

Self-Adaptation of Mutation Distribution in Evolutionary Algorithms

Renato Tinós and Shengxiang Yang

Abstract—This paper proposes a self-adaptation method to control not only the mutation strength parameter, but also the mutation distribution for evolutionary algorithms. For this purpose, the isotropic q -Gaussian distribution is employed in the mutation operator. 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 or infinite second moment distributions. In the proposed method, the real parameter q of the q -Gaussian distribution is encoded in the chromosome of an individual and is allowed to evolve. An evolutionary programming algorithm with the proposed idea is presented. Experiments were carried out to study the performance of the proposed algorithm.

I. INTRODUCTION

Stochastic search methods have been successfully applied to a large number of optimization problems. One major characteristic of stochastic search methods is that new candidate solutions are randomly generated from a given probability distribution. Evolutionary algorithms (EAs) are among the most known stochastic search methods. EAs are a class of meta-heuristic algorithms that are inspired by the principles of natural evolution.

When EAs are applied in real-valued optimization, new candidate solutions are traditionally generated employing multivariate samples taken from isotropic Gaussian distributions [1]. The use of an isotropic Gaussian distribution is interesting mainly because it 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 unconstrained real-valued search spaces [10]. The isotropic Gaussian distribution has a finite second moment and does not favour 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 infinite second moment in EAs. For example, in the Fast Evolutionary Programming (FEP) [15], the Cauchy distribution is employed, while in the Evolutionary Programming with Lévy mutation (LEP) [6], mutation based on Lévy distribution is used. The Lévy distribution is a class of probability distributions with infinite second moment, which includes the Cauchy distribution, and allows to control the tail of the

distribution by changing a scalar parameter α . In [6], 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.

The use of mutation taken from heavy tail distributions implies jumps of scale-free sizes, allowing to reach faster distant regions of the search space. This property is interesting when EAs are applied to multimodal problems or dynamic optimization problems as it allows 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 [4]. Most of the proposed algorithms use heavy tail distributions that are anisotropic, i.e., some directions of the mutation are privileged in the search space [8]. Moreover, in several fitness landscapes, it is difficult to reach a fair region of the search space from a long jump because the probability to reach a point with a lower fitness is generally much larger for a long jump [4].

There are mainly three classes of mutation strength parameter control in the literature [2]: 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. In most cases, only one mutation strength parameter is updated for all individuals, like in the 1/5 success rule for evolution strategies [1]. In some cases, one mutation strength parameter for each individual [2] is changed.

In this paper, we propose to control the mutation distribution along the evolutionary process, instead of only controlling the mutation strength parameter that defines the spread of a fixed distribution. Here, self-adaptation is employed, not only to control the mutation strength parameter, but also to control the mutation distribution. In this way, the decision of which distribution to choose is more indicated for a given problem and, at a given moment of the evolutionary process, is minimized by letting the proposed algorithm to decide which mutation distribution should be used. In the proposed algorithm, a real parameter that defines the distribution employed by the mutation operator is encoded in the chromosome of an individual and is allowed to evolve. For this purpose, the isotropic q -Gaussian distribution [10] derived from the Tsallis generalized entropy [13] is employed. The q -Gaussian distribution allows to control the shape of the distribution by setting a real parameter q and

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

Shengxiang Yang is with the Department of Computer Science, University of Leicester, University Road, Leicester, LE1 7RH, United Kingdom, (email: s.yang@mcs.le.ac.uk).

can reproduce either finite second moment distributions, like the Gaussian distribution, or infinite second moment distributions, like the heavy tail Lévy distribution. The use of the q -Gaussian mutation distribution in EAs is not new [7], [5]. However, in such algorithms like most other algorithms that uses heavy tail distributions, the new candidate solutions are produced by generating random deviates for each coordinate of the individual, which implies anisotropic distributions. Moreover, in such algorithms, the parameter q , which defines the shape of the distribution, is fixed along the generations or starts with a large value and decreases along the generations, like the temperature control in simulated annealing. In this way, the main contributions proposed in this paper are: (i) a q -Gaussian isotropic distribution is employed in the mutation operator, (ii) self-adaptation is employed to control the parameter q , which allows changing the shape of the distribution during the evolutionary process.

The rest of this paper is organized as follows. The q -Gaussian distribution is briefly introduced in Section II. In Section III, self-adaptation of the mutation distribution is proposed, while an evolutionary programming algorithm based on this idea is presented in Section IV. Experimental study with a set of test functions is presented in Section V. Finally, this article is concluded in Section VI.

II. 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

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

which 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 [10]. In this concern, Tsallis [13] proposed a generalized entropy form as follows:

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

where $q \in \mathbb{R}$. Eq. 2 recovers the entropy form given by Eq. 1 in the limit $q \rightarrow 1$. 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 for $q = 1$. 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 $q = (3 + m)/(1 + m)$ and $0 < m < \infty$, it becomes a Student's t -distribution with m degrees of freedom [9].

When $-\infty < q < 3$, the q -Gaussian distribution density [10] is given by

$$p(x; \bar{\mu}_q, \bar{\sigma}_q) = A_q \sqrt{B_q} e_q^{-B_q(x - \bar{\mu}_q)^2}, \quad (3)$$

where e_q^x is the q -exponential function defined as

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

which reduces to the usual exponential function when $q = 1$. The q -mean and the q -variance are defined as

$$\bar{\mu}_q \equiv \frac{\int xp(x)^q dx}{\int p(x)^q dx} \quad (5)$$

$$\bar{\sigma}_q^2 \equiv \frac{\int (x - \bar{\mu}_q)^2 p(x)^q dx}{\int p(x)^q dx} \quad (6)$$

and respectively reduce to the usual mean and variance when $q = 1$. In Eq. 3, A_q is the normalization factor [10] and B_q controls the width of the q -Gaussian distribution and is given by

$$B_q = \left((3 - q)\bar{\sigma}_q^2\right)^{-1} \quad (7)$$

A random variable x taken from a q -Gaussian distribution with q -mean $\bar{\mu}_q$ and q -variance $\bar{\sigma}_q^2$ is denoted by $x \sim \mathcal{N}_q(\bar{\mu}_q, \bar{\sigma}_q)$. In this paper, the generalized Box-Müller method proposed in [10] is employed to generate q -Gaussian random variables $x \sim \mathcal{N}_q(0, 1)$. The generalized Box-Müller method is very simple (see its pseudo-code in [10]) and allows to generate q -Gaussian distributions for $-\infty < q < 3$. Fig. 1 presents the empirical cumulative q -Gaussian distribution function for some values of q . It can be observed that larger values of q result in longer tails of the q -Gaussian distribution.

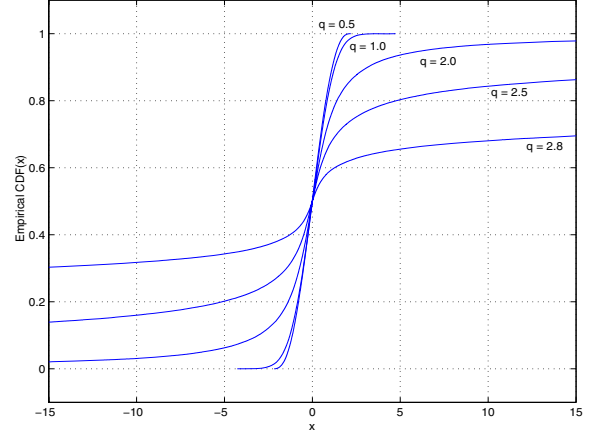


Fig. 1. Empirical cumulative distribution function (CDF) of 100000 observations of a q -Gaussian random variable $x \sim \mathcal{N}_q(0, 1)$ for: $q = 0.5$, $q = 1.0$ (Gaussian), $q = 2.0$ (Cauchy), $q = 2.5$, and $q = 2.8$.

III. SELF-ADAPTATION OF THE MUTATION DISTRIBUTION

In the m -dimensional real-valued search space, a new candidate solution is generated by the EA's mutation operator from the individual \vec{x}_i , where $i = 1, \dots, \mu$, as follows:

$$\vec{x}_i = \vec{x}_i + \mathbf{C} \vec{z}, \quad (8)$$

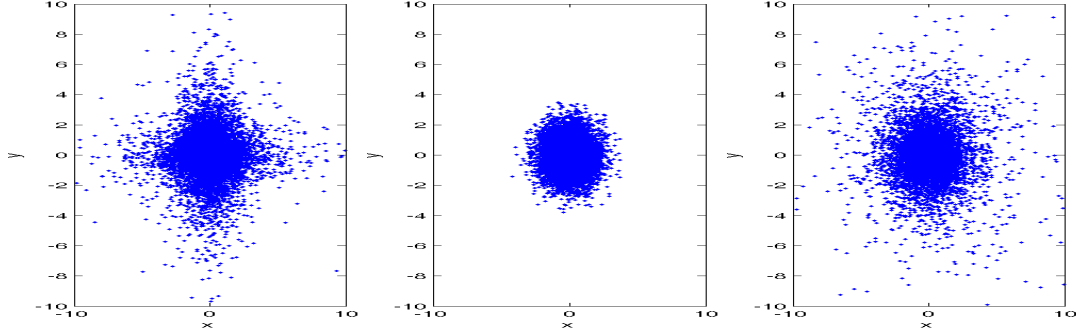


Fig. 2. Two-dimensional points from: random vector $\vec{z} \sim \mathcal{M}_q^m$ with anisotropic q -Gaussian distribution and $q = 1.5$, random vector $\vec{z} \sim \mathcal{N}^m$ with Gaussian distribution, and random vector $\vec{z} \sim \sqrt{2}\mathcal{N}_q^m$ with isotropic distribution q -Gaussian and $q = 1.5$.

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

$$\mathbf{C} = \sigma \mathbf{I}, \quad (9)$$

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

$$\mathbf{C} = \text{diag}(\vec{\sigma}) \quad (10)$$

i.e., \mathbf{C} is diagonal matrix with the main diagonal composed by the elements of the vector $\vec{\sigma} = [\sigma(1) \ \sigma(2) \ \dots \ \sigma(m)]^T$. In the most general situation, e.g., in the covariance matrix adaptation evolution strategy (CMA-ES) [3], \mathbf{C} is an orthogonal rotation matrix which indicates the correlation between the components of \vec{z} .

In EAs, the use of a multivariate Gaussian distribution is generally employed to generate the m -dimensional vector \vec{z} [1]. Here, an m -dimensional random vector generated from the multivariate Gaussian distribution is denoted by $\vec{z} \sim \mathcal{N}^m$. A Gaussian random vector \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 heavy tail distribution, some directions in the search space are more explored than others, i.e., the distribution is anisotropic.

To the best of the authors' knowledge, all stochastic search algorithms with q -Gaussian mutation, like the Generalized Simulated Annealing [12] and the Generalized Genetic Algorithm [7], make use of anisotropic multivariate q -Gaussian distributions. Most mutation operators for EAs that are based on heavy tail distributions, e.g., Fast Evolution Strategies [14], FEP [15], and LEP [6], make use of anisotropic multivariate distributions too. The use of random variables generated by sampling independent random variables taken from a heavy tail distribution is interesting for optimization

problems with separable evaluation functions, as most of the large steps occur close to the coordinate axis [8], [10] and the optimization can be solved by m one-dimensional optimization process parallel to the coordinate axes. However, the performance of the optimization process can be strongly affected for non-separable evaluation functions.

Obuchowicz [8] proposed a method to generate the random mutation vector \vec{z} with an isotropic Cauchy distribution. For this purpose, the random mutation vector \vec{z} is generated with: *i*) a random direction uniformly distributed on the surface of the m -dimensional unit hypersphere, and *ii*) an Euclidean norm obtained from a Cauchy distribution.

Based on the work in [8] and [10], we propose to generate the random mutation vector \vec{z} from an isotropic q -Gaussian distribution as follows:

$$\vec{z} \sim r\vec{u}, \quad (11)$$

where $r \sim \mathcal{N}_q(0, 1)$, i.e., a random variable with q -Gaussian distribution, and \vec{u} is a uniform random vector obtained by sampling a random vector with Gaussian distribution and normalizing it to length one, i.e., $\vec{u} = \vec{v}/\|\vec{v}\|$ where $\vec{v} \sim \mathcal{N}^m$ and $\|\vec{v}\|$ denotes the Euclidean norm of the vector \vec{v} . In this paper, an m -dimensional random vector generated from the multivariate isotropic q -Gaussian distribution is denoted by $\vec{z} \sim \mathcal{N}_q^m$, while an m -dimensional random vector generated by sampling m independent q -Gaussian random variables $\mathcal{N}_q(0, 1)$, i.e., with the multivariate anisotropic q -Gaussian distribution, is denoted by $\vec{z} \sim \mathcal{M}_q^m$. Fig. 2 presents two-dimensional multivariate samples obtained from anisotropic q -Gaussian distribution, Gaussian distribution, and isotropic q -Gaussian distribution. 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 in the interval $1 < q < 3$.

We also propose to self-adapt the parameter q which defines the shape of the distribution. Based on the mutation strength self-adaptation [1], we propose to multiplicatively

Algorithm 1 EP with the isotropic q -Gaussian mutation (Alg. qGEP)

- 1: Initialize the population composed of individuals $(\vec{x}_i, \vec{\sigma}_i, q_i)$ for $i = 1, \dots, \mu$
 - 2: **while** (stop criteria are not satisfied) **do**
 - 3: **for** $i \leftarrow 1$ to μ **do**
 - 4: $\tilde{\sigma}_i(j) = \sigma_i(j) \exp(\tau_b \mathcal{N}(0, 1) + \tau_c \mathcal{N}(0, 1)_j)$ for $j = 1$ to m
 - 5: $\tilde{q}_i = q_i \exp(\tau_a \mathcal{N}(0, 1))$
 - 6: $\vec{\tilde{x}}_i \leftarrow \vec{x}_i + \mathbf{C}_i \mathcal{N}_q^m$, where \mathcal{N}_q^m is a random vector generated from an isotropic q -Gaussian distribution with parameter \tilde{q}_i and $\mathbf{C}_i = \text{diag}(\vec{\tilde{\sigma}}_i^T)$
 - 7: **end for**
 - 8: Compute the fitness of the parents $(\vec{x}_i, \vec{\sigma}_i, q_i)$ and offspring $(\vec{\tilde{x}}_i, \vec{\tilde{\sigma}}_i, \tilde{q}_i)$ for $i = 1, \dots, \mu$
 - 9: Compute the winning function [6] of the population composed of μ parents and μ offspring
 - 10: Select, to compose the new population, the μ individuals with the largest winning function from the population composed of μ parents and μ offspring
 - 11: **end while**
-

TABLE I
TEST FUNCTIONS.

Function $f(\vec{x})$	Range	Dimension (m)
$f_a = \sum_{j=1}^m x_j^2$	$\vec{x} \in [-100, 100]^m$	30
$f_b = \sum_{j=1}^m (\sum_{i=1}^j x_i)^2$	$\vec{x} \in [-100, 100]^m$	30
$f_c = 1 + \frac{1}{4000} \sum_{j=1}^m x_j^2 - \prod_{j=1}^m \cos\left(\frac{x_j}{\sqrt{j}}\right)$	$\vec{x} \in [-600, 600]^m$	30
$f_d = 4x_1^2 - 2.1x_1^4 + x_1^6/3 + x_1x_2 - 4x_2^2 + 4x_2^2$	$\vec{x} \in [-5, 5]^m$	2
$f_e = \sum_{j=1}^m (x_j^2 - 10 \cos(2\pi x_j) + 10)$	$\vec{x} \in [-5, 5]^m$	10
$f_f = \sum_{j=1}^m (y_j^2 - 10 \cos(2\pi y_j) + 10)$, where $\vec{y} = \mathbf{M}(\vec{x})$	$\vec{x} \in [-5, 5]^m$	10

update the parameter q in individual i as follows:

$$\tilde{q}_i = q_i \exp(\tau_a \mathcal{N}(0, 1)), \quad (12)$$

where τ_a denotes the standard deviation of the Gaussian distribution.

IV. EVOLUTIONARY PROGRAMMING ALGORITHM WITH SELF-ADAPTATION OF THE MUTATION DISTRIBUTION

Self-adaptation of the mutation distribution, as presented in last section, can be applied in EAs with real-valued representation. Here, the evolutionary programming (EP) algorithm presented in Alg. 1, called $qGEP$, is used to test the ideas proposed in this paper. The main difference of the EP algorithm presented in Alg. 1 from Gaussian EP, FEP, and LEP [6] is that, in the proposed algorithm, the isotropic q -Gaussian mutation is employed (step 6) instead of the Gaussian (Gaussian EP), Cauchy (FEP), or Lévy (GEP) mutation, and a procedure to adapt the q parameter is adopted (step 5).

As suggested by theoretical and empirical work [1], the parameters τ_b and τ_c are defined here as:

$$\tau_b = \frac{1}{\sqrt{2m}} \quad (13)$$

$$\tau_c = \frac{1}{\sqrt{2\sqrt{m}}} \quad (14)$$

In this work, we propose to define the parameter τ_a as

$$\tau_a = \frac{1}{\sqrt{m}} \quad (15)$$

V. EXPERIMENTAL STUDY

In order to evaluate the proposed EP algorithm with self-adaptation of the mutation distribution, experiments with six test functions were carried out. The results of three approaches to define the parameter q in the EP algorithm (step 5 in Algorithm 1) are presented here. Each approach defines one algorithm as follows.

- Algorithm GEP (EP with Gaussian mutation): $q = 1$ for all individuals, i.e., Gaussian mutation is employed;
- Algorithm ICEP (EP with isotropic Cauchy mutation): $q = 2$ for all individuals, i.e., isotropic Cauchy mutation is employed;
- Algorithm qGEP (EP with isotropic q -Gaussian mutation): one changing q for each individual, i.e., Algorithm 1 (with step 5) described in last section is employed.

The six test functions used in this work are described in Section V-A. The experimental design is presented in Section V-B while the experimental results are presented and analyzed in Section V-C.

A. Test Problems

Six functions (see Table I) are selected as the test suite for the EP algorithms. The optimization process in this work is a minimization task. In Table I, the matrix \mathbf{M} is obtained by the

orthogonalization of a random matrix uniformly distributed in the unit hypersphere.

While the functions f_a and f_b are unimodal, the function f_d has a few local minima and the remaining functions are highly multimodal [6]. The functions f_e and f_f are, respectively, the axis parallel and rotated Rastrigin functions. The axis parallel Rastrigin function is separable, while the rotated Rastrigin function is not separable.

B. Experimental Design

In order to compare the algorithms, each one was executed 30 times (with 30 random seeds) for each test function described in Section V-A. For each run of an algorithm, the individuals of the initial population were randomly chosen. The population size (μ) was set to 100 individuals and the tournament size was set to 10. The initial mutation strength parameter $\sigma_i(j)$ is equal to $3.0\sqrt{m}$ and the initial q -Gaussian parameter q for Algorithm qGEP is equal to 1.0, i.e., the initial q -Gaussian distribution reproduces the Gaussian distribution. In Algorithm qGEP, the minimum and maximum values of the q -Gaussian parameter q are 0.9 and 2.5 respectively. In all algorithms, the minimum allowed value of the mutation strength parameter $\sigma_i(j)$ is 0.01.

C. Experimental Results

The experimental results of the fitness of the best individual in the last generation averaged over 30 runs in the experiments are presented in Table II. In Table III, the statistic comparison regarding Alg. qGEP - Alg. GEP, and Alg. qGEP - Alg. ICEP is carried out by t-test with 58 degrees of freedom at a 0.1 level of significance regarding the fitness of the best individual in the last generation.

TABLE II
EXPERIMENTAL RESULTS OF THE BEST-OF-GENERATION FITNESS OBTAINED IN THE LAST GENERATION.

problem	Alg. GEP	Alg. ICEP	Alg. qGEP
f_a	9.0000E-007	2.0667E-006	9.3333E-007
f_b	2.6400E-005	4.4800E-005	2.7533E-005
f_c	3.2790E-001	1.3866E-001	1.2982E-001
f_d	-1.0308E+000	-1.0295E+000	-1.0305E+000
f_e	8.4182E+000	8.8492E+000	9.0551E+000
f_f	1.3593E+001	1.2599E+001	1.3476E+001

TABLE III
STATISTICAL COMPARISON OF THE BEST-OF-GENERATION FITNESS IN THE LAST GENERATION FOR ALG. qGEP. THE RESULT REGARDING ALG. X IS SHOWN AS "+", "-", OR "~" WHEN ALG. qGEP IS, RESPECTIVELY, SIGNIFICANTLY BETTER THAN, SIGNIFICANTLY WORSE THAN, OR STATISTICALLY EQUIVALENT TO ALG. X.

problem	Alg. GEP	Alg. ICEP
f_a	~	+
f_b	~	+
f_c	+	~
f_d	~	+
f_e	~	~
f_f	~	~

Fig. 3 shows the results of the mean best-of-generation fitness in the experiments. The mean Euclidean norm of the mutation strength parameter vector and the mean distribution parameter q of the current best individual are presented in Fig. 4. From these tables and figures, some results can be observed.

First, it is observable from Figs. 3(a) and (b) that the algorithm with a Gaussian mutation (Alg. GEP) has a better performance in the minimization of functions f_a and f_b , when compared to the algorithm with an isotropic Cauchy mutation (Alg. ICEP). A similar result was reported when the Gaussian EP was compared to FEP [15], which uses an anisotropic Cauchy distribution, and LEP [6] in the minimization of f_a . In an unimodal function, like f_a and f_b , long jumps, which often occur when the isotropic Cauchy mutation is employed, generally causes a degradation in the performance of the algorithm in a later stage of the evolution as less offspring are generated to explore the local neighborhood.

It can be observed that the performance of the proposed algorithm (Alg. qGEP), in the minimization of functions f_a and f_b , is better when compared to the algorithm with an isotropic Cauchy mutation (Alg. ICEP) and is similar to the algorithm with a Gaussian mutation (Alg. GEP). These results can be explained because, in general, the individuals employed values of q close to 1 (Figs. 4(a) and (b)), i.e., the q -Gaussian mutation was similar to Gaussian mutation in these problems. It can be observed that the norm of the mutation strength parameter vector increases for all algorithms in the initial stage of the evolution, allowing a faster convergence in the initial steps. Similar results can be observed in the minimization of function f_d , which has few local optima (Figs. 3 and 4(d)).

Second, while the algorithm with Gaussian mutation (Alg. GEP) outperforms the algorithm with isotropic Cauchy mutation (Alg. ICEP) in the unimodal function f_a and f_b , it overperforms Alg. ICEP in the highly multimodal problem f_c (Fig. 3(c)). These results indicate that the long jumps generated by the Cauchy mutation were beneficial for the individuals to escape from local optima, mainly in the later stage of the evolution where the mutation strength parameters converged to small values (Fig. 4(c)).

The proposed algorithm (Alg. qGEP) outperforms Alg. GEP too, and is similar to Alg. ICEP. This result can be explained because higher values of the q parameter were employed on some occasions to allow longer jumps (see Fig. 5, where the distribution parameter q of the current best individual on the first trial of the experiment on function f_c is plotted against generation). It can be observed that, in Fig. 4(c), the mean q parameter increases to a value close to 2 in the initial steps and, then, decreases to smaller values. This does not occur in the minimization of function f_e and f_f . One can remember that the size of the search space in such problems is smaller (see the range of the solutions in Table I) than in problem f_c . Hence, long jumps may not be beneficial.

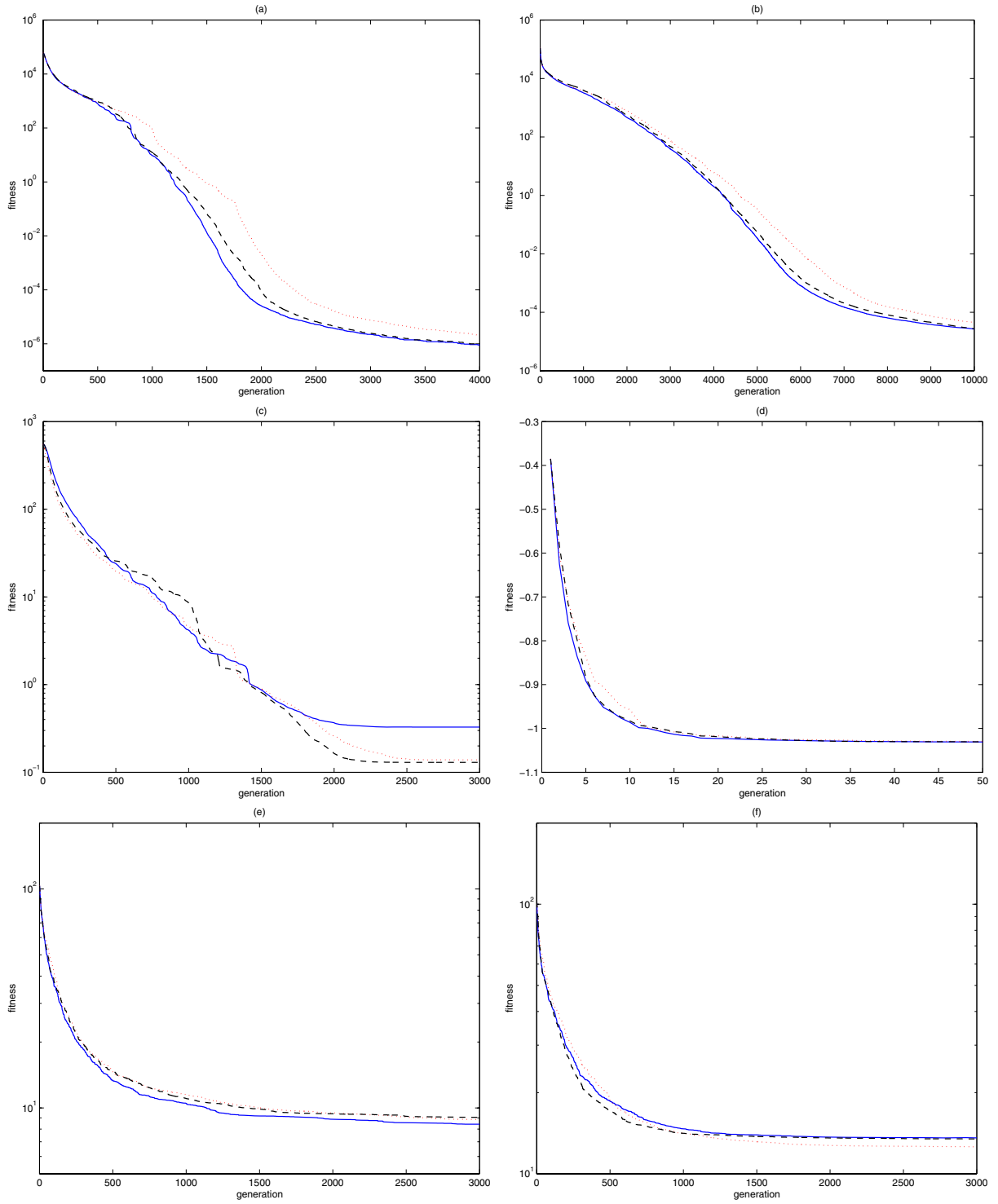


Fig. 3. Mean best-of-generation fitness on functions $f_a - f_f$. Alg. GEP: solid line; Alg. ICEP: dotted line; Alg. qGEP: dashed line.

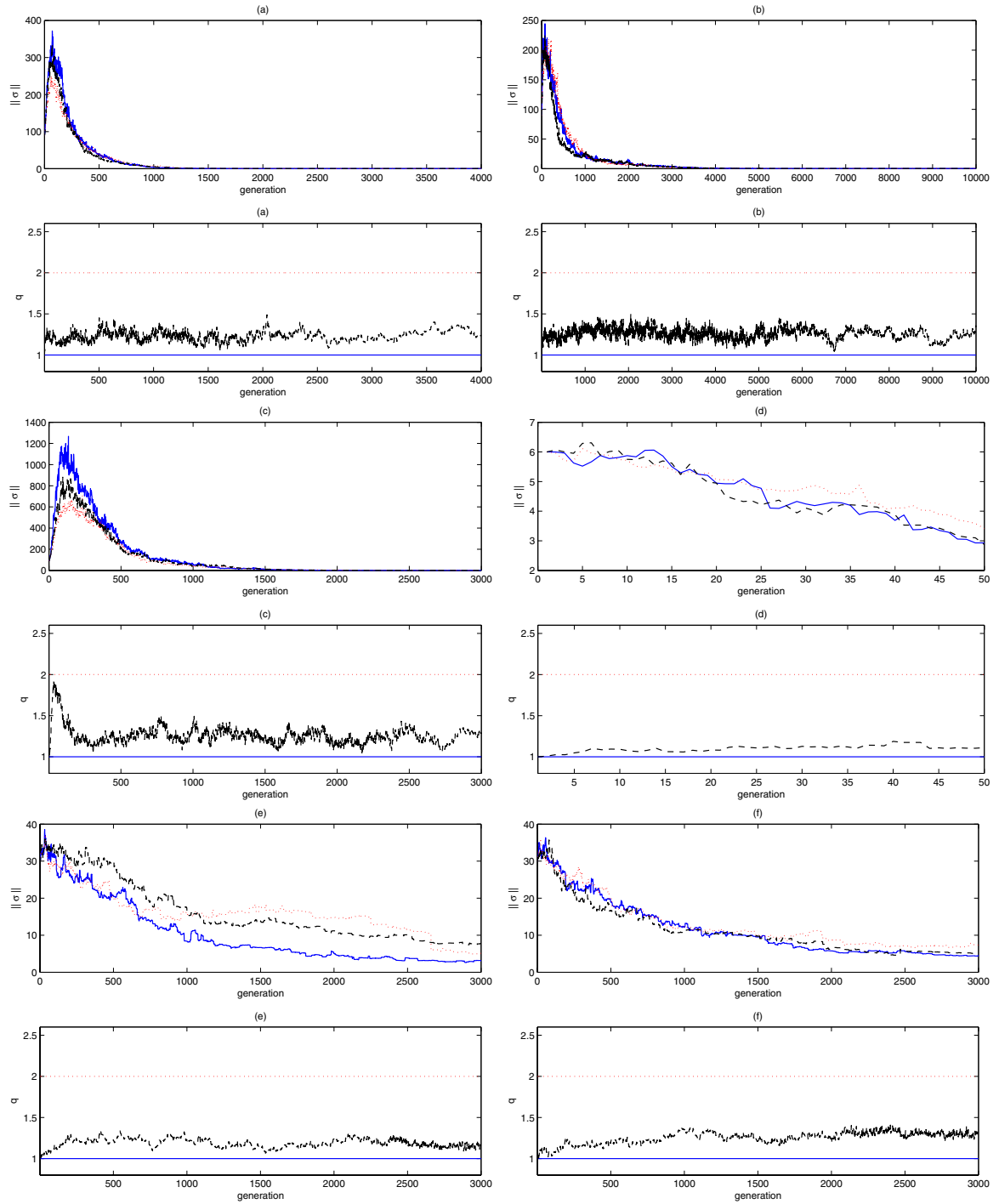


Fig. 4. Mean norm of the mutation strength parameter vector and mean distribution parameter q of the current best individual on functions $f_a - f_f$. Alg. GEP: solid line; Alg. ICEP: dotted line; Alg. qGEP: dashed line.

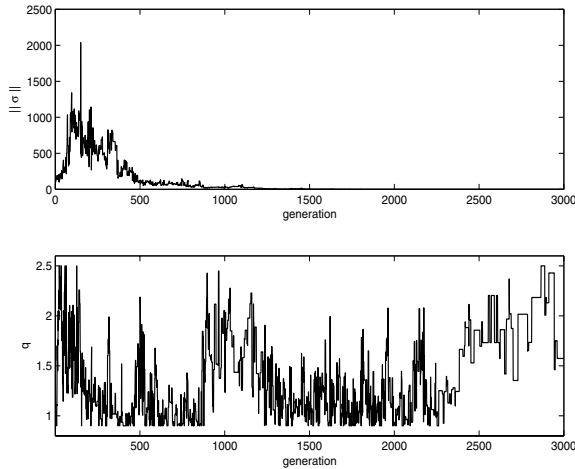


Fig. 5. Norm of the mutation strength parameter vector and distribution parameter q of the current best individual of Alg. qGEP on the first trial of the experiment on function f_c .

VI. CONCLUSIONS AND FUTURE WORK

In this paper, the use of self-adaptation is proposed, not only to control the mutation strength parameter, but also to control the mutation distribution. For this purpose, the isotropic q -Gaussian distribution is employed in the mutation operator. The q -Gaussian distribution allows to control the shape of the distribution by setting a real parameter q and can reproduce either finite or infinite second moment distributions. In the proposed algorithm, 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 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 minimized by letting the proposed algorithm to decide which mutation distribution should be used. This property can be observed in the experiments presented in Section V-C, where the proposed q -Gaussian EP algorithm presents a performance similar to the Gaussian EP algorithm when the Gaussian EP was better than the EP with isotropic Cauchy mutation and similar to the EP with isotropic Cauchy mutation when it was better than the Gaussian EP.

Much work can be further done in this area. New algorithms based on the ideas proposed here and other control methods for the q parameter should be investigated, including self-organization [11]. As a future work, the proposed algorithm will be investigated in continuous dynamic optimization problems.

ACKNOWLEDGMENTS

This work was supported by Brazil FAPESP (Proc. 04/04289-6) and UK EPSRC (No. EP/E060722/01).

REFERENCES

- [1] H.-G. Beyer and H. S. Schwefel, "Evolution strategies: a comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.
- [2] A. E. Eiben, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithms," *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 2, pp. 124–141, 1999.
- [3] N. Hansen and A. Ostermeier, "Completely derandomized self-adaptation in evolution strategies," *Evolutionary Computation*, vol. 9, no. 2, pp. 159–195, 2001.
- [4] N. Hansen, F. Gemperle, A. Auger, and P. Koumoutsakos, "When do heavy-tail distributions help?" *Proc. of the 9th Int. Conf. on Parallel Problem Solving from Nature (PPSN IX)*, *Lecture Notes in Computer Science*, vol. 4193, pp. 62–71, 2006.
- [5] M. Iwamatsu, "Generalized evolutionary programming with levy-type mutation," *Computer Physics Communications*, vol. 147, no. 1-2, pp. 729–732, 2002.
- [6] C.-Y. Lee and X. Yao, "Evolutionary programming using mutations based on the levy probability distribution," *IEEE Trans. on Evolutionary Computation*, vol. 8, no. 1, pp. 1–13, 2004.
- [7] 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*, vol. 363, no. 2, pp. 260–268, 2006.
- [8] A. Obuchowicz, "Multidimensional mutations in evolutionary algorithms based on real-valued representation," *Int. Journal of Systems Science*, vol. 34, no. 7, pp. 469–483, 2003.
- [9] 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*, vol. 236, no. 1-2, pp. 52–57, 1997.
- [10] W. Thistleton, J. A. Marsh, K. Nelson, and C. Tsallis, "Generalized Box-Muller method for generating q-Gaussian random deviates," *ArXiv Condensed Matter e-prints*, May 2006.
- [11] R. Tinós and S. Yang, "Self-organizing random immigrants genetic algorithm for dynamic optimization problems," *Genetic Programming and Evolvable Machines*, vol. 8, no. 3, 2007.
- [12] C. Tsallis and D. A. Stariolo, "Generalized simulated annealing," *Physica A*, vol. 233, no. 1-2, pp. 395–406, 1996.
- [13] C. Tsallis, "Possible generalization of boltzmann-gibbs statistics," *Journal of Statistical Physics*, vol. 52, pp. 479–487, 1988.
- [14] X. Yao and Y. Liu, "Fast evolution strategies," *Control and Cybernetics*, vol. 26, no. 3, pp. 467–496, 1997.
- [15] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 2, pp. 82–102, 1999.