “Geometric Probabilistic Evolutionary Algorithm”.
- The manuscript has been read and approved by all named authors and that there are no other persons who satisfied the criteria for authorship but are not listed. Additionally, the order of authors listed in the manuscript has been approved by all of us.
- We have given due consideration to the protection of intellectual property associated with this work and there are no impediments to publication, including the timing of publication, with respect to intellectual property.

Additionally, the declaration of interest statement, downloaded from the Journal’s homepage, is added below.

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

None.

Sincerely,

Ignacio Segovia-Dominguez (corresponding author)

Credit Author Statement

July 2, 2019

Dear Editor,

This document outlines all authors' individual contributions to the paper entitled **Geometric Probabilistic Evolutionary Algorithm**, using the Contributor Roles Taxonomy.

Ignacio Segovia-Domínguez.

Conceptualization, Software, Formal Analysis, Investigation, Validation, Writing - Original Draft

Rafael Herrera-Guzmán.

Conceptualization, Formal Analysis, Funding Acquisition, Writing - Review & Editing, Supervision

Juan Pablo Serrano-Rubio.

Investigation, Validation, Visualization, Writing - Review & Editing

Arturo Hernández-Aguirre.

Resources, Writing - Review & Editing, Supervision

Yours sincerely,

Ignacio Segovia-Domínguez (corresponding author)