Use of the q-Gaussian Mutation in Evolutionary Algorithms

Renato Tinós · Shengxiang Yang

Abstract This paper proposes the use of the q-Gaussian mutation with self-adaptation of the shape of the mutation distribution in evolutionary algorithms. The shape of the q-Gaussian mutation distribution is controlled by a real parameter q. In the proposed method, the real parameter q of the q-Gaussian mutation is encoded in the chromosome of individuals and hence is allowed to evolve during the evolutionary process. In order to test the new mutation operator, evolution strategy and evolutionary programming algorithms with self-adapted q-Gaussian mutation generated from anisotropic and isotropic distributions are presented. The theoretical analysis of the q-Gaussian mutation is also provided. In the experimental study, the q-Gaussian mutation is compared to Gaussian and Cauchy mutations in the optimization of a set of test functions. Experimental results show the efficiency of the proposed method of self-adapting the mutation distribution in evolutionary algorithms.

Keywords Evolutionary algorithms · q-Gaussian distribution · self-adaptation · evolutionary programming · mutation distribution

1 Introduction

Evolutionary algorithms (EAs), which are a class of stochastic search algorithms inspired by the principles

Renato Tinós
Department of Physics and Mathematics, FFCLRP, University of São Paulo (USP), 14040-901, Ribeirão Preto, SP, Brazil
E-mail: rtinos@ffclrp.usp.br

Shengxiang Yang
Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex UB8 3PH, U.K.
E-mail: shengxiang.yang@brunel.ac.uk

of natural evolution, have been successfully applied to a large number of optimization problems. Many of these problems are continuous optimization problems and, as a consequence, several interesting EA variants for real-valued optimization have been investigated, as evolution strategy (ES) with adaptive enconding [15], differential evolution (DE) [6,27,30], real-coded genetic algorithms [16,11], memetic algorithms [20,?,25], and hybrid approaches [42].

Stochastic search algorithms, like EAs, differ from other optimization algorithms by using random samples to generate new candidate solutions. Traditionally, when EAs are applied in real-valued optimization, new candidate solutions are generated by mutation using multivariate samples taken from Gaussian distributions with zero mean [5]. The isotropic Gaussian distribution, which has a finite second moment, maximizes the Boltzmann-Gibbs entropy (and the differential entropy, i.e., the extension of the Shannon's concept of information entropy to the continuous case) in the unconstrained real-valued search space [36] and, in this way, does not favor any direction in the search space. In this way, the generation of new candidate solutions by mutation does not require the knowledge of any information about the geometry of the search space.

However, in recent years, researchers have proposed the use of distributions with longer tails and an infinite second moment in EAs. For example, in the fast evolutionary programming (FEP) [45], the Cauchy distribution is employed, while in the evolutionary programming (EP) with Lévy mutation (LEP) [18], mutation based on the Lévy distribution is used. The Lévy distribution is a class of probability distributions with an infinite second moment, which includes the Cauchy distribution, and allows to control the tail of the distribution by changing a scalar parameter α.

The use of mutation taken from heavy tail distributions implies jumps of scale-free sizes, allowing to reach distant regions of the search space faster. This property can be interesting when EAs are applied to multimodal problems or dynamic optimization problems as it can allow the population to escape faster from local optima. However, some controversy about the benefits of the use of distributions with heavy tails in EAs have appeared in recent years [14]. For example, some researchers argue that most of the proposed algorithms use heavy tail distributions that are anisotropic, i.e., where some directions of the mutation are privileged in the search space [26]. Moreover, in some fitness landscapes, it can be difficult to reach a fair region of the search space from a long jump as the probability to reach a point with a worse fitness is generally much larger for a long jump [14].

In this paper, self-adaptation is employed to control the mutation distribution. In this way, the choice of a mutation distribution for a given problem and, at a given moment of the evolutionary process, is made by the algorithm, and not by the programmer. In the proposed algorithm, a real parameter that defines the shape of the distribution employed by the mutation operator is encoded in the chromosome of the individuals and is allowed to evolve. For this purpose, the q-Gaussian probability density function [36] derived from the Tsallis generalized entropy [39] is employed. The q-Gaussian distribution allows to control the shape of the distribution by setting a real parameter q and can reproduce either finite second moment distributions, like the Gaussian distribution, or infinite second moment distributions, like the Cauchy distribution.

The rest of this paper is organized as follows. Related work and the main contributions of the paper are presented in Section 2. The q-Gaussian distribution is briefly introduced in Section 3. In Section 4, self-adaptation of the mutation distribution in EAs is proposed. Evolutionary programming algorithms with mutations generated from anisotropic and isotropic distributions are presented in Section 5. In Section 6, the theorical analysis of the q-Gaussian mutation is provided. In Section 7, the q-Gaussian mutation is used in a (μ, λ)-ES with restart. The experimental study based on a suite of benchmark test functions is presented in Section 8. In the experiments with EP algorithms, Gaussian, Cauchy and q-Gaussian mutations generated from anisotropic and isotropic distributions are compared, while in the remaining experiments, a (μ, λ)-ES with q-Gaussian mutation and restart is compared to other continuous optimization EAs. Finally, Section 9 concludes this paper with discussions on the relevant future work.

2 Related work

In this paper, instead of only controlling the mutation strength parameter that defines the spread of a fixed mutation distribution like in ESs and EP, the shape of the mutation distribution is also controlled during the evolutionary process. There are three main classes of parameter control in EAs [9]: deterministic, where the parameters are changed by deterministic rules; adaptive, where feedback from the optimization process is employed for parameter control; and self-adaptive, where parameters of the EA are encoded in the chromosome and allowed to evolve. The use of self-adaptation to define the mutation strength parameters is usual in EP and ESs. Here, self-adaptation is used to control the parameter q, that allows to modify the shape of the q-Gaussian distribution employed to generate random mutations.

The use of mutations generated from q-Gaussian distributions in EAs is not new [17,23]. However, in such algorithms, like in most other algorithms that uses heavy tail distributions, new candidate solutions are produced by generating random deviates for each coordinate of the individual, which implies anisotropic distributions where large steps are generated close to the coordinate axes. Moreover, in such algorithms, the parameter q, which defines the shape of the distribution, is fixed during the searching process or starts with a large value and decreases during the searching process, like the temperature control in simulated annealing. That is, the control is deterministic. In [18], the authors proposed two schemes for LEP: in the first scheme, all offspring are generated from a distribution with a fixed α, and in the second, each parent generates five offspring, each of which is generated from a distribution with a different pre-defined value of α. All individuals in LEP use the same pre-fixed values of α during the whole evolutionary process.

Control of the mutation probability density function is not new too. In [7], the mutation probability density function is generated from a histogram with 101 bars representing the values of probability between a range of interest in a one-dimensional space. Self-adaptation is employed to control the heights of the bars, allowing to change the shape of the mutation distribution during the running of the EA [3]. Experiments indicated the formation of histograms with a peak in the center, suggesting that Gaussian and Cauchy distributions are good candidates as mutation distributions. However, the use of histograms is not suitable when the number of dimensions of the search space is high because the required number of histograms exponentially increases with the dimension of the search space. As

pointed in [3], the control of the mutation probability density function by a few control parameters would be more interesting.

In [8], four types of mutation can occurs: Cauchy, Lévy, and two types of Gaussian mutation (i.e., the standard Gaussian mutation and the single-point mutation, where, in each occurrence, only one dimension is changed). A four-string vector containing the probabilities of choosing each type of mutation is added to the individual and is modified according to the performance of each type of mutation.

The solution presented in this paper to control the shape of the mutation distribution employs only one parameter for each individual: the q-Gaussian distribution parameter q. Differently from the strategy used in [8], the control of the parameter q allows to smoothly and continuously change the shape of the distribution, as q is a real parameter and a small change in its value causes a small change in the shape of the mutation distribution. In this way, the main contributions of this paper are: 1) The use of the q-Gaussian mutation generated from anisotropic and isotropic distributions are proposed and compared in EP; 2) Self-adaptation is used to control the parameter q, which allows changing the shape of the distribution during the solving process; 3) Gaussian, Cauchy, and q-Gaussian mutations generated from anisotropic and isotropic distributions are compared in a series of experiments based on a suite of benchmark test functions; 4) An ES with q-Gaussian mutation is proposed and compared to other continuous optimization EAs.

3 The q-Gaussian distribution

One of the most interesting properties of the Gaussian distribution is that it maximizes, under certain constraints, the entropy in the form

$$S = \int_{-\infty}^{+\infty} p(x) \ln \big(p(x)\big) dx, \tag{1}$$

where $p(x)$ is the distribution density function. This entropy is known as the Boltzman-Gibbs entropy. While the Gaussian distribution is an attractor for independent systems with a finite second moment, it does not represent well correlated systems with an infinite second moment [36]. In this concern, Tsallis [39] proposed a generalized entropy form as follows:

$$S_q = \frac{1 - \int_{-\infty}^{+\infty} p(x)^q dx}{q - 1}, \tag{2}$$

where $q \in \mathbb{R}$. Eq. (2) recovers the entropy form given by Eq. (1) in the limit $q \to 1$. The q-Gaussian distribution arises when maximizing the generalized entropy form

given by Eq. (2). The q-Gaussian distribution has interesting properties. The parameter q controls the shape of the q-Gaussian distribution. The second order moment is finite for $q < 5/3$ and the q-Gaussian distribution reproduces the usual Gaussian distribution when $q \to 1$. When $q < 1$, the q-Gaussian distribution has a compact form, and decays asymptotically as a power law for $1 < q < 3$. When $q = 2$, the q-Gaussian distribution reproduces the Cauchy distribution, while for $q = (3 + d)/(1 + d)$ where $0 < d < \infty$, it becomes a Student's t-distribution with d degrees of freedom [34].

When $-\infty < q < 3$, the q-Gaussian distribution density [36] is given by

$$p_{q(\bar{\mu}_q, \bar{\sigma}_q)}(x) = \frac{\sqrt{B_q}}{A_q} \, e_q^{-B_q(x-\bar{\mu}_q)^2}, \tag{3}$$

where $\bar{\mu}_q$ and $\bar{\sigma}_q$ are the q-mean and the q-variance respectively, A_q is the normalization factor, B_q controls the width of the q-Gaussian distribution, and e_q^{-y} is the q-exponential function of $-y$ defined as follows:

$$e_q^{-y} \equiv \begin{cases} \left(1 + (q-1)y\right)^{-\frac{1}{q-1}}, & \text{if } 1 + (q-1)y \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

When $q \to 1$, the limit of the q-exponential function of $-y$, if we write $z = (q-1)y$, is given by

$$\lim_{q \to 1} e_q^{-y} = \lim_{z \to 0} \left(\left(1 + z\right)^{\frac{1}{z}} \right)^{-y}. \tag{5}$$

The limit of the function $(1 + z)^{\frac{1}{z}}$ is well know and converges to e when $z \to 0$. Thus, we have

$$\lim_{q \to 1} e_q^{-y} = e^{-y} \tag{6}$$

i.e., the q-gausssian exponential converges to the exponencial when $q \to 1$.

In Eq. (3), the q-mean $\bar{\mu}_q$ and the q-variance $\bar{\sigma}_q$ [36] are respectively defined as follows:

$$\bar{\mu}_q \equiv \frac{\int x p(x)^q dx}{\int p(x)^q dx}, \tag{7}$$

$$\bar{\sigma}_q^2 \equiv \frac{\int (x - \bar{\mu}_q)^2 p(x)^q dx}{\int p(x)^q dx}, \tag{8}$$

and respectively reduce to the usual mean and variance when $q \to 1$.

In Eq. (3), the normalization factor A_q is given by $A_q = \int_{-\infty}^{+\infty} e_q^{-(x-\bar{\mu}_q)^2} dx$ [41] and B_q is given by

$$B_q = \left((3 - q)\bar{\sigma}_q^2 \right)^{-1}. \tag{9}$$

A random variable x taken from a q-Gaussian distribution with q-mean $\bar{\mu}_q$ and q-variance $\bar{\sigma}_q^2$ is here denoted by $x \sim \mathcal{N}_q(\bar{\mu}_q, \bar{\sigma}_q)$. In this paper, the generalized Box-Müller method proposed in [36], which is very simple (see its pseudo-code in [36]) and allows to generate
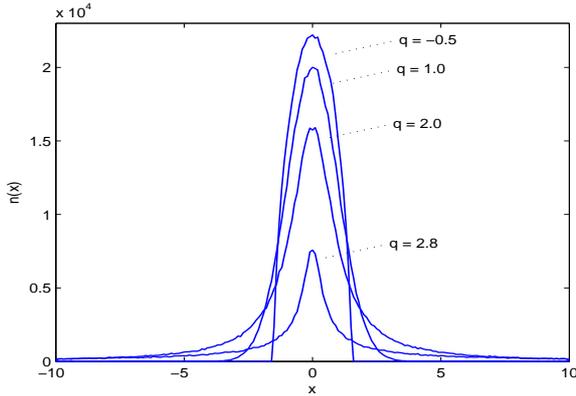
Fig. 1 Empirical distribution for 500000 observations of a q-Gaussian random variable $x \sim \mathcal{N}_q(0,1)$ for: $q = -0.5$, $q = 1.0$ (Gaussian), $q = 2.0$ (Cauchy), and $q = 2.8$. Only the values for x between -10 and $+10$ are shown.

samples from q-Gaussian distributions for $-\infty < q < 3$, is employed to generate q-Gaussian random variables $x \sim \mathcal{N}_q(0,1)$.

Fig. 1 presents the empirical q-Gaussian distribution for random variables $x \sim \mathcal{N}_q(0,1)$ with different values of q. It can be observed that larger values of q result in longer tails of the q-Gaussian distribution.

4 Self-adaptation of the mutation distribution

In an m-dimensional real-valued search space, a new candidate solution is generated by the EA's mutation operator from individual $\mathbf{x}_i$ as follows:

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i + \mathbf{C}\,\mathbf{z}, \tag{10}$$

where $i = 1, \ldots, \mu$, $\mathbf{z}$ is an m-dimensional random vector generated from a given multivariate distribution with zero mean, and $\mathbf{C}$ is the matrix which defines the mutation strength in each coordinate $j = 1, \ldots, m$. In the most simple case,

$$\mathbf{C} = \sigma\mathbf{I}, \tag{11}$$

where $\mathbf{I}$ is the identity matrix and the unique parameter, σ, defines the mutation strength for all components of $\mathbf{x}_i$. There are some cases, however, where it is interesting to define one different parameter $\sigma(j)$ for each component of $\mathbf{x}_i$. In this way, we have

$$\mathbf{C} = \mathrm{diag}(\boldsymbol{\sigma}^{\mathrm{T}}). \tag{12}$$

That is, $\mathbf{C}$ is a diagonal matrix with the main diagonal composed by the elements of vector $\boldsymbol{\sigma} = [\sigma(1)\ \sigma(2)\ldots\sigma(m)]^{\mathrm{T}}$. In the most general situation, e.g., in the covariance matrix adaptation ES (CMA-ES) [13], $\mathbf{C}$ is a matrix with elements indicating the correlation between the components of $\mathbf{z}$.

In general, when self-adaptation is used in the standard EP [45] and in ES [5], the mutation strength parameter for each offspring $i = 1, \ldots, \mu$ of the population is multiplicatively updated. If all elements of the mutation parameter vector are equal (i.e., $\sigma_i(j) = \sigma_i$ for $j = 1, \ldots, m$, see Eq. (11)), then the updated value of the mutation parameter is given by

$$\tilde{\sigma}_i = \sigma_i e^{\tau_b \mathcal{N}(0,1)}, \tag{13}$$

where τ_b denotes the standard deviation of the Gaussian distribution used to generate the change in σ_i. If each element of the vector $\mathbf{x}_i$ has an individual mutation strength parameter, as shown in Eq. (12), $\sigma_i(j)$ is updated according to the following formula

$$\tilde{\sigma}_i(j) = \sigma_i(j) e^{\tau_b \mathcal{N}(0,1)_i + \tau_c \mathcal{N}(0,1)}, \tag{14}$$

where τ_b denotes the standard deviation of the Gaussian distribution used to generate the random deviate $\mathcal{N}(0,1)_i$, which is common for all elements of the vector $\mathbf{x}_i$, and τ_c is the standard deviation of the Gaussian distribution used to generate the separated random deviate $\mathcal{N}(0,1)$ for each element $j = 1, \ldots, m$.

In EAs, the use of the Gaussian distribution is generally employed to generate the m-dimensional vector $\mathbf{z}$ [5]. Here, an m-dimensional random vector generated from the Gaussian distribution is denoted by $\mathbf{z} \sim \mathcal{N}^m$. A Gaussian random vector $\mathbf{z} \sim \mathcal{N}^m$ is generated by sampling m independent Gaussian variables $\mathcal{N}(0,1)$. It is important to observe that when the same procedure is adopted to generate multivariate random samples with a heavy tail distribution, some directions in the search space are much more explored than others, i.e., the distribution is highly anisotropic.

To the best of the authors' knowledge, all stochastic search algorithms with the q-Gaussian mutation, like the Generalized Simulated Annealing [40] and the Generalized Genetic Algorithm [23], make use of anisotropic q-Gaussian distributions generated by sampling m independent q-Gaussian variables. Most mutation operators for EAs that are based on heavy tail distributions, e.g., in Fast Evolution Strategies [44], FEP [45], and LEP [18], make use of anisotropic distributions generated by sampling independent random variables too. The use of random variables generated by sampling independent random variables taken from a heavy tail distribution is very interesting for optimization problems with separable functions, as most of the large steps occur close to the coordinate axes [26,36] and the optimization can be solved by m one-dimensional optimization processes parallel to the coordinate axes. However, the performance of the optimization process can be strongly affected for some non-separable functions.

In this paper, we investigate the use of two multivariate q-Gaussian distributions, the anisotropic q-

Gaussian distribution generated by sampling independent q-Gaussian random variables (Section 5.1) and the q-Gaussian distribution generated from isotropic distributions (Section 5.2), to produce new candidate solutions in EAs. As mentioned earlier, the use of the q-Gaussian distribution allows us to reproduce different distributions by changing only one real parameter q.

We propose to self-adapt the parameter q, which defines the shape of the distribution. Based on the mutation strength self-adaptation [5], we propose to multiplicatively update the parameter q in individual i as follows:

$$\tilde{q}_i = q_i e^{\tau_q \mathcal{N}(0,1)}, \tag{15}$$

where τ_q denotes the standard deviation of the Gaussian distribution. In this way, different distributions can be reproduced during the evolutionary process. However, it is not possible to identify the separated influence of a change in $\boldsymbol{\sigma}_i$ or q_i in the fitness of individual i if both mutation strength vector and parameter q are mutated in the same generation for individual i. For example, a beneficial mutation in $\boldsymbol{\sigma}_i$ can be masked in the fitness of individual i if the parameter q is mutated to a bad value in the same generation. Here, $\boldsymbol{\sigma}_i$ and q_i are not mutated together (i.e., in the same generation) in each individual. The mutation strength vector $\boldsymbol{\sigma}_i$ is updated for individual i in each generation if a uniform random number in the range $[0, 1]$ is equal to or larger than a real parameter $r_q \in [0, 1]$. Otherwise, the value of q_i is updated.

5 Evolutionary programming algorithms with q-Gaussian mutation

In order to test the proposed ideas, an EP algorithm with the q-Gaussian mutation, called *qGEP* for the anisotropic version and *IqGEP* for the version generated from isotropic distributions, is proposed and presented in Algorithm 1. EP was selected to test the q-Gaussian mutation because it only uses mutation as a transformation operator, which makes it easier to compare the q-Gaussian mutation to Cauchy and Gaussian mutation.

There are two main differences of the EP algorithm presented in Algorithm 1 from the Gaussian EP, FEP [45], and LEP [18]. Firstly, in the proposed algorithm, the q-Gaussian mutation is employed (step 10) instead of the Gaussian (Gaussian EP), Cauchy (FEP), or Lévy (LEP) mutation. Secondly, a procedure to adapt the q parameter is adopted in the proposed algorithm, i.e., steps 5 to 9 in Algorithm 1. For Gaussian EP, FEP, and LEP, steps 5, 7, 8, and 9 in Algorithm 1 are removed.

Algorithm 1 EP with q-Gaussian mutation (qGEP and IqGEP)

1: Initialize the population composed of individuals $(\mathbf{x}_i, \boldsymbol{\sigma}_i, q_i)$ for $i = 1, \ldots, \mu$
2: Evaluate the individuals $(\mathbf{x}_i, \boldsymbol{\sigma}_i, q_i)$ for $i = 1, \ldots, \mu$
3: **while** (stop criteria are not satisfied) **do**
4:     **for** $i \leftarrow 1$ to μ **do**
5:         **if** $rand(0, 1) \geq r_q$ **then**
6:             Update the mutation strength parameter $\tilde{\sigma}_i(j)$ for $j = 1$ to m. Use Eq. (14) for qGEP or Eq. (13) for IqGEP. In the last case, $\tilde{\sigma}_i(j) = \tilde{\sigma}_i$ for $j = 1$ to m
7:         **else**
8:             Update the parameter $\tilde{q}_i$ according to Eq. (15)
9:         **end if**
10:         $\tilde{\mathbf{x}}_i \leftarrow \mathbf{x}_i + \mathbf{C}_i \, \mathbf{z}$, where $\mathbf{z}$ is a q-Gaussian random vector generated from anisotropic (qGEP - Section 5.1) or isotropic (IqGEP - Section 5.2) distributions with parameter $\tilde{q}_i$ and $\mathbf{C}_i = \text{diag}(\tilde{\boldsymbol{\sigma}}_i^{\text{T}})$ (qGEP) or $\mathbf{C}_i = \tilde{\sigma}_i \mathbf{I}$ (IqGEP)
11:     **end for**
12:     Evaluate the offspring $(\tilde{\mathbf{x}}_i, \tilde{\boldsymbol{\sigma}}_i, \tilde{q}_i)$ for $i = 1, \ldots, \mu$
13:     Compute the winning function [18] of the population composed of μ parents and μ offspring
14:     Select, to compose the new population with individuals $(\mathbf{x}_i, \boldsymbol{\sigma}_i, q_i)$, the μ individuals with the largest winning function from the population composed of μ parents and μ offspring
15: **end while**

5.1 EP with q-Gaussian mutation generated from anisotropic distribution

In an EA with q-Gaussian mutation generated from anisotropic distribution, the vector $\mathbf{z}$ in Eq. (10) is created by sampling m independent q-Gaussian random variables. Each element $z(j)$, $j = 1, \ldots, m$, of the random mutation vector $\mathbf{z}$ is generated as follows:

$$z(j) \sim \mathcal{N}_q(0, 1). \tag{16}$$

Here, an m-dimensional random vector generated from the multivariate anisotropic q-Gaussian distribution is denoted by $\mathbf{z} \sim \mathcal{M}_q^m$.

In order to investigate the contribution of the scheme of self-adapting the parameter q, the qGEP algorithm is compared, in Section 8, to three other approaches where the parameter q is fixed, i.e. the shape of the mutation distribution is fixed during the evolutionary process. For fixed values of q, the q-Gaussian mutation can reproduce Gaussian and Cauchy mutations. Lévy mutation can be still reproduced, as Lévy and q-Gaussian distributions are related for some values of q [31]. In all approaches described here, Eq. (16) is employed to generate new candidate solutions. The approaches are defined as follows:

- Algorithm GEP: uses only one fixed parameter $q = 1.0$ for all individuals. That is, in Algorithm 1, the initial value of $\tilde{q}_i$ is equal to 1.0 for $i = 1, \ldots, \mu$
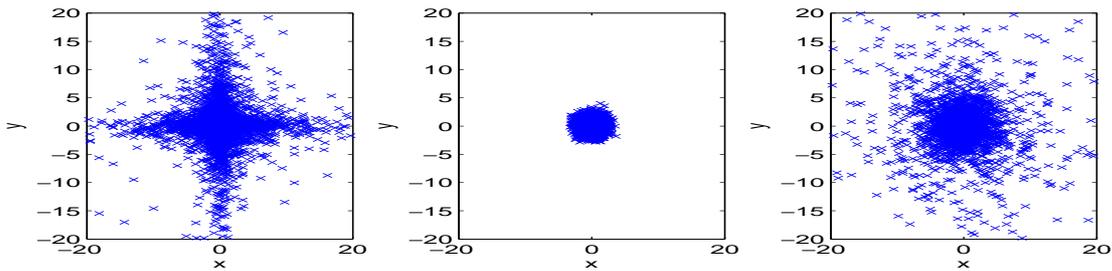
Fig. 2 Two-dimensional points from: random vector $\mathbf{z} \sim \mathcal{M}_q^m$ with anisotropic q-Gaussian distribution and $q = 2.0$ (left), random vector $\mathbf{z} \sim \mathcal{N}^m$ with Gaussian distribution (center), and q-Gaussian random vector $\mathbf{z} \sim \sqrt{2}\mathcal{N}_q^m$ generated from isotropic distribution and $q=2.0$ (right).

and $r_q = 0$. In this way, the Gaussian mutation generated from sampling m independent Gaussian random variables is reproduced.

– Algorithm CEP: uses only one fixed parameter $q = 2.0$ for all individuals. That is, in Algorithm 1, the initial value of $\tilde{q}_i$ is equal to 2.0 for $i = 1, \ldots, \mu$ and $r_q = 0$. In this way, the anisotropic Cauchy distribution is reproduced.

– Algorithm EP ($q = 1.5$): uses only one fixed parameter $q = 1.5$ for all individuals. That is, in Algorithm 1, the initial value of $\tilde{q}_i$ is equal to 1.5 for $i = 1, \ldots, \mu$ and $r_q = 0$.

– Algorithm qGEP: uses one changing q for each individual and $r_q > 0$, i.e., Eq. (15) is employed.

5.2 EP with q-Gaussian mutation generated from isotropic distribution

Here, the use of q-Gaussian mutations generated from isotropic distributions is proposed. In [26], a method to generate the Cauchy random mutation vector from an isotropic distribution was proposed. For this purpose, the random mutation vector is generated with: 1) a random direction uniformly distributed on the surface of the m-dimensional unit hypersphere, and 2) an Euclidean norm obtained from a Cauchy distribution.

Based on the works [26] and [36], we propose to generate the random mutation vector $\mathbf{z}$ as follows:

$$\mathbf{z} \sim r\mathbf{u}, \tag{17}$$

where $r \sim \mathcal{N}_q(0, 1)$, i.e., a random variable with the q-Gaussian distribution, and $\mathbf{u}$ is an uniform random vector obtained by sampling a random vector with a Gaussian distribution and normalizing it to length one, i.e., $\mathbf{u} = \mathbf{v}/\|\mathbf{v}\|$ where $\mathbf{v} \sim \mathcal{N}^m$ and $\|\mathbf{v}\|$ denotes the Euclidean norm of the vector $\mathbf{v}$. In this paper, an m-dimensional random vector generated from an isotropic distribution and step length given by a q-Gaussian distribution is denoted by $\mathbf{z} \sim \mathcal{N}_q^m$.

Figure 2 presents two-dimensional multivariate samples obtained from an anisotropic q-Gaussian distribution, a Gaussian distribution, and a q-Gaussian distribution generated as described in this section. It can be observed that, in the anisotropic q-Gaussian distribution, larger steps occur more often close to the coordinate axes. This effect is more evident in high dimensional spaces and/or for larger values of q for $q < 3$.

The EP algorithm with q-Gaussian mutation generated from isotropic distribution, called *IqGEP* here, is compared, in Section 8, to three other approaches where the parameter q is fixed. In all approaches, Eq. (17) is employed to generate the new candidate solutions. The approaches are defined as follows:

– Algorithm IGEP: uses only one fixed parameter $q = 1.0$ for all individuals. That is, in Algorithm 1, the initial value of $\tilde{q}_i$ is equal to 1.0 for $i = 1, \ldots, \mu$ and $r_q = 0$. In this way, the Gaussian mutation generated from an isotropic distribution and with step length given by the Gaussian distribution is reproduced.

– Algorithm ICEP: uses only one fixed parameter $q = 2.0$ for all individuals. That is, in Algorithm 1, the initial value of $\tilde{q}_i$ is equal to 2.0 for $i = 1, \ldots, \mu$ and $r_q = 0$. In this way, the Cauchy mutation generated from an isotropic distribution and with step length given by the Cauchy distribution is reproduced.

– Algorithm IEP ($q = 1.5$): uses only one fixed parameter $q = 1.5$ all individuals. That is, in Algorithm 1, the initial value of $\tilde{q}_i$ is equal to 1.5 for $i = 1, \ldots, \mu$ and $r_q = 0$.

– Algorithm IqGEP: uses one changing q for each individual and $r_q > 0$, i.e., Eq. (15) is employed.

6 Analysis of the q-Gaussian mutation

In this section, the impact of changing the mutation strength parameter σ and the q-Gaussian distribution parameter q on the probability of generating a jump σx,

where $x \sim \mathcal{N}_q(0,1)$, in the neighbourhood of a point x^* is analysed. The analyses presented here are similar to the analysis of Gaussian and Cauchy mutations presented in [45].

When $\bar{\mu}_q = 0$ and $\bar{\sigma}_q^2 = 1$, the q-Gaussian distribution density for $-\infty < q < 3$ (see Eq. (3)) is given by

$$p_q(x) = \frac{1}{\sqrt{3-q}A_q} \; e_q^{\frac{-x^2}{3-q}}, \tag{18}$$

where, considering $\left(1 + x^2(q-1)/\sqrt{3-q}\right) \geq 0$, the q-exponential is given by

$$e_q^{\frac{-x^2}{3-q}} = \left(1 + \frac{q-1}{3-q} \; x^2\right)^{\frac{1}{1-q}}. \tag{19}$$

We consider that the q-Gaussian mutation is applied in an EA in the one-dimensional case, i.e. the mutation produces a jump σx, where σ is the mutation strength parameter and $x \sim \mathcal{N}_q(0,1)$. For simplicity, we will consider $1 < q < 3$. The probability of reaching the neighbourhood of a point x^* from a jump σx, i.e., the probability that $x^* - \epsilon \leq \sigma x \leq x^* + \epsilon$, where $\epsilon > 0$ defines the neighbourhood size, is given by

$$P_q(|\sigma x - x^*| \leq \epsilon) = \int_{\frac{x^*-\epsilon}{\sigma}}^{\frac{x^*+\epsilon}{\sigma}} p_q(x)dx. \tag{20}$$

The mean value theorem for definite integrals states that there is a number δ ($0 < \delta < 2\epsilon$), at which the value of the integral given by Eq. (20) is equal to the difference between the limits of the integral multiplied by $p_q((x^*-\epsilon+\delta)/\sigma)$. In this way, Eq. (20) can be written as follows:

$$P_q(|\sigma x - x^*| \leq \epsilon) = \frac{2\epsilon}{\sigma} p_q\left(\frac{x^* - \epsilon + \delta}{\sigma}\right). \tag{21}$$

Substituting Eqs. (18) and (19) in Eq. (21), we obtain

$$P_q(|\sigma x - x^*| \leq \epsilon) = \frac{2\epsilon}{\sigma\sqrt{3-q}A_q}\left(1 + \frac{q-1}{3-q}\frac{c^2}{\sigma^2}\right)^{\frac{1}{1-q}}, \tag{22}$$

where $c = x^* - \epsilon + \delta$.

6.1 The impact of changing σ

Taking the derivative of Eq. (22) with respect to σ, we can write

$$\frac{\partial}{\partial\sigma}P_q(|\sigma x - x^*| \leq \epsilon) =$$
$$\frac{2\epsilon}{\sqrt{3-q}A_q}\frac{\partial}{\partial\sigma}\left(\frac{1}{\sigma}\left(1 + \frac{q-1}{3-q}\frac{c^2}{\sigma^2}\right)^{\frac{1}{1-q}}\right), \tag{23}$$

and, then,

$$\frac{\partial}{\partial\sigma}P_q(|\sigma x - x^*| \leq \epsilon) = \frac{2\epsilon}{\sqrt{3-q}A_q}\left(\frac{2q_a c^2}{(q-1)\sigma^4}\left(1 + q_a\frac{c^2}{\sigma^2}\right)^{\frac{q}{1-q}}\right.$$
$$\left. - \frac{1}{\sigma^2}\left(1 + q_a\frac{c^2}{\sigma^2}\right)^{\frac{1}{1-q}}\right), \tag{24}$$

where $q_a = (q-1)/(3-q)$. After some manipulation, we obtain

$$\frac{\partial}{\partial\sigma}P_q(|\sigma x - x^*| \leq \epsilon) =$$
$$\frac{2\epsilon}{\sqrt{3-q}A_q\sigma^2}\left(1 + q_a\frac{c^2}{\sigma^2}\right)^{\frac{1}{1-q}}\frac{c^2-\sigma^2}{q_a c^2+\sigma^2}. \tag{25}$$

From Eq. (25), we can write for $1 < q < 3$

$$\frac{\partial}{\partial\sigma}P_q(|\sigma x - x^*| \leq \epsilon) \begin{cases} > 0 \text{ if } |c| > \sigma \\ < 0 \text{ if } |c| < \sigma \end{cases}. \tag{26}$$

Eq. (26) states that an increase in the mutation strength σ results in an increase in the probability of reaching the point c, which is located in the neighbourhood of the point x^*, from a jump σx only if $|c| < \sigma$. In other words, an increase in the mutation strength is beneficial to reach the neighbourhood of a point x^* if it is distant ($|c| > \sigma$) from the current solution (before the mutation). A similar result was found for the Gaussian and Cauchy mutations [45]. The above analysis also states that the probability changing rate given by Eq. (25) depends on the value of q.

6.2 The impact of changing q

Taking the derivative of Eq. (22) with respect to q, we can write

$$\frac{\partial}{\partial q}P_q(|\sigma x - x^*| \leq \epsilon) = \frac{2\epsilon}{\sigma}\frac{\partial}{\partial q}\left(\frac{1}{\sqrt{3-q}A_q}y\right), \tag{27}$$

where

$$y = \left(1 + \frac{q-1}{3-q}\frac{c^2}{\sigma^2}\right)^{\frac{1}{1-q}} \tag{28}$$

is the q-exponential given by Eq. (19) at the point $x = c/\sigma$. We now analyse the derivative of the q-exponential y given by Eq. (28). Applying the natural logarithm in both sides of Eq. (28) and taking the derivative on q, we have

$$\frac{\partial y}{\partial q} = y\frac{\partial}{\partial q}\left(\frac{1}{1-q}\ln\left(1 + \frac{q-1}{3-q}\frac{c^2}{\sigma^2}\right)\right). \tag{29}$$

After some manipulation, we can write

$$\frac{\partial y}{\partial q} = \frac{1}{q-1}p^{\frac{1}{1-q}}\left(\frac{\ln(p)}{q-1} - \frac{2}{(3-q)^2}\frac{c^2}{\sigma^2}p^{-1}\right), \tag{30}$$

where

$$p = 1 + \frac{q-1}{3-q}\frac{c^2}{\sigma^2}, \tag{31}$$

i.e., $y = p^{\frac{1}{1-q}}$. In Eq. (30), $p \geq 1$ for $1 < q < 3$. In this way, we can write

$$\frac{\partial y}{\partial q} \begin{cases} > 0 \text{ if } a > b \\ < 0 \text{ if } a < b \end{cases}, \tag{32}$$
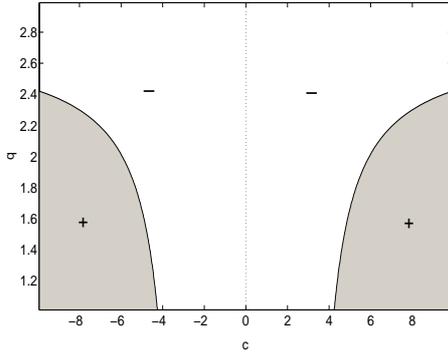
Fig. 3 Positive (gray) and negative (white) regions of the derivative of the q-exponential given by Eq. (28) for $\sigma = 3$.

where $a = \frac{\ln(p)}{q-1}$ and $b = \frac{2}{(3-q)^2}\frac{c^2}{\sigma^2}p^{-1}$. While $|a| > |b|$ for small values of $|c|$ (until the value of c where $a = b$), $|a| < |b|$ for larger values of $|c|$ as $\lim_{|c|\to\infty} a = +\infty$ and $\lim_{|c|\to\infty} b = \frac{2}{(3-q)(q-1)}$.

Figure 3 shows the regions where the derivative of the q-exponential given by Eq. (28) is positive or negative for $\sigma = 3$ and $|c| \leq 10$. When the derivative is positive, increasing (or decreasing) q by a small value implies increasing (or decreasing) the value of the q-exponential at the point c, while the opposite occurs for a negative derivative. It can be observed that the location c where the derivative changes its sign moves according to the value of q. The larger the value of q, the farther the locations of $|c|$ where the derivative changes its sign. Fig. 3 (and Eq. (30)) also suggests that values of q close to 3 are not interesting as the location where the derivative changes its sign is very far away from the current solution.

Using the derivative of the q-exponential given by Eq. (28), we can now write, after some manipulations, the derivative of Eq. (22) with respect to q (see Eq. (27))

$$\frac{\partial}{\partial q}P_q(|\sigma x - x^*| \leq \epsilon) =$$
$$\frac{2\epsilon p^{\frac{1}{1-q}}}{A_q\sigma\sqrt{3-q}(q-1)}\left(\frac{(q-1)(A_q - (3-q)A_q')}{(3-q)A_q} + a - b\right). \quad (33)$$

As the first term inside the parenthesis does not depend on c, the analysis is similar to that one presented before for the q-exponential given by Eq. (30). Eq. (33) indicates where a small change in the value of q is beneficial to increase the probability of reaching the neighbourhood of a point x^* from a jump σx. In other words, an increase in the value of q is beneficial to reach the neighbourhood of a point x^* if it is distant (at a location where the derivative given by Eq. (33) is positive) from the current solution (before the mutation). Otherwise, the value of q should be decreased.

Algorithm 2 Restart (μ,λ)-ES with q-Gaussian mutation (RqGES)

1: Initialize the population composed of individuals $(\mathbf{x}_k, \sigma_k, q_k)$ for $k = 1, \ldots, \mu$
2: Evaluate the individuals $(\mathbf{x}_k, \sigma_k, q_k)$ for $k = 1, \ldots, \mu$
3: **while** (stop criteria are not satisfied) **do**
4:   **if** the best fitness found in the last g_s generations did not change **then**
5:     **if** $\lambda \leq \lambda_{max}$ **then**
6:       $\lambda = 2\lambda$ and $\mu = 2\mu$.
7:     **end if**
8:     Restart the population
9:   **end if**
10:   Use recombination to generate the individuals $(\tilde{\mathbf{x}}_i, \tilde{\sigma}_i, \tilde{q}_i)$ for $i = 1, \ldots, \lambda$ from the individuals $(\mathbf{x}_k, \sigma_k, q_k)$ for $k = 1, \ldots, \mu$
11:   **for** $i \leftarrow 1$ to λ **do**
12:     **if** $rand(0,1) \geq r_q$ **then**
13:       Update the mutation strength parameter $\tilde{\sigma}_i$ using Eq. (13).
14:     **else**
15:       Update the parameter $\tilde{q}_i$ according to Eq. (15)
16:     **end if**
17:     $\tilde{\mathbf{x}}_i \leftarrow \tilde{\mathbf{x}}_i + \mathbf{C}_i \mathbf{z}$, where $\mathbf{z}$ is a q-Gaussian random vector generated from isotropic distribution with parameter $\tilde{q}_i$ and $\mathbf{C}_i = \tilde{\sigma}_i\mathbf{I}$
18:   **end for**
19:   Evaluate the offspring $(\tilde{\mathbf{x}}_i, \tilde{\sigma}_i, \tilde{q}_i)$ for $i = 1, \ldots, \lambda$
20:   Select, to compose the new population with individuals $(\mathbf{x}_k, \sigma_k, q_k)$, the μ individuals with best fitness from the population composed of the offspring $(\tilde{\mathbf{x}}_i, \tilde{\sigma}_i, \tilde{q}_i)$ for $i = 1, \ldots, \lambda$
21: **end while**

---

7 Restart q-Gaussian evolution strategy

In Section 5, EP algorithms with q-Gaussian mutation were presented. However, it is important to observe that the proposed self-adapted q-Gaussian mutation can be used in other EAs. In this section, the q-Gaussian mutation is proposed in a (μ, λ)-ES with recombination. The proposed algorithm, called *RqGES*, is presented in Algorithm 2.

Besides mutation, like in the EP algorihtms presented in Section 5, the proposed RqGES employs an intermediate recombination with $\rho = 2$, i.e., two parents are randomly chosen from the parent population and are mixed to generate an offspring. The intermediate recombination is applied to the variables $(\mathbf{x}_i)$ and parameters (σ_i and q_i). After recombination, the q-Gaussian mutation generated from isotropic distribution is used. In multimodal problems, larger population sizes can be useful to help the population escaping from local optima. In this way, following the scheme proposed in [1], the population is restarted with a larger number of individuals if a convergence criterion is met. Here, like in [1], the population is increased by a factor of 2 until a maximum allowed size λ_{max}.

Table 1 Test functions, where the vector $\mathbf{x}^*$ is the global optimum and the range is for each element of $\mathbf{x}$.

Prob.	Name	$f(\mathbf{x}^*)$	Range	Properties[1]
f_1	Shifted Sphere Function	-450	[-100,100]	U, S
f_2	Shifted Schwefel's Prob. 1.2	-450	[-100,100]	U, NS
f_3	Shifted Rotated High Conditioned Elliptic Function	-450	[-100,100]	U, NS, R
f_4	Shifted Schwefel's Prob. 1.2 with Noise	-450	[-100,100]	U, NS, N
f_5	Schwefel's Prob. 2.6 with Global Optimum on Bounds	-310	[-100,100]	U, NS, B
f_6	Shifted Rosenbrock's Function	390	[-100,100]	M, NS
f_7	Shifted Rotated Griewank's Function without Bounds	-180	$[-\infty, \infty]$	M, NS, R
f_8	Shifted Rotated Ackley's Function with Global Optimum on Bounds	-140	[-32,32]	M, NS, R
f_9	Shifted Rastrigin's Function	-330	[-5,5]	M, S
f_{10}	Shifted Rotated Rastrigin's Function	-330	[-5,5]	M, NS, R
f_{11}	Shifted Rotated Weierstrass' Function	90	[-0.5,0.5]	M, NS, R
f_{12}	Schwefel's Prob. 2.13	-460	$[-\pi, \pi]$	M, NS
f_{13}	Shifted Expanded Griewank's plus Rosenbrock's Function	-130	[-3,1]	M, NS
f_{14}	Shifted Rotated Expanded Schaffer's F6 Function	-300	[-100,100]	M, NS
f_{15}	Hybrid Composition Function	120	[-5,5]	M, SO
f_{16}	Rotated Version of Hybrid Composition Function f_{15}	120	[-5,5]	M, NS, R
f_{17}	Function f_{16} with Noise	120	[-5,5]	M, NS, R, N
f_{18}	Rotated Hybrid Composition Function 1	10	[-5,5]	M, NS, R
f_{19}	Rotated Hybrid Composition Function 1 with Narrow Basin Global Optimum	10	[-5,5]	M, NS, NB
f_{20}	Rotated Hybrid Composition Function 1 with Global Optimum on Bounds	10	[-5,5]	M, NS, B
f_{21}	Rotated Hybrid Composition Function 2	360	[-5,5]	M, NS, R
f_{22}	Rotated Hybrid Composition Function 2 with High Condition Number Matrix	360	[-5,5]	M, NS, B
f_{23}	Non-Continuous Rotated Hybrid Composition Function 2	360	[-5,5]	M, NS, B
f_{24}	Rotated Hybrid Composition Function 3	260	[-5,5]	M, NS, R
f_{25}	Rotated Hybrid Composition Function 3 without Bounds	260	$[-\infty, \infty]$	M, NS, R

[1] U: unimodal, M: multimodal, S: separable, NS: non-separable, SO: separable near $\mathbf{x}^*$, R: rotated, N: with noise in fitness, B: with $\mathbf{x}^*$ on bounds, NB: narrow basin for $\mathbf{x}^*$

8 Experimental study

In order to investigate the performance of the proposed algorithms, twenty five benchmark functions as described in [35] are selected as the test suite in our experiments. The test functions, which should be minimized, are presented in Table 1 and are used with the same parameters as described in [35].

In the test suite, functions f_1 to f_5 are unimodal and the remaining functions are multimodal. Functions f_{15} to f_{25} are hybrid composition functions [35]. Such functions are composed of basic functions, which results in hybrid composite functions with different basic function properties, and are given by

$$f(\mathbf{x}) = \sum_{i=1}^{10} \left(w_i \Big(f_i'\big(\mathbf{M}_i(\mathbf{x}-\mathbf{o}_i)/\lambda_i\big) + bias_i \Big) \right) + f_{bias}, \quad (34)$$

where $f_i'(.)$ is the normalized i-th basic function, $\mathbf{M}_i$, λ_i, and w_i are the linear transformation matrix, the compress rate, and the weight value for each function $f_i(.)$ respectively, $\mathbf{o}_i$ defines the position of the local and global optima, $bias_i$ defines which optimum is the global optimum, and f_{bias} is the bias in the function value. For example, function f_{15} is composed of five basic functions: Rastrigin, Weierstrass, Griewank, Ackley, and Sphere. See [35] for details and parameters.

The functions presented in Table 1 allow comparing the three types of mutation operators described in this paper, i.e., Gaussian, Cauchy, and q-Gaussian mutations, on problems with different properties. For example, while some functions in Table 1 are separable, others are non-separable. Some important properties of the functions are presented in the last column of Table

1. The comparison of Gaussian, Cauchy, and q-Gaussian mutations on EP is performed in Section 8.2.

The functions presented in Table 1 also allow comparing the results of the algorithms with other EAs found in the literature. The comparison of RqGES, which presented better performance than qGEP and IqGEP, and other EAs for continuous optimization is performed in Section 8.3. In the following section, the design of all experiments presented in this paper is described.

8.1 Experimental design

In order to compare the EP algorithms with different types of mutation, each one was executed 25 times for each test function presented in Table 1 with $m = 10$ and $m = 30$ [35]. The same was done for RqES, but also with $m = 50$. For each run of an algorithm, the individuals of the initial population were randomly chosen with uniform distribution in the range of each function (see Table 1), except for the experiments with functions f_7 and f_{25}, where the populations were respectively initialized in the ranges $[0, 600]^m$ and $[2, 5]^m$. The number of fitness evaluations was set to $10000m$. In the EP algorithms, the population size was set to 100 individuals and the tournament size in the winning function [18] for selection was set to 10. In RqGES, the initial λ was set to 50, $\lambda_{max} = 200$, the initial μ was set to 15, and $g_s = 120$.

As suggested by the theoretical and empirical work [5], the parameters τ_b and τ_c are defined by $\tau_b = \frac{1}{\sqrt{2m}}$ and $\tau_c = \frac{1}{\sqrt{2\sqrt{m}}}$. Here, we propose to define the parameter τ_q as follows:

$$\tau_q = \frac{1}{\sqrt{2m}}. \tag{35}$$

The initial mutation strength parameter $\sigma_i(j)$ was set to $0.4|x_{max} - x_{min}|/\sqrt{m}$ for the algorithms with anisotropic mutation distributions, as employed in [22], and $0.4|x_{max} - x_{min}|$ for the algorithms with mutations generated from isotropic distributions, where x_{min} and x_{max} are respectively the minimum and maximum allowed values of each element of the solution in the initial population. The initial q-Gaussian parameter q in algorithms with q-Gaussian mutation was set to 1.0 (value where the Gaussian distribution is reproduced). In the algorithms with q-Gaussian mutation, $r_q = 0.8$ and the minimum and maximum values of the q-Gaussian parameter q were set to 0.8 and $0.8e$ respectively, i.e., values respectively smaller and higher than the values of q where the Gaussian and Cauchy mutations are reproduced. Experiments, not presented here, using the EP

algorithm with different parameter settings (e.g., with the minimum and maximum values of the q-Gaussian parameter q respectively set to 0.5 and $0.5e$) showed similar results to those presented in the following section.

8.2 Experimental results: evolutionary programming

8.2.1 Experimental results on qGEP

The experimental results of the fitness error $f(\mathbf{x^{best}}) - f(\mathbf{x}^*)$, where $f(\mathbf{x^{best}})$ is the best value of fitness found during the run, averaged over 25 runs in the experiments with anisotropic mutation distributions for $m = 10$ and $m = 30$ are presented in Tables 2 and 3, respectively. Figure 4 shows the convergence of the mean best-of-generation fitness error in the experiments with $m = 30$ for functions f_9 and f_{10}.

In Tables 4 and 5, the statistic comparison regarding the fitness error of the best individual found during each run is carried out by the Wilcoxon Signed Rank Test [12]. Tables 4 and 5 show the p-value of the Wilcoxon Signed Rank Test for each function, which indicates the significance for testing the null hypothesis that the difference between the matched samples of the results regarding Alg. A and Alg. B comes from a distribution with median equal to zero. For each problem f_i, the result regarding the comparison Alg. A - Alg. B is shown, in parentheses, as "=" when the values of the median of Alg. A and Alg. B are equal. When the values of the median are different but the p-value is higher than 0.05, i.e., the test indicates that the hypothesis that the median of the difference between the results are zero cannot be rejected at the 5% level, the result is respectively shown as "+" when the median of Alg. A is smaller than that of Alg. B and "−" when the median of Alg. A is higher than that of Alg. B. Otherwise, when the result is statistically significant, the result is respectively shown as "$s+$" or "$s−$" when the median of Alg. A is smaller or higher than the median of Alg. B.

In the last line of Tables 4 and 5, following the method proposed in [12], the comparison Alg. A - Alg. B regarding the results of all functions is presented, where the p-values of the Wilcoxon Signed Rank Test are obtained comparing the vectors with the mean results (shown in Tables 4 and 5) obtained for each function by Alg. A and Alg. B, respectively. In the second parentheses, the values in the last line also indicate the difference between the number of times (functions) that the median of Alg. A is smaller than the median of Alg. B (i.e., the number of "+" and "$s+$" in the respective column) and the number of times (functions) that the median of Alg. B is smaller than the median of Alg. A (i.e., the

Table 2 Experimental results of the error of the best fitness in each run obtained for algorithms GEP, CEP, EP ($q = 1.5$), and qGEP on the test functions with $m = 10$.

Prob		GEP	CEP	EP ($q = 1.5$)	qGEP
f_1	median	1.25E-10	4.83E-10	1.87E-10	**9.32E-11**
	mean	1.31E-10	4.71E-10	1.89E-10	9.06E-11
	std	3.88E-11	1.04E-10	4.83E-11	2.45E-11
f_2	median	8.07E-06	1.46E-04	4.54E-05	**4.12E-07**
	mean	2.44E-04	8.52E-02	5.06E-03	1.95E-02
	std	9.37E-04	4.18E-01	2.41E-02	9.36E-02
f_3	median	2.40E+05	5.14E+05	2.61E+05	**2.36E+05**
	mean	2.71E+05	5.92E+05	3.01E+05	2.79E+05
	std	1.93E+05	3.73E+05	1.92E+05	2.41E+05
f_4	median	1.02E+01	5.75E+00	3.30E+00	**3.70E-02**
	mean	4.04E+01	2.06E+01	4.23E+01	2.48E-01
	std	6.90E+01	3.32E+01	9.90E+01	4.17E-01
f_5	median	1.76E-01	**3.58E-04**	5.35E-04	4.11E-04
	mean	7.25E+01	8.78E-01	1.43E+01	4.90E+00
	std	1.88E+02	3.11E+00	5.12E+01	2.32E+01
f_6	median	5.70E+00	1.63E+01	**5.16E+00**	8.21E+00
	mean	5.08E+01	1.33E+02	2.23E+01	6.19E+01
	std	7.01E+01	2.81E+02	3.94E+01	1.55E+02
f_7	median	7.26E-01	4.73E-01	6.95E-01	**8.11E-02**
	mean	1.90E+00	9.36E-01	1.08E+00	1.09E-01
	std	4.56E+00	1.30E+00	1.20E+00	7.65E-02
f_8	median	2.04E+01	2.04E+01	**2.03E+01**	**2.03E+01**
	mean	2.03E+01	2.03E+01	2.03E+01	2.03E+01
	std	8.68E-02	7.99E-02	6.88E-02	7.95E-02
f_9	median	1.29E+01	**4.97E+00**	1.49E+01	8.95E+00
	mean	1.49E+01	6.77E+00	1.54E+01	8.24E+00
	std	7.96E+00	6.09E+00	5.79E+00	2.70E+00
f_{10}	median	2.09E+01	2.09E+01	2.19E+01	**1.09E+01**
	mean	2.36E+01	2.23E+01	2.12E+01	1.21E+01
	std	1.14E+01	1.07E+01	9.28E+00	5.05E+00
f_{11}	median	4.43E+00	4.24E+00	4.99E+00	**3.12E+00**
	mean	4.24E+00	4.08E+00	5.15E+00	3.07E+00
	std	1.15E+00	1.42E+00	1.12E+00	1.08E+00
f_{12}	median	2.58E+02	2.10E+02	5.95E+01	**3.59E+01**
	mean	1.17E+03	4.83E+02	5.82E+02	3.41E+02
	std	2.07E+03	6.58E+02	1.34E+03	7.39E+02
f_{13}	median	1.09E+00	**7.49E-01**	1.09E+00	9.11E-01
	mean	1.18E+00	6.88E-01	1.20E+00	9.76E-01
	std	4.84E-01	2.89E-01	4.54E-01	3.21E-01
f_{14}	median	3.46E+00	3.28E+00	3.29E+00	**2.68E+00**
	mean	3.27E+00	3.21E+00	3.27E+00	2.71E+00
	std	4.64E-01	4.36E-01	3.29E-01	4.64E-01
f_{15}	median	4.31E+02	4.30E+02	4.43E+02	**4.16E+02**
	mean	3.80E+02	3.57E+02	4.10E+02	4.18E+02
	std	1.11E+02	1.31E+02	1.00E+02	1.20E+01
f_{16}	median	1.44E+02	1.37E+02	1.40E+02	**1.18E+02**
	mean	1.45E+02	1.46E+02	1.45E+02	1.21E+02
	std	2.65E+01	2.92E+01	2.65E+01	1.63E+01
f_{17}	median	1.43E+02	1.51E+02	1.38E+02	**1.22E+02**
	mean	1.41E+02	1.49E+02	1.41E+02	1.23E+02
	std	2.08E+01	2.44E+01	2.69E+01	1.60E+01
f_{18}	median	7.90E+02	4.85E+02	8.00E+02	**3.55E+02**
	mean	6.81E+02	5.63E+02	6.97E+02	4.80E+02
	std	2.28E+02	2.06E+02	2.19E+02	2.26E+02
f_{19}	median	8.00E+02	5.04E+02	6.79E+02	**3.48E+02**
	mean	7.14E+02	5.70E+02	6.55E+02	4.39E+02
	std	2.37E+02	2.05E+02	2.08E+02	2.09E+02
f_{20}	median	8.00E+02	4.25E+02	6.37E+02	**4.07E+02**
	mean	7.15E+02	4.87E+02	6.35E+02	4.94E+02
	std	2.15E+02	1.84E+02	1.97E+02	2.23E+02
f_{21}	median	9.08E+02	**5.00E+02**	**5.00E+02**	**5.00E+02**
	mean	8.26E+02	6.39E+02	6.75E+02	6.27E+02
	std	3.16E+02	2.68E+02	2.42E+02	2.20E+02
f_{22}	median	7.68E+02	7.78E+02	7.67E+02	**7.42E+02**
	mean	7.65E+02	7.91E+02	7.58E+02	7.45E+02
	std	1.02E+02	3.93E+01	1.01E+02	2.04E+01
f_{23}	median	9.64E+02	**5.59E+02**	7.21E+02	**5.59E+02**
	mean	9.12E+02	7.38E+02	8.39E+02	7.08E+02
	std	2.75E+02	2.89E+02	2.84E+02	2.64E+02
f_{24}	median	**2.00E+02**	2.00E+02	2.00E+02	**2.00E+02**
	mean	2.00E+02	2.32E+02	2.00E+02	2.00E+02
	std	0.00E+00	1.59E+02	0.00E+00	0.00E+00
f_{25}	median	4.11E+02	4.10E+02	4.10E+02	**4.06E+02**
	mean	4.12E+02	4.11E+02	4.11E+02	3.98E+02
	std	1.01E+01	5.78E+00	9.57E+00	4.15E+01

Table 3 Experimental results of the error of the best fitness obtained for algorithms GEP, CEP, EP ($q = 1.5$), and qGEP on the test functions with $m = 30$.

Prob		GEP	CEP	EP ($q = 1.5$)	qGEP
f_1	median	**3.10E-09**	3.17E-08	5.54E-09	4.33E-09
	mean	3.16E-09	3.09E-08	5.75E-09	4.53E-07
	std	3.52E-010	3.71E-09	1.05E-09	1.82E-06
f_2	median	7.30E+02	2.77E+02	4.04E+02	**2.04E+02**
	mean	7.02E+02	4.94E+02	6.27E+02	3.96E+02
	std	5.19E+02	6.13E+02	5.59E+02	5.61E+02
f_3	median	5.62E+06	6.41E+06	4.47E+06	**4.17E+06**
	mean	5.86E+06	6.28E+06	5.36E+06	4.32E+06
	std	2.75E+06	2.53E+06	3.26E+06	2.34E+06
f_4	median	1.06E+04	1.14E+04	1.11E+04	**1.72E+03**
	mean	1.31E+04	1.31E+04	1.38E+04	2.37E+03
	std	8.21E+03	6.45E+03	7.95E+03	1.81E+03
f_5	median	7.53E+03	6.80E+03	7.42E+03	**3.56E+03**
	mean	7.28E+03	6.71E+03	7.52E+03	3.80E+03
	std	1.58E+03	1.34E+03	1.67E+03	1.39E+03
f_6	median	1.13E+02	**7.56E+01**	8.02E+01	8.38E+01
	mean	1.88E+02	2.16E+02	1.04E+02	2.33E+02
	std	2.81E+02	3.50E+02	1.02E+02	3.21E+02
f_7	median	2.08E+01	2.00E+00	1.79E+01	**1.13E+00**
	mean	2.89E+01	2.88E+00	2.45E+01	4.15E+00
	std	2.20E+01	1.76E+00	2.19E+01	8.35E+00
f_8	median	**2.09E+01**	2.10E+01	2.10E+01	2.10E+01
	mean	2.09E+01	2.09E+01	2.09E+01	2.10E+01
	std	5.19E-02	4.21E-02	5.69E-02	5.01E-02
f_9	median	9.45E+01	**1.10E+01**	8.76E+01	3.58E+01
	mean	9.34E+01	1.16E+01	8.56E+01	3.75E+01
	std	2.72E+01	3.48E+00	1.94E+01	1.19E+01
f_{10}	median	1.05E+02	9.35E+01	1.03E+02	**5.31E+01**
	mean	1.11E+02	9.72E+01	9.68E+01	5.40E+01
	std	2.86E+01	2.34E+01	2.24E+01	1.72E+01
f_{11}	median	2.29E+01	2.00E+01	2.07E+01	**1.32E+01**
	mean	2.26E+01	1.97E+01	2.14E+01	1.37E+01
	std	3.31E+00	2.62E+00	2.61E+00	1.98E+00
f_{12}	median	2.42E+04	1.98E+04	2.58E+04	**1.69E+04**
	mean	3.14E+04	2.54E+04	3.30E+04	2.10E+04
	std	2.36E+04	2.04E+04	2.75E+04	1.17E+04
f_{13}	median	7.86E+00	**2.48E+00**	8.55E+00	3.36E+00
	mean	8.25E+00	3.03E+00	1.03E+01	3.29E+00
	std	2.81E+00	1.13E+00	5.24E+00	8.48E-01
f_{14}	median	**1.19E+01**	1.20E+01	1.20E+01	1.22E+01
	mean	1.19E+01	1.19E+01	1.19E+01	1.22E+01
	std	6.33E-01	5.43E-01	5.63E-01	3.67E-01
f_{15}	median	4.42E+02	**3.02E+02**	4.08E+02	4.07E+02
	mean	4.70E+02	3.12E+02	4.24E+02	4.32E+02
	std	6.26E+01	1.05E+02	3.95E+01	6.27E+01
f_{16}	median	1.22E+02	1.07E+02	1.35E+02	**6.79E+01**
	mean	1.32E+02	1.18E+02	1.41E+02	7.89E+01
	std	6.34E+01	3.29E+01	6.57E+01	3.03E+01
f_{17}	median	1.62E+02	1.14E+02	1.50E+02	**7.91E+01**
	mean	1.86E+02	1.33E+02	1.55E+02	9.21E+01
	std	9.40E+01	7.53E+01	4.12E+01	3.47E+01
f_{18}	median	9.45E+02	9.16E+02	9.38E+02	**9.12E+02**
	mean	9.48E+02	9.19E+02	9.39E+02	9.14E+02
	std	2.17E+01	8.09E+00	1.43E+01	6.45E+00
f_{19}	median	9.41E+02	9.16E+02	9.33E+02	**9.13E+02**
	mean	9.44E+02	9.19E+02	9.36E+02	9.15E+02
	std	1.95E+01	9.32E+00	1.48E+01	7.53E+00
f_{20}	median	9.38E+02	9.13E+02	9.38E+02	**9.11E+02**
	mean	9.44E+02	9.16E+02	9.40E+02	9.14E+02
	std	1.84E+01	6.38E+00	1.61E+01	6.97E+00
f_{21}	median	**5.00E+02**	**5.00E+02**	**5.00E+02**	**5.00E+02**
	mean	6.57E+02	7.17E+02	6.65E+02	5.37E+02
	std	2.86E+02	2.95E+02	2.78E+02	1.02E+02
f_{22}	median	9.30E+02	**8.76E+02**	9.05E+02	8.84E+02
	mean	9.27E+02	8.81E+02	9.10E+02	8.85E+02
	std	3.47E+01	3.41E+01	3.00E+01	3.02E+01
f_{23}	median	8.97E+02	**5.34E+02**	7.95E+02	5.40E+02
	mean	9.12E+02	6.96E+02	8.26E+02	6.24E+02
	std	1.51E+02	2.64E+02	1.78E+02	1.74E+02
f_{24}	median	6.09E+02	9.74E+02	**2.00E+02**	**2.00E+02**
	mean	5.83E+02	7.91E+02	4.81E+02	2.43E+02
	std	3.91E+02	3.39E+02	3.66E+02	1.60E+02
f_{25}	median	2.53E+02	**2.20E+02**	2.46E+02	2.25E+02
	mean	4.19E+02	2.21E+02	3.05E+02	2.30E+02
	std	3.92E+02	4.15E+00	2.21E+02	2.06E+01
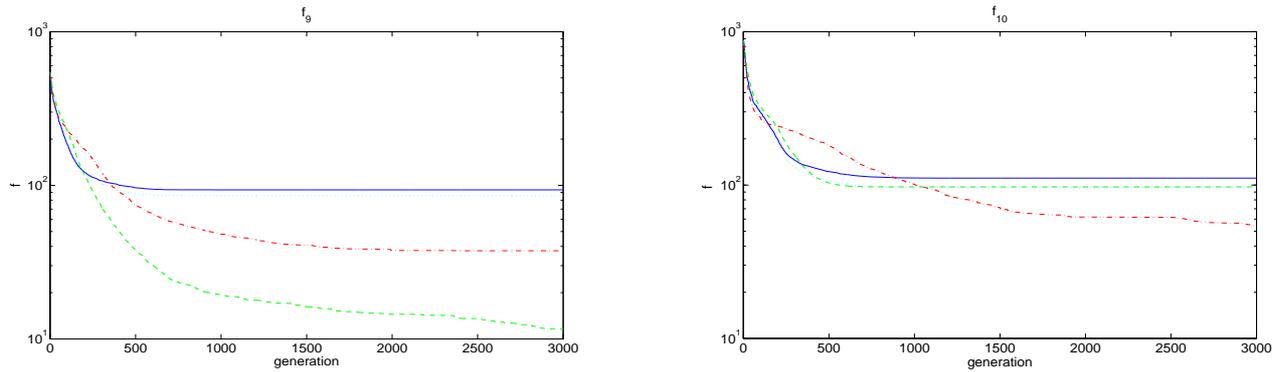
Fig. 4 Mean $f(\mathbf{x}_{\mathbf{g}}^{\mathbf{best}}) - f(\mathbf{x}^*)$, where $\mathbf{x}_{\mathbf{g}}^{\mathbf{best}}$ is the best-of-generation individual and $\mathbf{x}^*$ is the global optimum, in the experiments with functions f_9 and f_{10} with $m = 30$ for the algorithms: GEP (solid line), CEP (dashed line), EP ($q = 1.5$) (dotted line) and qGEP (dash-dotted line).

Table 4 Statistical comparison of algorithms GEP, CEP, EP ($q = 1.5$), and qGEP regarding the error of the best fitness on the test functions with $m = 10$. The p-value of the Wilcoxon Signed Rank Test regarding the comparison Alg. A - Alg. B is presented. When the median of Alg. A is smaller or higher than the median of Alg. B, the result is respectively shown in parentheses as "+" or "−". When the result is statistically significant, an "s" is added. In the last line, the difference between the number of "+" and "−" of each column is still shown in the second parentheses.

Prob	CEP - GEP	qGEP - GEP	qGEP - CEP	qGEP - EP ($q = 1.5$)
f_1	1.23E-05 (s-)	2.40E-04 (s+)	1.23E-05 (s+)	1.23E-05 (s+)
f_2	2.83E-02 (s-)	9.46E-01 (+)	3.67E-01 (+)	8.82E-01 (+)
f_3	2.47E-03 (s-)	7.78E-01 (+)	2.66E-04 (+)	3.67E-01 (+)
f_4	2.11E-01 (+)	5.76E-05 (s+)	1.13E-04 (s+)	4.03E-04 (s+)
f_5	3.22E-03 (s+)	2.30E-02 (s+)	3.67E-01 (−)	4.12E-01 (+)
f_6	7.98E-01 (−)	4.59E-01 (−)	3.97E-01 (+)	5.81E-01 (−)
f_7	4.43E-01 (+)	1.23E-05 (s+)	2.26E-05 (s+)	1.77E-05 (s+)
f_8	8.19E-01 (+)	8.40E-01 (+)	6.38E-01 (+)	9.25E-01 (+)
f_9	1.40E-04 (s+)	2.35E-03 (s+)	7.14E-02 (−)	3.03E-05 (s+)
f_{10}	6.96E-01 (=)	1.62E-04 (s+)	3.11E-04 (s+)	1.30E-03 (s+)
f_{11}	9.25E-01 (+)	7.33E-04 (s+)	4.53E-03 (s+)	5.76E-05 (s+)
f_{12}	3.00E-01 (+)	4.80E-02 (s+)	4.50E-02 (s+)	6.77E-01 (+)
f_{13}	1.74E-04 (s+)	5.11E-02 (+)	1.72E-03 (s-)	2.47E-02 (s+)
f_{14}	3.00E-01 (+)	8.91E-04 (s+)	1.19E-03 (s+)	2.96E-04 (s+)
f_{15}	4.59E-01 (+)	4.76E-01 (+)	9.04E-01 (+)	6.53E-02 (+)
f_{16}	7.78E-01 (+)	6.02E-04 (s+)	2.70E-03 (s+)	2.70E-03 (s+)
f_{17}	3.53E-01 (−)	1.19E-02 (s+)	5.45E-04 (s+)	1.38E-02 (s+)
f_{18}	6.53E-02 (+)	2.70E-03 (s+)	1.71E-01 (+)	1.01E-04 (s+)
f_{19}	2.64E-02 (s+)	3.18E-04 (s+)	5.11E-02 (+)	1.43E-03 (s+)
f_{20}	6.02E-04 (s+)	1.89E-03 (s+)	7.53E-01 (+)	8.14E-03 (s+)
f_{21}	5.09E-02 (+)	1.66E-02 (s+)	6.69E-01 (=)	4.66E-01 (=)
f_{22}	5.10E-01 (−)	2.16E-04 (s+)	4.03E-04 (s+)	1.57E-03 (s+)
f_{23}	5.34E-02 (+)	8.22E-03 (s+)	5.01E-01 (=)	1.26E-01 (+)
f_{24}	1.00E+00 (=)	1.00E+00 (=)	1.00E+00 (=)	1.00E+00 (=)
f_{25}	8.19E-01 (+)	6.65E-04 (s+)	1.43E-03 (s+)	2.96E-04 (s+)
$f_1 - f_{25}$	9.46E-02 (+),(+11)	5.60E-03 (s+),(+22)	3.00E-03 (s+),(+16)	5.88E-04 (s+),(+21)

number of "−" and "$s-$" in the respective column). For example, the result 5.60E-03 (+s,+22) in the last line, third column of Table 4 indicates that the p-value of the Wilcoxon Signed Rank Test regarding the comparison between qGEP and GEP is 5.60E-03, which means that the result is statistically significant ("$s+$"). The result also indicates that the difference between the number of times that the median of qGEP is smaller than the median of GEP and the number of times that the median of qGEP is higher than the median of GEP is 22. The analysis of the experimental results shown here is presented in Section 8.2.3.

8.2.2 Experimental results on IqGEP

The experimental results of the fitness error averaged over 25 runs in the experiments with mutation generated from isotropic distributions for $m = 10$ and $m = 30$ are respectively presented in Tables 6 and 7. Figure 5 shows the convergence of the mean best-of-generation fitness error in the experiments with $m = 30$ for functions f_9 and f_{10}. In Tables 8 and 9, the statistic comparison regarding the fitness error is carried out by the Wilcoxon Signed Rank Test. Tables 10 and 11 show the p-value of the Wilcoxon Signed Rank Test regarding the comparison between the isotropic and anisotropic

Table 5 Statistical comparison of algorithms GEP, CEP, EP ($q = 1.5$), and qGEP regarding the error of the best fitness on the test functions with $m = 30$.

Prob	CEP - GEP	qGEP - GEP	qGEP - CEP	qGEP - EP ($q = 1.5$)
f_1	1.23E-05 (s-)	1.73E-02 (s-)	6.53E-02 (+)	5.81E-01 (+)
f_2	3.96E-02 (s+)	3.96E-02 (s+)	1.92E-01 (+)	2.64E-02 (s+)
f_3	7.78E-01 ($-$)	2.30E-02 (s+)	3.70E-02 (s+)	1.04E-01 (+)
f_4	8.19E-01 ($-$)	1.39E-05 (s+)	1.77E-05 (s+)	1.23E-05 (s+)
f_5	1.74E-01 (+)	2.00E-05 (s+)	9.04E-05 (s+)	2.54E-05 (s+)
f_6	5.10E-01 (+)	4.27E-01 (+)	4.43E-01 ($-$)	2.11E-01 ($-$)
f_7	2.54E-05 (s+)	2.66E-04 (s+)	5.44E-02 (+)	3.64E-04 (s+)
f_8	3.67E-01 ($-$)	6.77E-01 ($-$)	5.27E-01 ($-$)	6.57E-01 (+)
f_9	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s-)	1.39E-05 (s+)
f_{10}	4.50E-02 (s+)	1.23E-05 (s+)	2.86E-05 (s+)	1.39E-05 (s+)
f_{11}	8.08E-04 (s+)	1.23E-05 (s+)	2.00E-05 (s+)	1.23E-05 (s+)
f_{12}	1.35E-01 (+)	1.66E-01 (+)	9.89E-01 (+)	6.15E-02 (+)
f_{13}	1.23E-05 (s+)	1.39E-05 (s+)	1.92E-01 ($-$)	1.23E-05 (s+)
f_{14}	9.68E-01 ($-$)	2.83E-02 (s-)	2.30E-02 (s+)	8.75E-02 ($-$)
f_{15}	3.62E-05 (s+)	4.53E-03 (s+)	1.43E-03 (s-)	9.54E-01 (+)
f_{16}	4.76E-01 (+)	2.66E-04 (s+)	4.93E-04 (s+)	1.40E-04 (s+)
f_{17}	1.19E-03 (s+)	3.62E-05 (s+)	3.82E-03 (s+)	6.45E-05 (s+)
f_{18}	1.23E-05 (s+)	2.00E-05 (s+)	6.93E-02 (+)	2.86E-05 (s+)
f_{19}	1.40E-04 (s+)	1.39E-05 (s+)	1.50E-01 (+)	6.45E-05 (s+)
f_{20}	1.39E-05 (s+)	4.07E-05 (s+)	1.83E-01 (+)	3.22E-05 (s+)
f_{21}	1.00E+00 (=)	4.69E-02 (=)	6.84E-03 (=)	2.34E-02 (=)
f_{22}	7.33E-04 (s+)	6.65E-04 (s+)	8.82E-01 ($-$)	3.82E-03 (s+)
f_{23}	1.19E-03 (s+)	9.04E-05 (s+)	7.37E-01 ($-$)	1.08E-03 (s+)
f_{24}	1.01E-01 ($-$)	1.12E-03 (s+)	1.20E-04 (s+)	1.25E-02 (=)
f_{25}	1.39E-05 (s+)	3.64E-04 (s+)	1.10E-02 (s-)	6.65E-04 (s+)
$f_1 - f_{25}$	3.62E-02 (s+),(+12)	1.02E-04 (s+),(+18)	1.84E-02 (s+),(+6)	3.55E-04 (s+),(+19)

algorithms. Table 12 summarizes the results regarding the comparison among qGEP, IqGEP, and other investigated algorithms for different groups of functions. In Table 12, the results regarding the comparison Alg. A - Alg. B indicate the difference between the number of times (functions) that the median of Alg. A is smaller than the median of Alg. B (i.e., the number of "+" and "$s+$" for the group of functions in the respective tables) and the number of times (functions) that the median of Alg. B is smaller than the median of Alg. A (i.e., the number of "$-$" and "$s-$" for the group of functions in the respective tables). The analysis of the experimental results shown here is presented in Secion 8.2.3.

8.2.3 Analysis

Some observations can be made from the experimental results presented in the tables and figures shown in last sections. They are described and analyzed below.

First, it is observable that the algorithms with the Gaussian mutation, i.e., GEP and IGEP, generally have a better performance than those with the Cauchy mutation, i.e., CEP and ICEP, on unimodal functions (lines for functions $f_1 - f_5$ in column 2 of Tables 4, 5, 8, and 9). Similar results of comparing long tail and Gaussian distributions in the minimization of unimodal functions were reported in the literature [18, 45]. In an unimodal function, long jumps, which often occur when the Cauchy mutation is employed, generally cause a degradation in the performance of the algorithm at the later stage of the evolution because less offspring are gen-

erated to explore the local neighborhood. For all functions and algorithms, the norm of the mutation strength parameter vector generally remains large in the initial stage of the evolution, allowing a faster convergence in the initial steps, and decays to small values in later generations.

It can also be observed that the performance of the proposed algorithms (i.e., qGEP and IqGEP) is generally better than that of the algorithms with Cauchy mutation (i.e., CEP and ICEP), and is, in general, equal to or better than that of the algorithms with Gaussian mutation (i.e., GEP and IGEP) on the unimodal functions (lines for functions $f_1 - f_5$ in Tables 4, 5, 8, and 9). These results can be explained because, in general, the q-Gaussian mutation generates less long jumps than the Cauchy mutation and more long jumps than the Gaussian mutation for $q \geq 1$. However, when $q < 1$, the q-Gaussian distribution is more compact than the Gaussian distribution, which implies shorter jumps.

Second, while the algorithms with Gaussian mutation (i.e., GEP and IGEP) generally outperform the algorithms with Cauchy mutation (i.e., CEP and ICEP) on the unimodal functions, they are generally outperformed on the multimodal functions, especially on the highly multimodal functions, e.g., functions f_9 and f_{10}. These results indicate that the long jumps generated by the Cauchy mutation are advantageous for the individuals to escape from local optima, especially in the later stage of the evolution when the mutation strength parameters have converged to small values.

Table 6 Experimental results of the error of the best fitness obtained for algorithms IGEP, ICEP, IEP ($q = 1.5$) and IqGEP on the test functions with $m = 10$

Prob		IGEP	ICEP	IEP ($q = 1.5$)	IqGEP
f_1	median	**4.45E-14**	9.74E-14	6.47E-14	2.69E-12
	mean	4.60E-14	9.61E-14	6.51E-14	3.03E-07
	std	1.57E-14	3.45E-14	2.10E-14	1.31E-06
f_2	median	**6.82E-13**	1.69E-12	7.13E-13	1.92E-09
	mean	6.90E-13	1.96E-12	9.02E-13	1.40E-05
	std	4.06E-13	1.07E-12	5.61E-13	6.01E-05
f_3	median	2.16E+05	2.39E+05	**2.04E+05**	2.14E+05
	mean	3.00E+05	2.82E+05	2.28E+05	2.51E+05
	std	2.59E+05	1.74E+05	1.96E+05	1.70E+05
f_4	median	**1.50E-12**	3.75E-12	2.90E-12	1.53E-08
	mean	2.26E-12	8.21E-11	5.85E-12	3.71E-03
	std	2.48E-12	3.49E-10	8.91E-12	9.01E-03
f_5	median	**8.30E-06**	1.29E-05	9.85E-06	1.54E-01
	mean	8.61E-06	1.28E-05	1.06E-05	1.30E+00
	std	2.88E-06	4.71E-06	3.16E-06	2.55E+00
f_6	median	1.44E+02	3.02E+01	9.69E+01	**8.18E+00**
	mean	9.77E+02	1.55E+03	4.22E+02	1.90E+02
	std	2.05E+03	3.00E+03	8.06E+02	3.95E+02
f_7	median	1.38E-01	1.82E-01	1.33E-01	**7.39E-02**
	mean	1.45E-01	2.52E-01	1.92E-01	7.80E-02
	std	8.19E-02	1.87E-01	1.29E-01	2.88E-02
f_8	median	**2.03E+01**	2.04E+01	2.04E+01	**2.03E+01**
	mean	2.03E+01	2.04E+01	2.03E+01	2.03E+01
	std	6.37E-02	6.73E-02	8.27E-02	6.68E-02
f_9	median	1.89E+01	1.89E+01	2.19E+01	**1.29E+01**
	mean	1.81E+01	1.91E+01	2.07E+01	1.41E+01
	std	6.40E+00	6.76E+00	7.49E+00	4.98E+00
f_{10}	median	1.79E+01	1.89E+01	2.29E+01	**1.29E+01**
	mean	1.88E+01	2.10E+01	2.31E+01	1.25E+01
	std	7.28E+00	8.96E+00	9.66E+00	4.48E+00
f_{11}	median	3.60E+00	3.45E+00	2.84E+00	**1.77E+00**
	mean	3.36E+00	3.75E+00	3.07E+00	1.58E+00
	std	1.39E+00	1.45E+00	1.28E+00	1.08E+00
f_{12}	median	3.50E+01	2.08E+02	**2.54E+01**	3.06E+01
	mean	1.71E+03	1.00E+03	1.23E+03	8.05E+02
	std	2.86E+03	1.76E+03	2.80E+03	1.43E+03
f_{13}	median	9.26E-01	1.01E+00	1.16E+00	**8.13E-01**
	mean	1.05E+00	1.04E+00	1.19E+00	8.89E-01
	std	3.86E-01	4.32E-01	5.10E-01	3.12E-01
f_{14}	median	3.11E+00	3.12E+00	3.06E+00	**2.66E+00**
	mean	3.05E+00	3.13E+00	3.06E+00	2.66E+00
	std	4.86E-01	3.05E-01	3.58E-01	4.47E-01
f_{15}	median	4.11E+02	4.29E+02	4.05E+02	**2.50E+02**
	mean	3.35E+02	3.48E+02	3.50E+02	3.10E+02
	std	1.14E+02	1.22E+02	1.02E+02	1.11E+02
f_{16}	median	1.40E+02	1.36E+02	1.33E+02	**1.19E+02**
	mean	1.38E+02	1.42E+02	1.33E+02	1.19E+02
	std	2.00E+01	2.03E+01	1.32E+01	1.24E+01
f_{17}	median	1.36E+02	1.37E+02	1.37E+02	**1.18E+02**
	mean	1.39E+02	1.44E+02	1.34E+02	1.17E+02
	std	1.63E+01	2.39E+01	1.68E+01	1.07E+01
f_{18}	median	8.00E+02	3.56E+02	4.72E+02	**3.00E+02**
	mean	6.19E+02	5.26E+02	5.77E+02	4.25E+02
	std	3.00E+02	2.73E+02	2.74E+02	2.28E+02
f_{19}	median	8.00E+02	3.72E+02	8.00E+02	**3.00E+02**
	mean	6.02E+02	5.05E+02	6.11E+02	4.95E+02
	std	2.80E+02	2.58E+02	2.75E+02	2.68E+02
f_{20}	median	4.25E+02	8.00E+02	8.00E+02	**3.00E+02**
	mean	6.05E+02	6.22E+02	6.26E+02	5.58E+02
	std	2.79E+02	2.86E+02	2.67E+02	2.76E+02
f_{21}	median	**5.00E+02**	**5.00E+02**	5.00E+02	5.00E+02
	mean	7.14E+02	6.25E+02	5.54E+02	5.74E+02
	std	2.55E+02	2.08E+02	2.32E+02	2.05E+02
f_{22}	median	7.49E+02	7.54E+02	7.50E+02	**7.47E+02**
	mean	7.43E+02	7.67E+02	7.40E+02	7.38E+02
	std	9.57E+01	2.85E+01	9.38E+01	9.44E+01
f_{23}	median	**5.59E+02**	**5.59E+02**	5.59E+02	5.59E+02
	mean	7.80E+02	7.70E+02	7.37E+02	6.69E+02
	std	2.70E+02	2.57E+02	2.55E+02	1.87E+02
f_{24}	median	**2.00E+02**	2.00E+02	2.00E+02	2.00E+02
	mean	2.12E+02	2.00E+02	2.00E+02	2.00E+02
	std	6.00E+01	0.00E+00	0.00E+00	9.88E-05
f_{25}	median	4.06E+02	4.06E+02	4.06E+02	**4.05E+02**
	mean	4.05E+02	4.05E+02	4.05E+02	4.25E+02
	std	4.02E+00	4.52E+00	3.98E+00	1.56E+02

Table 7 Experimental results of the error of the best fitness obtained for algorithms IGEP, ICEP, IEP ($q = 1.5$) and IqGEP on the test functions with $m = 30$

Prob		IGEP	ICEP	IEP ($q = 1.5$)	IqGEP
f_1	median	**1.87E-13**	3.75E-13	2.51E-13	1.97E-07
	mean	1.92E-13	3.74E-13	2.54E-13	3.02E-03
	std	3.75E-14	8.30E-14	5.58E-14	1.31E-02
f_2	median	1.95E-03	1.12E-02	3.28E-03	**3.17E-04**
	mean	3.71E-03	1.57E-02	6.34E-03	1.09E-02
	std	4.24E-03	1.47E-02	8.76E-03	5.09E-02
f_3	median	2.65E+06	**2.09E+06**	2.45E+06	2.69E+06
	mean	2.59E+06	2.72E+06	2.76E+06	2.55E+06
	std	1.42E+06	1.30E+06	1.61E+06	1.01E+06
f_4	median	5.47E+02	3.96E+02	5.14E+02	**4.90E+00**
	mean	6.76E+02	6.76E+02	8.75E+02	1.28E+01
	std	5.99E+02	6.71E+02	8.95E+02	2.00E+01
f_5	median	5.08E+03	5.39E+03	5.75E+03	**3.43E+03**
	mean	5.31E+03	5.29E+03	5.61E+03	3.49E+03
	std	1.44E+03	1.49E+03	1.27E+03	1.08E+03
f_6	median	**2.70E+02**	5.81E+02	3.11E+02	1.15E+03
	mean	9.53E+02	3.05E+03	3.81E+03	3.49E+03
	std	2.03E+03	4.65E+03	5.07E+03	4.90E+03
f_7	median	3.69E-02	3.44E-02	2.95E-02	**2.28E-02**
	mean	3.64E-02	3.78E-02	3.06E-02	3.33E-02
	std	1.36E-02	3.79E-02	1.68E-02	4.37E-02
f_8	median	**2.09E+01**	2.09E+01	2.09E+01	2.09E+01
	mean	2.09E+01	2.09E+01	2.09E+01	2.09E+01
	std	4.42E-02	6.94E-02	7.84E-02	5.73E-02
f_9	median	1.17E+02	1.18E+02	1.27E+02	**9.47E+01**
	mean	1.27E+02	1.26E+02	1.28E+02	1.05E+02
	std	3.80E+01	2.71E+01	3.70E+01	3.11E+01
f_{10}	median	1.25E+02	1.16E+02	1.23E+02	**8.98E+01**
	mean	1.23E+02	1.14E+02	1.26E+02	9.58E+01
	std	2.85E+01	2.73E+01	3.88E+01	2.44E+01
f_{11}	median	1.81E+01	2.06E+01	1.73E+01	**1.64E+01**
	mean	1.85E+01	2.04E+01	1.81E+01	1.65E+01
	std	2.75E+00	3.00E+00	2.79E+00	2.82E+00
f_{12}	median	3.95E+04	**2.30E+04**	3.32E+04	2.94E+04
	mean	3.60E+04	3.28E+04	4.55E+04	3.11E+04
	std	2.45E+04	3.83E+04	4.33E+04	2.21E+04
f_{13}	median	5.97E+00	**5.43E+00**	5.73E+00	6.63E+00
	mean	6.54E+00	5.96E+00	6.05E+00	6.43E+00
	std	2.20E+00	2.06E+00	1.72E+00	1.85E+00
f_{14}	median	1.25E+01	1.26E+01	1.25E+01	**1.20E+01**
	mean	1.25E+01	1.26E+01	1.24E+01	1.21E+01
	std	4.25E-01	3.90E-01	4.13E-01	4.34E-01
f_{15}	median	5.02E+02	4.55E+02	5.00E+02	**4.03E+02**
	mean	4.81E+02	4.63E+02	4.79E+02	4.41E+02
	std	8.94E+01	6.38E+01	6.95E+01	5.60E+01
f_{16}	median	1.56E+02	1.49E+02	1.59E+02	**1.30E+02**
	mean	1.71E+02	1.66E+02	1.62E+02	1.39E+02
	std	6.87E+01	6.15E+01	4.50E+01	6.51E+01
f_{17}	median	1.55E+02	1.55E+02	1.70E+02	**1.20E+02**
	mean	1.65E+02	1.63E+02	1.66E+02	1.31E+02
	std	5.16E+01	6.98E+01	4.32E+01	3.28E+01
f_{18}	median	9.16E+02	9.19E+02	9.17E+02	**9.12E+02**
	mean	9.18E+02	9.19E+02	9.23E+02	9.12E+02
	std	7.11E+00	7.93E+00	1.41E+01	3.75E+00
f_{19}	median	9.20E+02	9.18E+02	9.17E+02	**9.10E+02**
	mean	9.19E+02	9.20E+02	9.17E+02	9.11E+02
	std	8.50E+00	7.93E+00	5.72E+00	2.99E+00
f_{20}	median	9.16E+02	9.18E+02	9.17E+02	**9.11E+02**
	mean	9.19E+02	9.21E+02	9.18E+02	9.12E+02
	std	7.51E+00	8.21E+00	6.68E+00	3.98E+00
f_{21}	median	**5.00E+02**	5.00E+02	5.00E+02	5.00E+02
	mean	5.66E+02	5.62E+02	6.03E+02	5.62E+02
	std	1.66E+02	1.80E+02	2.40E+02	1.26E+02
f_{22}	median	8.93E+02	9.06E+02	8.97E+02	**8.84E+02**
	mean	8.90E+02	9.06E+02	8.97E+02	8.90E+02
	std	2.66E+01	3.66E+01	2.37E+01	2.40E+01
f_{23}	median	5.53E+02	5.45E+02	5.44E+02	**5.34E+02**
	mean	6.17E+02	5.97E+02	5.79E+02	5.86E+02
	std	1.55E+02	1.37E+02	1.15E+02	1.38E+02
f_{24}	median	**2.00E+02**	2.00E+02	2.00E+02	2.00E+02
	mean	2.41E+02	2.31E+02	2.62E+02	2.00E+02
	std	2.07E+02	1.57E+02	2.15E+02	3.61E-03
f_{25}	median	2.12E+02	2.12E+02	2.12E+02	**2.11E+02**
	mean	2.12E+02	2.53E+02	2.12E+02	2.11E+02
	std	4.40E-01	2.04E+02	9.98E-01	2.34E+00
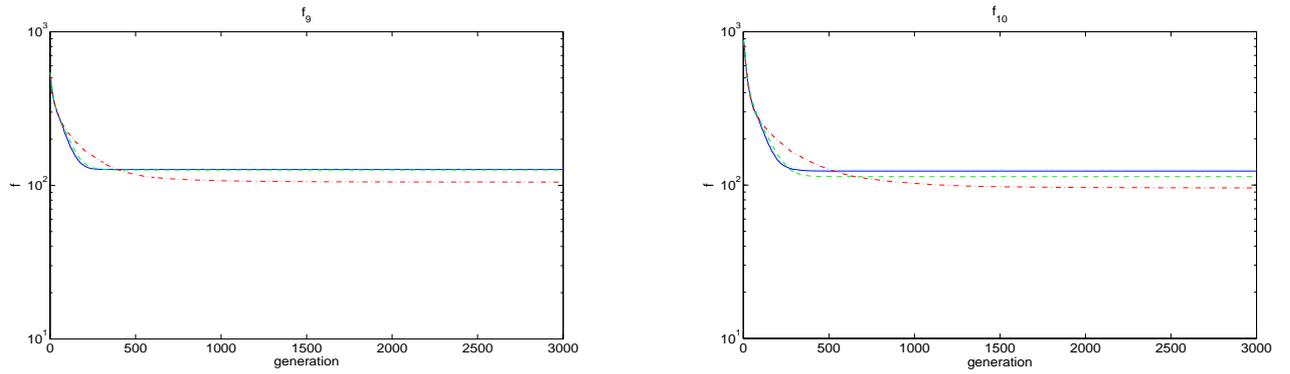
Fig. 5 Mean $f(\mathbf{x_g^{best}}) - f(\mathbf{x^*})$, in the experiments with functions f_9 and f_{10} with $m = 30$ for the algorithms: IGEP (solid line), ICEP (dashed line), IEP ($q = 1.5$) (dotted line), and IqGEP (dash-dotted line).

Table 8 Statistical comparison of algorithms IGEP, ICEP, IEP ($q = 1.5$) and IqGEP regarding the error of the best fitness on the test functions with $m = 10$

Prob	ICEP - IGEP	IqGEP - IGEP	IqGEP - ICEP	IqGEP - IEP ($q = 1.5$)
f_1	3.62E-05 (s-)	1.23E-05 (s-)	1.23E-05 (s-)	1.23E-05 (s-)
f_2	1.26E-04 (s-)	1.23E-05 (s-)	1.57E-05 (s-)	1.23E-05 (s-)
f_3	7.37E-01 ($-$)	5.10E-01 (+)	3.82E-01 (+)	3.67E-01 ($-$)
f_4	3.28E-04 (s-)	1.23E-05 (s-)	5.13E-05 (s-)	1.23E-05 (s-)
f_5	1.19E-03 (s-)	1.23E-05 (s-)	1.23E-05 (s-)	1.23E-05 (s-)
f_6	7.98E-01 (+)	5.82E-03 (s+)	2.64E-01 (+)	1.28E-01 (+)
f_7	1.73E-02 (s-)	4.03E-04 (s+)	2.86E-05 (s+)	6.45E-05 (s+)
f_8	5.10E-01 ($-$)	7.16E-01 (+)	4.76E-01 (+)	6.00E-01 (+)
f_9	6.00E-01 (=)	2.14E-02 (s+)	2.64E-02 (s+)	2.70E-03 (s+)
f_{10}	3.17E-01 ($-$)	1.08E-03 (s+)	4.93E-04 (s+)	1.33E-04 (s+)
f_{11}	6.38E-01 (+)	1.08E-03 (s+)	2.54E-05 (s+)	3.28E-04 (s+)
f_{12}	5.30E-01 ($-$)	2.11E-01 (+)	3.97E-01 (+)	9.89E-01 ($-$)
f_{13}	5.63E-01 ($-$)	1.04E-01 (+)	1.66E-01 (+)	6.31E-03 (s+)
f_{14}	6.38E-01 ($-$)	4.93E-03 (s+)	4.03E-04 (s+)	1.89E-03 (s+)
f_{15}	5.10E-01 ($-$)	1.58E-01 (+)	1.58E-01 (+)	8.27E-02 (+)
f_{16}	3.67E-01 (+)	1.89E-03 (s+)	8.09E-05 (s+)	1.08E-03 (s+)
f_{17}	4.93E-01 ($-$)	5.13E-05 (s+)	4.57E-05 (s+)	2.47E-03 (s+)
f_{18}	3.38E-01 (+)	1.71E-03 (s+)	1.31E-01 (+)	9.46E-03 (s+)
f_{19}	3.89E-01 (+)	1.17E-01 (+)	8.21E-01 (+)	2.61E-02 (s+)
f_{20}	9.10E-01 ($-$)	3.22E-01 (+)	3.48E-01 (+)	7.84E-03 (s+)
f_{21}	1.04E-01 (=)	2.66E-02 (=)	4.35E-01 (=)	7.74E-01 (=)
f_{22}	4.58E-01 ($-$)	2.53E-01 (+)	1.16E-01 (+)	3.17E-01 (+)
f_{23}	9.04E-01 (=)	2.18E-01 ($-$)	2.10E-01 ($-$)	4.32E-01 ($-$)
f_{24}	1.00E+00 (=)	8.75E-01 (=)	2.50E-01 (=)	2.50E-01 (=)
f_{25}	1.28E-01 (+)	5.11E-02 (+)	6.85E-03 (s+)	6.85E-03 (s+)
$f_1 - f_{25}$	7.97E-01($-$),(-9)	4.41-04 (s+),(+13)	4.41-04 (s+),(+13)	2.64E-02 (s+),(+9)

The proposed algorithms (qGEP and IqGEP) generally outperform the algorithms with Gaussian mutation on the multimodal functions, especially on the highly multimodal (e.g., functions f_9 and f_{10}) or hybrid functions (functions f_{15} - f_{20}).

The good performance of the proposed algorithms on multimodal functions can be explained because higher values of the parameter q were eventually employed to allow longer jumps (see Fig. 6, where the distribution parameter q of the current best individual on the first run of two experiments is plotted against the generation index). On the multimodal functions, after the initial stage, the evolution generally occurs in steps. It can be observed that such steps are generally coincident with larger changes in the value of q in Fig. 6, which indicates that jumps in individuals with higher values of q, or in individuals where the value of q has increased, were eventually beneficial. Similar results can be observed in the minimization of other multimodal functions.

These results indicate that an eventual increase in the value of q in the proposed algorithms can lead to longer jumps (when compared to those produced by Gaussian mutation), which can be advantageous to the evolution since they allow the population to escape from local optima, especially in later stages of the evolution when the mutation strength parameters are generally small. For the algorithms with mutation generated from isotropic distributions, the computed mean change in the best-of-generation fitness generated by advantageous mutations is generally higher for IqGEP on all functions, even when the final fitness is smaller for

Table 9 Statistical comparison of algorithms IGEP, ICEP, IEP ($q = 1.5$) and IqGEP regarding the error of the best fitness on the test functions with $m = 30$

Prob	ICEP - IGEP	IqGEP - IGEP	IqGEP - ICEP	IqGEP - IEP ($q = 1.5$)
f_1	1.39E-05 (s-)	1.23E-05 (s-)	1.23E-05 (s-)	1.23E-05 (s-)
f_2	8.08E-04 (s-)	3.82E-03 (s+)	2.16E-04 (s+)	1.57E-03 (s+)
f_3	5.27E-01 (+)	9.25E-01 (−)	7.16E-01 (−)	6.96E-01 (−)
f_4	7.37E-01 (+)	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)
f_5	6.96E-01 (−)	8.09E-05 (s+)	1.40E-04 (s+)	1.77E-05 (s+)
f_6	7.36E-02 (−)	4.50E-02 (s-)	3.82E-01 (−)	6.38E-01 (−)
f_7	7.57E-01 (+)	1.28E-01 (+)	3.82E-01 (+)	3.97E-01 (+)
f_8	7.37E-01 (+)	9.46E-01 (+)	9.25E-01 (−)	6.93E-02 (−)
f_9	9.89E-01 (−)	2.47E-02 (s+)	2.47E-02 (s+)	3.03E-02 (s+)
f_{10}	1.09E-01 (+)	1.08E-03 (s+)	3.03E-02 (s+)	8.04E-03 (s+)
f_{11}	6.15E-02 (−)	2.30E-02 (s+)	1.19E-03 (s+)	1.60E-02 (s+)
f_{12}	1.04E-01 (+)	4.76E-01 (+)	5.27E-01 (−)	2.01E-01 (+)
f_{13}	3.39E-01 (+)	9.04E-01 (−)	4.93E-01 (−)	4.93E-01 (−)
f_{14}	3.53E-01 (−)	7.42E-03 (s+)	1.43E-03 (s+)	2.06E-03 (s+)
f_{15}	5.45E-01 (+)	1.64E-02 (s+)	2.01E-01 (+)	3.91E-03 (s+)
f_{16}	7.98E-01 (+)	1.86E-02 (s+)	6.85E-03 (s+)	9.26E-02 (+)
f_{17}	8.19E-01 (−)	4.53E-03 (s+)	5.44E-02 (+)	9.80E-04 (s+)
f_{18}	8.40E-01 (−)	1.89E-03 (s+)	1.13E-04 (s+)	5.45E-04 (s+)
f_{19}	5.45E-01 (+)	1.74E-04 (s+)	1.40E-04 (s+)	6.45E-05 (s+)
f_{20}	5.27E-01 (−)	5.45E-04 (s+)	1.26E-04 (s+)	2.47E-03 (s+)
f_{21}	1.00E+00 (=)	8.17E-02 (−)	1.77E-01 (−)	3.95E-01 (−)
f_{22}	1.28E-01 (−)	9.04E-01 (+)	9.80E-02 (+)	2.88E-01 (+)
f_{23}	1.83E-01 (+)	1.72E-03 (s+)	1.38E-02 (s+)	5.44E-02 (+)
f_{24}	1.00E+00 (=)	2.81E-01 (=)	1.95E-01 (=)	5.78E-01 (=)
f_{25}	1.10E-02 (s-)	2.06E-03 (s+)	5.76E-05 (s+)	1.19E-03 (s+)
$f_1 - f_{25}$	6.59E-01 (−),(-1)	5.88E-04 (s+),(+14)	5.25-04 (s+),(+10)	2.41E-04 (s+),(+12)

Table 10 Statistical comparison of the EP algorithms with mutation generated from anisotropic or isotropic distributions regarding the best fitness on the test functions with $m = 10$

Prob	IGEP - GEP	ICEP - CEP	IEP ($q = 1.5$) - EP ($q = 1.5$)	IqGEP - qGEP
f_1	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)	7.78E-01 (+)
f_2	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)	8.04E-03 (s+)
f_3	8.40E-01 (+)	8.91E-04 (s+)	3.24E-02 (s+)	4.43E-01 (+)
f_4	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)	1.74E-04 (s+)
f_5	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)	2.06E-03 (s-)
f_6	4.16E-03 (s-)	1.74E-01 (−)	1.57E-03 (s-)	2.53E-01 (+)
f_7	2.54E-05 (s+)	4.46E-04 (s+)	1.26E-04 (s+)	1.74E-01 (+)
f_8	5.81E-01 (+)	6.77E-01 (−)	6.77E-01 (−)	7.37E-01 (−)
f_9	1.06E-01 (−)	6.45E-05 (s-)	2.22E-02 (s-)	1.56E-04 (s-)
f_{10}	1.26E-01 (+)	6.19E-01 (+)	9.57E-01 (−)	8.61E-01 (−)
f_{11}	3.03E-02 (s+)	6.38E-01 (+)	4.57E-05 (s+)	4.03E-04 (s+)
f_{12}	3.53E-01 (+)	5.63E-01 (+)	7.16E-01 (+)	5.27E-01 (+)
f_{13}	4.12E-01 (+)	3.82E-03 (s-)	8.19E-01 (−)	2.42E-01 (+)
f_{14}	8.75E-02 (+)	6.00E-01 (+)	2.83E-02 (s+)	7.78E-01 (+)
f_{15}	1.50E-01 (+)	9.89E-01 (+)	5.82E-03 (s+)	1.57E-03 (s+)
f_{16}	2.76E-01 (+)	8.82E-01 (+)	1.15E-01 (+)	6.77E-01 (−)
f_{17}	6.00E-01 (+)	5.27E-01 (+)	5.27E-01 (+)	2.01E-01 (+)
f_{18}	2.78E-01 (−)	5.92E-01 (+)	1.58E-01 (+)	1.77E-01 (+)
f_{19}	9.26E-02 (+)	3.00E-01 (+)	5.30E-01 (−)	7.94E-01 (+)
f_{20}	4.80E-02 (s+)	2.62E-02 (s-)	7.37E-01 (−)	6.61E-01 (+)
f_{21}	1.03E-01 (+)	5.76E-01 (+)	1.47E-01 (=)	4.96E-01 (=)
f_{22}	1.99E-02 (s+)	7.42E-03 (s+)	1.28E-01 (+)	2.31E-01 (−)
f_{23}	4.55E-02 (s+)	8.39E-01 (=)	2.27E-01 (+)	7.17E-01 (−)
f_{24}	1.00E+00 (=)	1.00E+00 (=)	1.00E+00 (=)	2.50E-01 (=)
f_{25}	1.39E-05 (s+)	1.13E-04 (s+)	3.62E-05 (s+)	8.71E-03 (s+)
$f_1 - f_{25}$	8.13E-02 (+),(+18)	1.70-01 (+),(+12)	1.38E-02 (s+),(+9)	4.47E-01 (+),(+9)

the other two algorithms. For IqGEP, when the mean changes in the fitness generated by advantageous mutations are analysed for different intervals of q after the middle of the evolutionary process (i.e., when the population has converged to local optima in most of the runs with multimodal functions), an interesting result can be observed. When these values are interpolated by a linear function, while the resulting line has a negative inclination for the unimodal functions (with exception of function f_3, where the inclination is close to zero), indicating that smaller values of q generally result in higher changes in the fitness function, the resulting line generally has a positive inclination for the multimodal functions, i.e., the selection of higher values of q resulted in higher changes in the fitness function.

When the performance of the algorithms with q-Gaussian mutation is statistically compared to that of the algorithms with Cauchy mutation on the multi-

Table 11 Statistical comparison of the EP algorithms with mutation generated from anisotropic or isotropic distributions regarding the best fitness on the test functions with $m = 30$

Prob	IGEP - GEP	ICEP - CEP	IEP ($q = 1.5$) - EP ($q = 1.5$)	IqGEP - qGEP
f_1	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)	1.30E-03 (s-)
f_2	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)
f_3	2.40E-04 (s+)	5.13E-05 (s+)	1.08E-02 (s+)	2.70E-03 (s+)
f_4	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)
f_5	6.45E-05 (s+)	2.06E-03 (s+)	1.89E-03 (s+)	6.19E-01 (+)
f_6	1.02E-02 (s-)	4.03E-04 (s-)	7.33E-04 (s-)	9.80E-04 (s-)
f_7	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)	1.23E-05 (s+)
f_8	1.83E-01 (+)	3.47E-02 (s+)	2.06E-03 (s+)	6.15E-02 (+)
f_9	2.26E-03 (s-)	1.23E-05 (s-)	6.45E-05 (s-)	1.23E-05 (s-)
f_{10}	7.80E-02 (−)	7.80E-02 (−)	5.35E-03 (s-)	2.26E-05 (s-)
f_{11}	5.45E-04 (s+)	3.82E-01 (−)	2.26E-03 (s+)	1.26E-04 (s-)
f_{12}	4.76E-01 (−)	7.98E-01 (−)	3.26E-01 (−)	2.42E-01 (−)
f_{13}	1.73E-02 (s+)	2.54E-05 (s-)	1.57E-04 (s+)	1.39E-05 (s-)
f_{14}	2.26E-03 (s-)	2.66E-04 (s-)	4.93E-03 (s-)	3.39E-01 (+)
f_{15}	9.54E-01 (−)	1.57E-04 (s-)	2.96E-03 (s-)	7.61E-01 (+)
f_{16}	2.47E-02 (s-)	2.70E-03 (s-)	7.36E-02 (−)	1.23E-05 (s-)
f_{17}	6.19E-01 (+)	8.04E-03 (s-)	2.11E-01 (−)	1.94E-04 (s-)
f_{18}	2.26E-05 (s+)	6.38E-01 (−)	5.45E-04 (s+)	6.93E-02 (+)
f_{19}	2.26E-05 (s+)	4.27E-01 (−)	1.77E-05 (s+)	3.47E-02 (s+)
f_{20}	7.22E-05 (s+)	1.49E-02 (s-)	1.23E-05 (s+)	3.39E-01 (+)
f_{21}	1.48E-01 (=)	7.71E-02 (=)	4.32E-01 (=)	3.43E-02 (s-)
f_{22}	1.57E-03 (s+)	8.71E-03 (s-)	1.83E-01 (+)	4.93E-01 (+)
f_{23}	2.86E-05 (s+)	7.78E-01 (−)	1.13E-04 (s+)	8.04E-03 (s+)
f_{24}	4.27E-04 (s+)	1.32E-04 (s+)	1.61E-02 (=)	9.84E-01 (=)
f_{25}	1.23E-05 (s+)	2.16E-04 (s+)	1.23E-05 (s+)	1.23E-05 (s+)
$f_1 - f_{25}$	4.87E-02 (s+),(+10)	9.03-01 (−),(-6)	1.37E-01 (+),(+7)	9.54E-01 (+),(+4)

Table 12 Comparison of the EP algorithms for different groups of functions. The difference between the number of "+" and "−" for the results presented in Tables 4-11 for each group of functions is shown.

		CEP - GEP	qGEP - GEP	qGEP - CEP	qGEP - EP ($q = 1.5$)
Unimodal	$m = 10$	-1	+5	+4	+5
($f_1 - f_5$)	$m = 30$	-1	+4	+5	+5
		ICEP - IGEP	IqGEP - IGEP	IqGEP - ICEP	IqGEP - IEP ($q = 1.5$)
	$m = 10$	-5	-4	-4	-5
	$m = 30$	-1	+1	+1	+1
Multimodal		CEP - GEP	qGEP - GEP	qGEP - CEP	qGEP - EP ($q = 1.5$)
	$m = 10$	+12	+17	+13	+16
($f_6 - f_{25}$)	$m = 30$	+13	+15	+1	+14
		ICEP - IGEP	IqGEP - IGEP	IqGEP - ICEP	IqGEP - IEP ($q = 1.5$)
	$m = 10$	-4	+16	+16	+14
	$m = 30$	0	+13	+9	+11
Composition Functions		CEP - GEP	qGEP - GEP	qGEP - CEP	qGEP - EP ($q = 1.5$)
	$m = 10$	+6	+10	+8	+9
($f_{15} - f_{25}$)	$m = 30$	+8	+10	+2	+9
		ICEP - IGEP	IqGEP - IGEP	IqGEP - ICEP	IqGEP - IEP ($q = 1.5$)
	$m = 10$	0	+7	+7	+7
	$m = 30$	-1	+8	+8	+8
Noise		CEP - GEP	qGEP - GEP	qGEP - CEP	qGEP - EP ($q = 1.5$)
	$m = 10$	0	+2	+2	+2
(f_4 and f_{17})	$m = 30$	0	+2	+2	+2
		ICEP - IGEP	IqGEP - IGEP	IqGEP - ICEP	IqGEP - IEP ($q = 1.5$)
	$m = 10$	-2	0	0	0
	$m = 30$	0	+2	+2	+2

modal functions, it can be observed that the proposed algorithms generally present a better or statistically similar performance (Tables 4, 5, 8, and 9). The better performance of the algorithm qGEP is the result of a better compromise between long and local jumps of the candidate solutions as the tail of the mutation distribution is self-adapted in qGEP. The use of q-Gaussian mutation is advantageous in noisy functions, too. It can be observed that the performance of the proposed algorithms is better, when compared to other algorithms (Tables 4, 5, and 9), with the exception in isotropic al-

gorithms on the function f_4 with $m = 10$, which is composed by adding noise to the unimodal function f_2.

Following the method proposed in [12], we can observe that the performance of the algorithms with q-Gaussian mutation is significantly better (with the exception for IqGEP when compared to IGEP and ICEP in the experiment with $m = 10$, where the performance of IqGEP is better, but not statistically significant) than the performance of all other algorithms when all functions are considered (last lines in Tables 4, 5, 8, and 9).
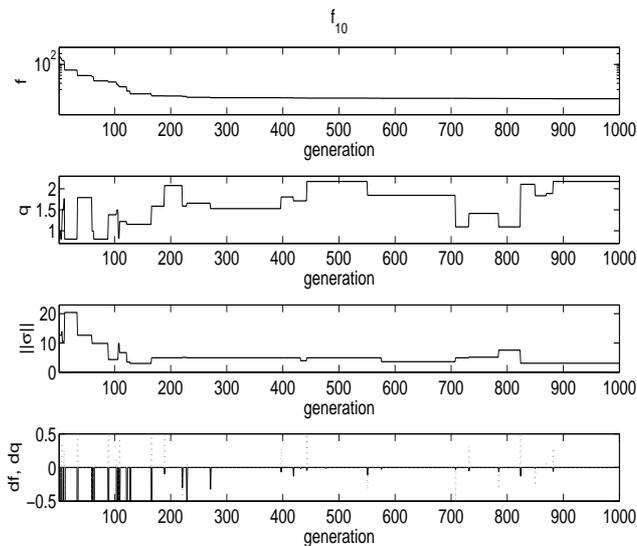
Fig. 6 $f(\mathbf{x}_{\mathbf{g}}^{\mathbf{best}}) - f(\mathbf{x}^*)$, distribution parameter q, and norm of the mutation strength parameter vector for the current best individual in the last run of IqGEP on function f_{10} with $m = 10$. The last graphic shows the differences dq (solid line), between the values of q in two consecutive generations, and df (dashed line), between the values of fitness errors in two consecutive generations, both in the range -0.5 to 0.5.

Finally, the algorithms with mutations generated from isotropic distributions present a better performance than the algorithms with anisotropic mutations on the unimodal functions (see Tables 10 and 11). The performance of the mutations generated from isotropic distributions is better than the anisotropic mutations on more multimodal functions with $m = 10$ than with $m = 30$. In the functions where the anisotropic mutations present a better performance, good solutions can be found in the subspace formed by the points close to the coordinate axes (Fig. 2) and the optimization can be done mainly by coordinate-wise search steps. Such results agree with the conjecture presented in [14] and affirm that anisotropic mutation distributions with longer tails are interesting on those multimodal functions with the low dimensional space where most of the longer jumps occur and better optima than the current best local solution may be located. When Cauchy and q-Gaussian mutations generated from isotropic distributions are employed, it is more difficult to reach a fair region of the search space from a long jump, especially when $m = 30$, as the number of points that can be reached exponentially increases with the size of the jump. That is, the sub-space that is more explored by the algorithm is much larger when mutation generated from isotropic distributions with longer tails are used.

However, mutation generated from isotropic distributions with longer tails are generally better than the

anisotropic mutation distributions on functions with heavy influence of the crossed component terms in the function evaluation, and where better optima are not located in the subspace explored by the anisotropic algorithm. These facts can be observed when the results of the experiments with the axis parallel Rastrigin function (f_9), which is separable, and the rotated Rastrigin function (f_{10}), where crossed component terms present a heavy influence in the function evaluation, are analyzed (Figs. 4 and 5). While the anisotropic Cauchy mutation presents a better performance than the q-Gaussian mutation on function f_9, they present a worse performance on function f_{10}. On the other hand, IqGEP presents a better performance than ICEP on both functions. In the experiments on function f_9, larger jumps to points close to the coordinate axes are advantageous for the population to escape from local optima. In the algorithms ICEP and IqGEP, where the mutations are generated from isotropic distributions, such jumps are rare, while they are common in the anisotropic algorithms CEP and qGEP. However, such jumps are less advantageous on f_{10}, where better optima are located far away from the coordinate axes. As the search space is much larger, it is more difficult to reach fair regions, though such regions can be eventually reached by longer jumps produced by ICEP and IqGEP. In general, for larger spaces, it would be more difficult to reach such fair regions if there are only a few of them. It is important to observe that these results depend on the fitness space, its size, and the number and size of the regions where better optima are located.

8.3 Comparison of restart q-Gaussian evolution strategy to other continuous EAs

The experimental results of the error of the best fitness averaged over 25 runs in the experiments with RqGES for $m = 10$, $m = 30$, and $m = 50$ are presented in Tables 13, 14, and 15. When the results of the RqGES are compared to those obtained by IqGEP, the benefits of using recombination and the restart scheme are clear, mainly on multimodal functions. The results for the fitness errors of RqGES can also be compared to those obtained for continuous optimization EAs applied to the problems described in Table 1.

In order to test the performance of RqGES against other EAs, the results obtained are compared to those obtained by the algorithms presented in the Special Session on Real Parameter Optimization for the 2005 IEEE Congress on Evolutionary Computation [35]: BLX-GL50 [10], BLX-MA [21], CoEVO [28], DE [32], DMS-L-PSO [19], EDA [46], G-CMA-ES [1], K-PCX [33], L-CMA-ES [2], L-SADE [29], and SPC-PNX [4]. Their re-

Table 13 Experimental results of the error of the best fitness obtained for algorithm RqGES on the test functions with $m = 10$.

Prob	median	mean	std
f_1	1.27E-08	3.77E-08	7.80E-08
f_2	2.38E-08	3.76E-08	5.58E-08
f_3	1.19E+05	1.55E+05	9.45E+04
f_4	2.84E-08	6.29E-08	8.39E-08
f_5	1.11E-04	1.05E-04	3.07E-05
f_6	1.02E+02	8.23E+02	2.06E+03
f_7	1.81E-08	1.40E-03	3.64E-03
f_8	2.04E+01	2.04E+01	5.95E-02
f_9	1.99E+00	1.49E+00	1.14E+00
f_{10}	1.92E+00	1.72E+00	1.21E+00
f_{11}	7.59E-03	1.69E-01	3.83E-01
f_{12}	6.39E+01	1.05E+03	2.15E+03
f_{13}	1.07E+00	1.09E+00	2.90E-01
f_{14}	2.90E+00	2.91E+00	2.63E-01
f_{15}	1.03E+02	1.98E+02	1.56E+02
f_{16}	9.40E+01	9.37E+01	7.20E+00
f_{17}	9.35E+01	9.59E+01	1.31E+01
f_{18}	8.00E+02	6.77E+02	2.71E+02
f_{19}	8.00E+02	5.88E+02	2.87E+02
f_{20}	8.00E+02	6.41E+02	2.85E+02
f_{21}	8.00E+02	8.45E+02	2.08E+02
f_{22}	7.67E+02	7.70E+02	1.62E+01
f_{23}	9.71E+02	9.45E+02	1.22E+02
f_{24}	2.00E+02	2.12E+02	6.00E+01
f_{25}	4.03E+02	3.47E+02	9.39E+01

Table 14 Experimental results of the error of the best fitness obtained for algorithm RqGES on the test functions with $m = 30$.

Prob	median	mean	std
f_1	3.65E-07	1.54E-06	5.31E-06
f_2	1.77E-06	1.74E-06	4.29E-07
f_3	8.55E+05	9.43E+05	3.22E+05
f_4	6.71E-02	4.58E-01	8.53E-01
f_5	2.41E+03	2.31E+03	5.57E+02
f_6	2.29E+02	1.75E+03	2.74E+03
f_7	1.12E-06	2.69E-03	4.36E-03
f_8	2.10E+01	2.10E+01	3.76E-02
f_9	9.95E+00	9.74E+00	3.71E+00
f_{10}	9.95E+00	9.24E+00	3.09E+00
f_{11}	3.71E+00	3.60E+00	1.92E+00
f_{12}	4.23E+03	8.90E+03	1.03E+04
f_{13}	3.03E+00	3.26E+00	7.49E-01
f_{14}	1.25E+01	1.24E+01	3.22E-01
f_{15}	3.70E+02	3.37E+02	9.37E+01
f_{16}	4.43E+01	1.77E+02	1.78E+02
f_{17}	3.07E+01	1.04E+02	1.14E+02
f_{18}	9.09E+02	8.99E+02	3.01E+01
f_{19}	9.07E+02	8.94E+02	3.54E+01
f_{20}	9.09E+02	9.04E+02	2.20E+01
f_{21}	5.00E+02	5.71E+02	2.18E+02
f_{22}	8.92E+02	8.91E+02	6.12E+00
f_{23}	5.34E+02	5.59E+02	1.29E+02
f_{24}	2.00E+02	2.00E+02	7.40E-04
f_{25}	2.00E+02	2.00E+02	2.13E+00

Table 15 Experimental results of the error of the best fitness obtained for algorithm RqGES on the test functions with $m = 50$.

Prob	median	mean	std
f_1	4.50E-06	7.21E-03	1.76E-02
f_2	1.85E-03	2.16E-03	1.08E-03
f_3	1.13E+06	1.13E+06	3.09E+05
f_4	6.82E+02	7.85E+02	5.64E+02
f_5	6.18E+03	6.38E+03	1.33E+03
f_6	3.91E+02	6.74E+02	8.36E+02
f_7	2.89E-06	1.43E-02	4.73E-02
f_8	2.11E+01	2.11E+01	3.49E-02
f_9	5.37E+01	5.89E+01	2.73E+01
f_{10}	2.39E+01	2.40E+01	6.88E+00
f_{11}	1.28E+01	1.28E+01	3.62E+00
f_{12}	3.64E+04	4.31E+04	3.73E+04
f_{13}	5.45E+00	5.63E+00	7.70E-01
f_{14}	2.18E+01	2.18E+01	2.13E-01
f_{15}	4.00E+02	3.53E+02	1.02E+02
f_{16}	2.67E+01	4.99E+01	7.72E+01
f_{17}	2.12E+01	3.20E+01	3.37E+01
f_{18}	9.00E+02	9.00E+02	3.20E-02
f_{19}	9.00E+02	8.81E+02	9.29E+01
f_{20}	9.00E+02	8.83E+02	9.37E+01
f_{21}	1.17E+03	8.79E+02	3.43E+02
f_{22}	9.37E+02	9.34E+02	1.17E+01
f_{23}	5.40E+02	6.57E+02	2.44E+02
f_{24}	2.00E+02	2.00E+02	6.53E-02
f_{25}	2.09E+02	3.20E+02	3.07E+02

in the special session were compared, G-CMA-ES was pointed as the algorithm with the best average performance [12].

Figures 7, 8, and 9 show the mean of the error $f(\mathbf{x^{best}}) - f(\mathbf{x}^*)$ (error between the best fitness found during the run and the fitness at the global optimum) for the test functions, respectively, with $m = 10$, $m = 30$, and $m = 50$ for the algorithms (including RqGES). The results shown in those figures were obtained in the respective papers (see last paragraph) and in Tables 13, 14, and 15 for RqGES. The results for RqGES with different values of m are still presented in Figure 10.

Tables 16, 17, 18 present the rank (for the performance regarding the mean best fitness error) for RqGES when compared to the results of other algorihtms. A rank i means that RqGES was the i-th best algorithm. When k algorithms present the same mean best fitness error values, the mean between the maximum and minimum rank for such group is considered and a "t" is indicated in the table (e.g., when RqGES and another algorithm presented the best performance, the rank was set to 1.5t).

Tables 19, 20, and 21 present the statistical comparison of RqGES to the other algorithms regarding the results of two groups of functions ($f_1 - f_{25}$ and the hybrid composition functions $f_{15} - f_{25}$), as in [12]. The p-value of the Wilcoxon Signed Rank Test presented in the tables is obtained comparing the vectors with the mean results obtained for each function for RqGES and Alg. A. The values in the second parentheses indicate the difference between the number of times (functions)

sults for the problems presented in Table 1 for $m = 10$ were avalaible in the respective papers[1]. Eight of those algorithms presented results for $m = 30$ and two algorithms also for $m = 50$. For all algorithms, the same maximum number of fitness evaluations and number of runs presented in Section 8.1 were considered. When the results for all functions of the algorithms presented

---

[1] Tables with the results and other relevant information about the Real Parameter Optimization Competition of the 2005 IEEE Congress on Evolutionary Computation can be found at ¡a href="http://sci2s.ugr.es/EAMHCO"¿http://sci2s.ugr.es/EAMHCO¡/a¿.

Fig. 7 Mean of $f(\mathbf{x}^{\mathbf{best}}) - f(\mathbf{x}^*)$ on the test functions with $m = 10$. The termination error value 10^{-8} is adopted as the low limit.
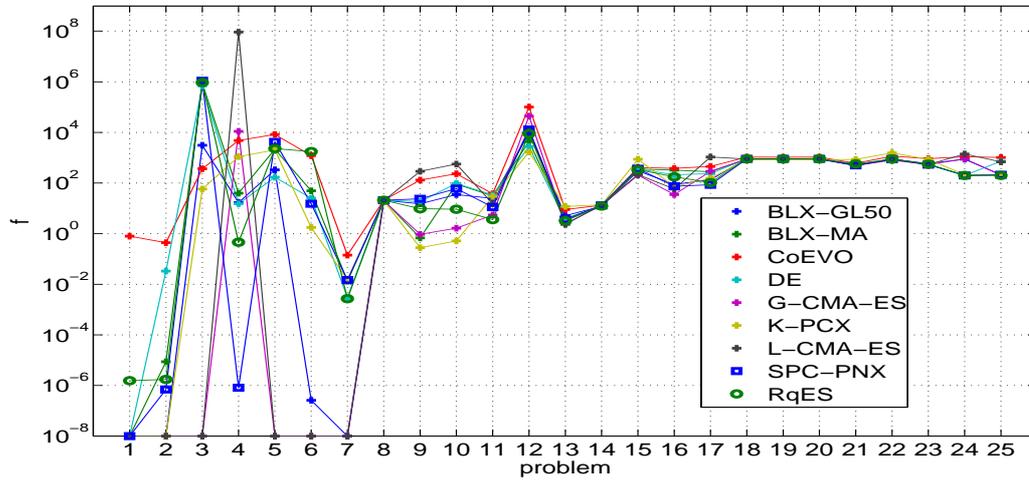

Fig. 8 Mean of $f(\mathbf{x}^{\mathbf{best}}) - f(\mathbf{x}^*)$ on the test functions with $m = 30$.
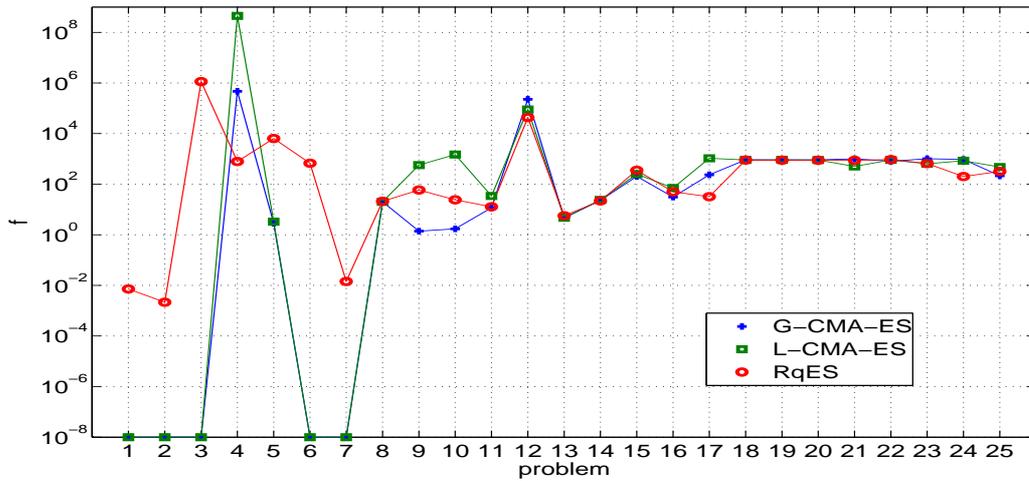

**Fig. 9** Mean of $f(\mathbf{x}^{\mathbf{best}}) - f(\mathbf{x}^*)$ on the test functions with $m = 50$.
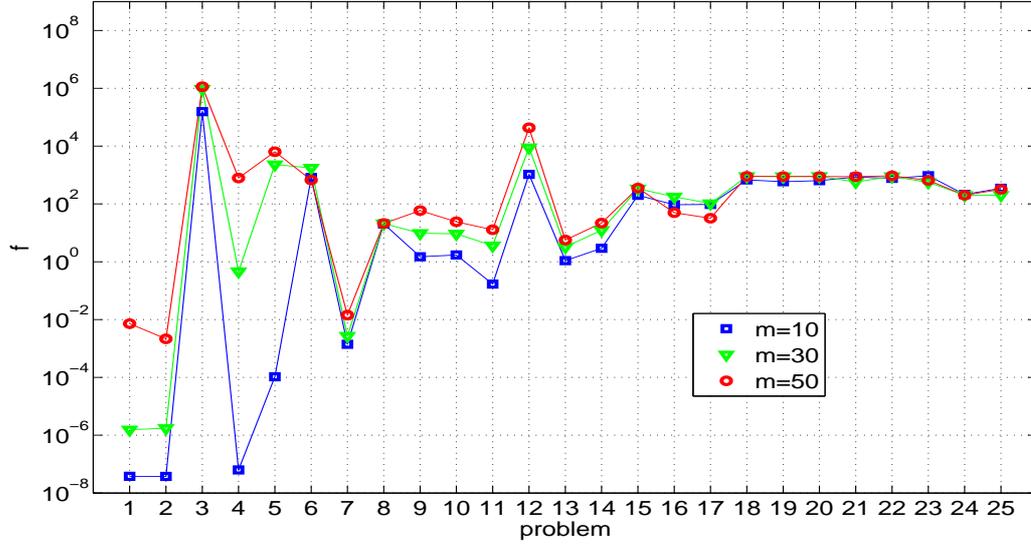
Fig. 10 Mean of $f(\mathbf{x}^{\mathbf{best}}) - f(\mathbf{x}^*)$ for RqGES on the test functions with $m = 10$, $m = 30$, and $m = 50$.

Table 16 Rank of the mean of the fitness error of RqGES (when compared to 11 algorithms: BLX-GL50, BLX-MA, CoEVO, DE, DMS-L-PSO, EDA, G-CMA-ES, K-PCX, L-CMA-ES, L-SADE, and SPC-PNX) on the test functions with $m = 10$.

Prob	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
Rank	12	12	12	8	8	12	3	10t	8
Prob	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
Rank	3	1	12	10	6	3	3	1	7
Prob	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}		
Rank	7	6	11	10	11	7	2		

Table 17 Rank of the mean of the fitness error of RqGES (when compared to 8 algorithms: BLX-GL50, BLX-MA, CoEVO, DE, G-CMA-ES, K-PCX, L-CMA-ES, and SPC-PNX) on the test functions with $m = 30$.

Prob	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
Rank	8	6	8	2	7	9	4	8	4
Prob	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
Rank	3	1	4	4	2	4	6	2	4
Prob	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}		
Rank	3	5	7	5	6	2.5t	1		

Table 18 Rank of the mean of the fitness error of RqGES (when compared to 2 algorithms: G-CMA-ES and L-CMA-ES) on the test functions with $m = 50$.

Prob	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
Rank	3	3	3	1	3	3	3	3	2
Prob	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}
Rank	2	2	1	3	1	3	2	1	1
Prob	f_{19}	f_{20}	f_{21}	f_{22}	f_{23}	f_{24}	f_{25}		
Rank	1	1	2	3	2	1	2		

Table 19 Statistical comparison of RqGES to other algorithms regarding the mean of the fitness error on the test functions with $m = 10$.

Alg.	$f_1 - f_{25}$	$f_{15} - f_{25}$
RqGES - BLX-GL50	2.99E-02 (s-),(-12)	2.06E-01 (−),(-5)
RqGES - BLX-MA	9.25E-01 (−),(-1)	2.06E-01 (+),(+5)
RqGES - CoEVO	4.43E-01 (+),(+5)	1.75E-01 (+),(+5)
RqGES - DE	1.10E-01 (−),(-8)	3.20E-01 (−),(-3)
RqGES - DMS-L-PSO	4.93E-01 (−),(-3)	8.98E-01 (+),(+1)
RqGES - EDA	5.45E-01 (−),(-3)	7.65E-01 (+),(+1)
RqGES - G-CMA-ES	5.81E-03 (s-),(-15)	6.54E-02 (−),(-5)
RqGES - K-PCX	3.39E-01 (+),(+3)	9.77E-03 (s+),(+9)
RqGES - L-CMA-ES	5.45E-01 (−),(-5)	7.65E-01 (−),(-1)
RqGES - L-SADE	4.43E-01 (−),(-1)	7.65E-01 (+),(+1)
RqGES - SPC-PNX	1.58E-01 (−),(-5)	1.75E-01 (−),(-3)

Table 20 Statistical comparison of RqGES to other algorithms regarding the mean of the fitness error on the test functions with $m = 30$.

Alg.	$f_1 - f_{25}$	$f_{15} - f_{25}$
RqGES - BLX-GL50	9.89E-01 (−),(-1)	1.00E+00 (+),(+1)
RqGES - BLX-MA	7.32E-01 (=),(0)	8.46E-01 (=),(0)
RqGES - CoEVO	1.57E-03 (s+),(+19)	9.77E-04 (s+),(11)
RqGES - DE	4.75E-01 (+),(+8)	1.93E-01 (+),(+6)
RqGES - G-CMA-ES	4.93E-01 (−),(-5)	8.98E-01 (+),(+1)
RqGES - K-PCX	8.61E-01 (−),(-1)	3.65E-01 (+),(+3)
RqGES - L-CMA-ES	9.68E-01 (−),(-5)	9.66E-01 (−),(-3)
RqGES - SPC-PNX	3.17E-01 (+),(+4)	4.92E-01 (=),(0)

Table 21 Statistical comparison of RqGES to other algorithms regarding the mean of the fitness error on the test functions with $m = 50$.

Alg.	$f_1 - f_{25}$	$f_{15} - f_{25}$
RqGES - G-CMA-ES	8.82E-01 (−),(-5)	3.65E-01 (+),(+3)
RqGES - L-CMA-ES	3.97E-01 (+),(+1)	4.65E-01 (+),(+3)

that the mean of the error for RqGES is smaller than the mean of the error for Alg. A and the number of

times (functions) that the mean of the error for Alg. A is smaller than the mean of the error for RqGES. A negative value indicates that Alg. A presented smaller mean (when compared to RqGES) on more functions.

One can observe that, for the experiments with $m = 10$, the performance of RqES was significantily worse than the performance of BLX-GL50 and G-CMA-ES for the group $f_1 - f_{25}$, while it was significantily better than the performance of K-PCX for the group $f_{15} - f_{25}$ (Table 19). For the experiments with $m = 30$, RqES was significantily better than CoEVO in both groups (Table 20). In general, RqES presented better performance for high dimension and multimodal functions.

It can be observed (Tables 16, 17, and 18 and Figures 7, 8, and 9) that RqGES presents good results on the functions with noise (functions f_4 and f_{17}). In the multimodal function f_{17}, RqES was the best algorithm for $m = 10$ and $m = 50$, while it was the second best algorithm for $m = 30$ (one can observe in figures 8, and 9 that the performance of RqGES, differently for the other algorithms, did not deteriorated from function f_{16} to function f_{17}, obtained by adding noise to function f_{16}). In the unimodal function f_4, RqES is the best algorithm for $m = 50$ and the second best for $m = 30$. RqGES presents the best results on function f_{11} (with the exception for $m = 50$ where RqGEs is the second best algorithm), which is differentiable only on a set of points [35]. The properties of function f_{11} can also explain the better results of RqGES on the hybrid composition functions where function f_{11} was used.

Such properties (noise and rugged landscapes) can represent obstacles for algorithms that intensivelly explore local information about the search landscape to guide the optimization process, like some of the best EAs used in continuous optimization problems. The good performance of RqGES on such problems can be explained by the eventual use of higher values of q to produce larger jumps on the search landscape. However, eventual large jumps makes RqGES to generate less candidate solutions, when compared to other algorithms, close to the best found solutions, which explains the worse performance of RqGES on the unimodal functions (with exception for function f_4). Besides, RqGES does not employ additional mechanisms to explore local information about the search space, what explains the worse performance, when compared to the best algorithms, in unimodal and multimodal functions where such information is useful to guide the optimization process.

On function f_{25}, higher values of q were useful too, as the global optimum is outside the inicialization range. It can also be observed that RqGES has a good performance on rotated functions, e.g., f_{10}, because the

q-Gaussian mutation is generated from an isotropic distribution. One can observe, in figures 8 and 9, that RqGES is the only algorithm where the performance did not deteriorate when function f_9 was rotated to generate function f_{10}. In the figures, one can observe that the error for f_{10} is higher than for f_9 for all algorithms, except for RqGES.

9 Conclusion and future work

The use of self-adaptation is proposed in this paper to control not only the mutation strength parameter, but also the mutation distribution for EAs. For this purpose, self-adapted q-Gaussian mutations generated from anisotropic and isotropic distributions are employed. The q-Gaussian distribution allows to smoothly change the shape of the mutation distribution by setting a real parameter q and can reproduce either finite or infinite second moment distributions. In the proposed method, the real parameter q of the q-Gaussian distribution, which defines the shape of the distribution employed by the mutation operator, is encoded in the chromosome of the individual and is allowed to evolve.

In the proposed method, the decision of choosing which distribution is more indicated for a given problem and at a given moment of the evolutionary process is made by the algorithm. This property can be observed in the experimental results presented in Section 8, where the proposed q-Gaussian mutation generally presents a performance similar to or better than the Gaussian mutation when the Gaussian mutation is better than the Cauchy mutation and generally presents a performance similar to or better than the Cauchy mutation when it is better than the Gaussian mutation. Generally speaking, the experimental results indicate the efficiency of the proposed self-adaptation scheme.

In this paper, EP was selected to compare the Gaussian, Cauchy, and q-Gaussian mutations because it only uses mutation as a transformation operator. In this way, it is easier to analyse the influence of each type of mutation operators. In fact, the q-Gaussian mutation can be used in other types of EAs. The results for the EP with q-Gaussian mutaion can be improved if recombination and other heuristics are used, which was shown by the experimental results of the restart q-Gaussian ES presented in this paper. RqGES presented competitive performance when compared to some of the best EAs for real-parameter optimization. The possible use of the q-Gaussian mutation in ESs with adaptive enconding [15], which is a form of applying the representation changes given by the covariance matrix adaptation (CMA) in the continuous domain, is a relevant

future work. Other future works include the investigation of other control methods for the q parameter (including self-organization [38]), and the use of the q-Gaussian mutation for dynamic optimization problems [43], like the optimization of synaptic weights in evolutionary neural networks in dynamic environments [37].

References

1. A. Auger and N. Hansen. A restart CMA evolution strategy with increasing population size. In *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, pp. 1769–1776, 2005.

2. A. Auger and N. Hansen. Performance evaluation of an advanced local search evolutionary algorithm. In *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, pp. 1777–1784, 2005.

3. T. Bäck. Self-adaptation. In T. Bäck, D. B. Fogel, and Z. Michalewicz, editors, *Evolutionary computation 2: advanced algorithms and operators*. Institute of Physicis Publishing, 2000.

4. P. J. Ballester, J. Stephenson, J. N. Carter, K. Gallagher. Real-parameter optimization performance study on the CEC-2005 benchmark with SPC-PNX. In *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, pp. 498–505, 2005.

5. H.-G. Beyer and H. S. Schwefel. Evolution strategies: a comprehensive introduction. *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

6. S. Das, A. Abraham, U. K. Chakraborty, and A. Konar. Differential evolution using a neighborhood-based mutation operator. *IEEE Trans. on Evol. Comput.*, 13(3): 526–553, 2009.

7. M. W. Davis. The natural formation of gaussian mutation strategies in evolutionary programming. In *Proc. of the 3rd Annual Conf. on Evolutionary Programming*. World Scientific, 1994.

8. H. Dong, J. He, H. Huang, and W. Hou. Evolutionary programming using a mixed mutation strategy. *Information Sciences*, vol. 177, no. 1, pp. 312–327, 2007.

9. A. E. Eiben, R. Hinterding, and Z. Michalewicz. Parameter control in evolutionary algorithms. *IEEE Trans. on Evol. Comput.*, vol. 3, no. 2, pp. 124–141, 1999.

10. C. García-Martínez and M. Lozano. Hybrid real-coded genetic algorithms with female and male differentiation. In *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, pp. 896–903, 2005.

11. C. García-Martínez, M. Lozano, F. Herrera, D. Molina, and A. M. Sánchez. Global and local real-coded genetic algorithms based on parent-centric crossover operators. *European Journal of Operational Research*, 185(3): 1088–1113, 2008.

12. S. García, D. Molina, M. Lozano, and F. Herrera. A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization. *Journal of Heuristics*, vol. 15, pp. 617–644, 2009.

13. N. Hansen and A. Ostermeier. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.*, vol. 9, no. 2, pp. 159–195, 2001.

14. N. Hansen, F. Gemperle, A. Auger, and P. Koumoutsakos. When do heavy-tail distributions help? *Proc. 9th Int. Conf. on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, vol. 4193, pp. 62–71, 2006.

15. N. Hansen. Adaptive encoding: how to render search coordinate system invariant *Proc. 10th Int. Conf. on Parallel Problem Solving from Nature*, Lecture Notes in Computer Science, vol. 5199, pp. 205–214, 2008.

16. F. Herrera, M. Lozano, and J. L. Verdegay. Tackling real-coded genetic algorithms: operators and tools for the behavioral analysis. *Artificial Intelligence Reviews*, 12(4): 265–319, 1998.

17. M. Iwamatsu. Generalized evolutionary programming with levy-type mutation. *Computer Physics Communications*, 147(1-2): 729–732, 2002.

18. C. Y. Lee and X. Yao. Evolutionary programming using mutations based on the levy probability distribution. *IEEE Trans. on Evol. Comput.*, 8(1): 1–13, 2004.

19. J. J. Liang and P. N. Suganthan. Dynamic multi-swarm particle swarm optimizer with local search. In *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, pp. 522–528, 2005.

20. M. Lozano, F. Herrera, N. Krasnogor, and D. Molina. Real-coded memetic algorithms with crossover hill-climbing. *Evol. Comput.*, 12 (3): 273–302, 2004.

21. D. Molina, F. Herrera, and M. Lozano. Adaptive local search parameters for real-coded memetic algorithms. In *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, pp. 888–895, 2005.

22. E. Mezura-Montes and C. A. C Coello. An improved diversity mechanism for solving constrained optimization problems using a multimembered evolution strategy. In *Proc. of the 2004 Genetic and Evol. Comput. Conf. (GECCO-2004)*, pp. 700–712, 2004.

23. M. A. Moret, P. G. Pascutti, P. M. Bisch, M. S. P. Mundim, and K. C. Mundim. Classical and quantum conformational analysis using generalized genetic algorithm. *Physica A: Statistical Mechanics and its Applications*, 363(2): 260–268, 2006.

24. Q. H. Nguyen, Y. S. Ong, and M. H. Lim. A probabilistic memetic framework. *IEEE Trans. on Evol. Comput.*, 13(3): 604–623, 2009.

25. N. Noman, and H. Iba. Accelerating differential evolution using an adaptive local search. *IEEE Trans. on Evol. Comput.*, 12(1): 107–125, 2008.

26. A. Obuchowicz. Multidimensional mutations in evolutionary algorithms based on real-valued representation. *Int. Journal of Systems Science*, 34(7): 469–483, 2003.

27. K. V. Price, R. M. Storn, and J. A. Lampinen. *Differential evolution: a practical approach to global optimization*. Springer Verlag, 2005.

28. P. Pošík. Real-parameter optimization using the mutation step co-evolution. In *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, pp. 872–879, 2005.

29. A. K. Qin, and P. N. Suganthan. Self-adaptive differential evolution algorithm for numerical optimization. In *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, pp. 1785–1791, 2005.

30. A. K. Qin, V. L. Huang, and P. N. Suganthan. Differential evolution algorithm with strategy adaptation for global numerical optimization. *IEEE Trans. on Evol. Comput.*, 13(2): 298–417, 2009.

31. P. N. Rathie and S. Da Silva. Shannon, Lévy, and Tsallis: A Note. *Applied Mathematical Sciences*, 2(28): 1359–1363, 2008.

32. J Rönkkönen, S. Kukkonen, and K. V. Price. Real-parameter optimization with differential evolution. In *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, pp. 506–513, 2005.

33. A. Sinha, S. Tiwari and K. Deb. A population-based, steady-state procedure for real-parameter optimization. In *Proc. of the 2005 IEEE Congress on Evolutionary Computation*, pp. 514–521, 2005.

34. A. M. C. Souza and C. Tsallis. Student's t- and r-distributions: unified derivation from an entropic variational principle. *Physica A: Statistical Mechanics and its Applications*, 236(1-2): 52–57, 1997.

35. P. N. Suganthan, N. Hansen, J. J. Liang, K. Deb, Y. P. Chen, A. Auger, and S. Tiwari. Problem definitions and evaluation criteria for the CEC 2005 special session on real parameter optimization. *Technical Report*, Nanyang Technological University, 2005.

36. W. Thistleton, J. A. Marsh, K. Nelson, and C. Tsallis, "Generalized Box-Muller method for generating q-Gaussian random deviates," *IEEE Trans. on Information Theory*, vol. 53, no. 12, pp. 4805–4810, 2007.

37. R. Tinós and A. C. P. L. F. Carvalho. Use of gene dependent mutation probability in evolutionary neural networks for non-stationary problems. *Neurocomputing*, 70(1-3): 44–54, 2006.

38. R. Tinós and S. Yang. Self-organizing random immigrants genetic algorithm for dynamic optimization problems. *Genetic Programming and Evolvable Machines*, 8(3): 255–286, 2007.

39. C. Tsallis. Possible generalization of boltzmann-gibbs statistics. *Journal of Statistical Physics*, 52: 479–487, 1988.

40. C. Tsallis and D. A. Stariolo. Generalized simulated annealing. *Physica A:*, 233(1-2): 395–406, 1996.

41. S. Umarov, C. Tsallis, and S. Steinberg. On a q-central limit theorem consistent with nonextensive statistical mechanics. *Milan Journal of Mathematic*, 76(1): 307–328, 2008.

42. J. A. Vrugt, B. A. Robinson, and J. M. Hyman. Self-adaptive multimethod search for global optimization in real-parameter spaces. *IEEE Trans. on Evol. Comput.*, 13(2): 243–259, 2009.

43. H. Wang, S. Yang, W. H. Ip, and D. Wang. Adaptive primal-dual genetic algorithms in dynamic environments. *IEEE Trans. on Syst., Man, and Cybern., Part B: Cybern.*, 39(6): 1348–1361, 2009.

44. X. Yao and Y. Liu. Fast evolution strategies. *Control and Cybernetics*, 26(3): 467–496, 1997.

45. X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *IEEE Trans. on Evol. Comput.*, 3(2): 82–102, 1999.

46. B. Yuan and M. Gallagher. Experimental results for the special session on real-parameter optimization at CEC 2005: A simple, continuous EDA. In *Proc. of the 2005 IEEE Congress on Evol. Comput.*, pp. 1792–1799, 2005.