Evolutionary Programming with q-Gaussian Mutation for Dynamic Optimization Problems

Renato Tinós and Shengxiang Yang

***Abstract*— The use of evolutionary programming algorithms with self-adaptation of the mutation distribution for dynamic optimization problems is investigated in this paper. In the proposed method, the q-Gaussian distribution is employed to generate new candidate solutions by mutation. A real parameter q, which defines the shape of the distribution, is encoded in the chromosome of individuals and is allowed to evolve. Algorithms with self-adapted mutation generated from isotropic and anisotropic distributions are presented. In the experimental study, the q-Gaussian mutation is compared to Gaussian and Cauchy mutation on three dynamic optimization problems.**

I. Introduction

One of the main differences between Evolutionary Algorithms (EAs) and traditional deterministic optimization techniques is that, in EAs, stochastic operators are used during the search process. Traditionally, in Evolutionary Programming (EP) and Evolution Strategies (ES) applied to real-valued optimization, new candidate solutions are generated by mutation using multivariate samples taken from isotropic Gaussian distributions [1]. The use of an isotropic Gaussian distribution in EAs 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 [3]. The isotropic Gaussian distribution has a finite second moment and does not favor any direction in the search space.

In recent years, researchers have proposed the use of mutation distributions with longer tails and infinite second moment in EAs. For example, in the Fast Evolutionary Programming (FEP) [10], the Cauchy distribution is employed. The use of mutation taken from heavy tail distributions implies jumps of scale-free sizes, eventually allowing to reach distant regions of the search space faster. This property is interesting when EAs are applied to some multimodal problems or dynamic optimization problems (DOPs) as it can allow the population to escape faster from local optima. However, in some problems, as less local candidate solutions are generated, the convergence to the (local) optima can be slower. In this way, the choice of the mutation distribution can be very important for the EA, particularly in DOPs, where the use of one or other mutation distribution can result in a very different performance.

It is important to observe that the use of EP and ES in real-valued DOPs is interesting because self-adaptation provides an intrinsic mechanism for adaptation to the changes of the problem. In [9], the behavior of ES with self-adaptation of the mutation strength parameter, used to control the spread of a Gaussian mutation distribution, was investigated in DOPs.

In this paper, the use of EP with self-adaptation of the mutation distribution along the evolutionary process proposed in [6] is investigated for DOPs. In this strategy, named EP with q-Gaussian mutation, self-adaptation is employed, not only to control the mutation strength parameter, but also to control the mutation distribution. A real parameter that defines 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 distribution [3] 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 main contribution of this paper is the investigation of the use of the EP with q-Gaussian mutation in DOPs. Additionally, a comparison of the EP with Gaussian and Cauchy mutations in three dynamic environments is provided.

The rest of this paper is organized as follows. The EP algorithm with q-Gaussian mutation taken from anisotropic and isotropic distributions are presented in Section II. The experimental study with DOPs generated from three stationary test functions is presented in Section III. Finally, this article is concluded in Section IV.

II. EP with Self-Adaptation of the Mutation Distribution

In [6], the use of q-Gaussian mutation was proposed in EP. In an m-dimensional real-valued search space, the new candidate solution is generated by the mutation operator from individual $\mathbf{x}_i$ as follows:

$$\tilde{\mathbf{x}}_i = \mathbf{x}_i + \mathbf{C}\ \mathbf{z}, \tag{1}$$

where $i = 1, \ldots, \mu$, $\mathbf{z}$ is an m-dimensional random vector generated from a q-Gaussian distribution with zero mean, and $\mathbf{C}$ is a diagonal matrix with the main diagonal composed by the elements of vector $\sigma = [\sigma(1)\ \sigma(2) \ldots \sigma(m)]^{\mathrm{T}}$, which defines the mutation strength in each coordinate.

The q-Gaussian distribution has interesting properties. Differently from the Gaussian distribution, which is an attractor

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).

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

978-1-4244-1823-7/08/$25.00©2008 IEEE

only for independent systems with a finite second moment, the q-Gaussian can be used to represent correlated systems with an infinite second moment too [3]. 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 [4]. In this paper, the generalized Box-Müller method proposed in [3] is employed to generate q-Gaussian random variables for $-\infty < q < 3$.

Based on the mutation strength self-adaptation [1], the authors in [6] have proposed to add the parameter q_i of each individual i to its chromosome and to multiplicatively update it as follows:

$$\tilde{q}_i = q_i \exp\big(\tau_a \mathcal{N}(0,1)\big), \tag{2}$$

where τ_a denotes the standard deviation of the Gaussian distribution and $\mathcal{N}(0,1)$ denotes a sample variable taken from the Gaussian distribution with zero mean and standard deviation one. In this way, different distributions can be reproduced during the evolutionary process.

Two procedures can be used to generate the m-dimensional random vector $\mathbf{z}$ in Eq. 1. In the first, the q-Gaussian random vector $\mathbf{z}$ is generated by sampling m independent q-Gaussian variables $\mathcal{N}_q(0,1)$. It is important to observe that when q is high, this procedure implies that some directions in the search space are more explored than others (i.e., the distribution that generates the random vector $\mathbf{z}$ is anisotropic). The use of random variables generated by sampling independent random variables taken from a heavy tail distribution can be interesting for optimization problems with separable evaluation functions, as most of the large steps occurs close to the coordinate axes [5] 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 non-separable evaluation functions. The EP algorithm with mutation generated from anisotropic q-Gaussian distribution is called $qGEP$ in this paper.

In the second approach, based on the works [5] and [3], the random mutation vector $\mathbf{z}$ is generated from an isotropic q-Gaussian distribution as follows:

$$\mathbf{z} \sim r\mathbf{u}, \tag{3}$$

where $r \sim \mathcal{N}_q(0,1)$, i.e., a random variable with q-Gaussian distribution, and $\mathbf{u}$ is an uniform random vector obtained by sampling a random vector with Gaussian distribution and normalizing it to length one, i.e., $\mathbf{u} = \mathbf{v}/\|\mathbf{v}\|$ where $\mathbf{v}$ is a random vector with Gaussian distribution and $\|\mathbf{v}\|$ denotes the Euclidean norm of the vector $\mathbf{v}$. The EP algorithm with mutation generated from isotropic q-Gaussian distribution is called $IqGEP$ in this paper.

The algorithms qGEP and IqGEP are presented in Alg. 1. The main difference of the EP algorithm presented in Alg. 1 from Gaussian EP and FEP [2] is that, in the proposed algorithm, the q-Gaussian mutation is employed (step 6) instead of the Gaussian (Gaussian EP), or Cauchy (FEP) mutation, and a procedure to adapt the q parameter is adopted (step 5).

Algorithm 1 EP with q-Gaussian mutation.

1: Initialize the population composed of individuals ($\mathbf{x}_i$, σ_i, q_i) for $i = 1, \ldots, \mu$
2: **while** (stop criteria are not satisfied) **do**
3:   **for** $i \leftarrow 1$ to μ **do**
4:     update the mutation strength parameter $\tilde{\sigma}_i(j)$ for $j = 1$ to m
5:     update the parameter $\tilde{q}_i$ (Eq. 2)
6:     $\tilde{\mathbf{x}}_i \leftarrow \mathbf{x}_i + \mathbf{C}_i\ \mathbf{z}$, where $\mathbf{z}$ is a random vector generated from an anisotropic or isotropic q-Gaussian distribution with parameter $\tilde{q}_i$, and $\mathbf{C}_i = \mathrm{diag}(\tilde{\sigma}_i^{\mathrm{T}})$
7:   **end for**
8:   Compute the fitness of the parents ($\mathbf{x}_i$, σ_i, q_i) and offspring ($\tilde{\mathbf{x}}_i$, $\tilde{\sigma}_i$, $\tilde{q}_i$) for $i = 1, \ldots, \mu$
9:   Compute the winning function [2] 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**

III. EXPERIMENTAL STUDY

In the experiments presented in this work, the dynamic problem generator with control of rotation of the individuals [7] is employed to construct DOPs based on three stationary test problems. The dynamic problem generator is presented in Section III-A while the three stationary test problems used here are described in Section III-B. The experimental design is described in Section III-C, and the results are presented and analyzed in Section III-D.

Experiments with algorithms qGEP and IqGEP are presented in this paper (see last section). In all experiments, each EP algorithm with q-Gaussian mutation distribution is compared to two other approaches where the parameter q is fixed. The approaches are defined as follows:

- Algorithms qGEP (anisotropic) and IqGEP (isotropic): use of one changing q for each individual, i.e., Eq. 2 is employed.
- Algorithms GEP and IGEP: use of only one fixed parameter $q = 1.0$ for all individuals, i.e., step 5 in Alg. 1 is changed by $\tilde{q}_i = 1.0$ and the Gausssian mutation is reproduced.
- Algorithms CEP and ICEP: use of only one fixed parameter $q = 2.0$ for all individuals, i.e., step 5 in Alg. 1 is changed by $\tilde{q}_i = 2.0$ and the Cauchy mutation is reproduced.

TABLE I

TEST FUNCTIONS.

Function $f(\mathbf{x})$	Range	m
$f_1 = \sum_{j=1}^{m} x_j^2$	$\mathbf{x} \in [-100, 100]^m$	30
$f_2 = \sum_{j=1}^{m} \left(x_j^2 - 10\cos(2\pi x_j) + 10 \right)$	$\mathbf{x} \in [-5, 5]^m$	10
$f_3 = \sum_{j=1}^{m} \left(y_j^2 - 10\cos(2\pi y_j) + 10 \right)$, where $\mathbf{y} = \mathbf{M}(\mathbf{x})$	$\mathbf{x} \in [-5, 5]^m$	10

In the algorithms qGEP, GEP, and CEP, the mutations are generated by sampling m independent random variables, while in the algorithms IqGEP, IGEP, and ICEP, Eq. 3, where r is respectively taken from q-Gaussian, Gaussian, and Cauchy distributions, is employed to generate the new candidate solutions. In this way, the EP with q-Gaussian mutation is compared to the EP with Gaussian and Cauchy mutation in the isotropic and anisotropic versions.

A. Dynamic Problem Generator

In order to evaluate the performance of different EAs in DOPs, a dynamic problem generator that can generate DOPs from any continuous encoded stationary problem was proposed in [7]. Given a stationary problem where the fitness function is $f(\mathbf{x}(t))$ and $\mathbf{x}(t) \in \mathbb{R}^m$, the DOP is obtained by periodically moving all the individuals of the population every τ generations. The new location of each individual after the change is obtained by unnormalizing the vector:

$$\mathbf{z^n}(t) = \mathbf{A}(k)\mathbf{x^n}(t), \tag{4}$$

where $\mathbf{z^n}(t) \in [-1,1]^m$, $\mathbf{x^n}(t) \in [-1,1]^m$ is the normalized vector $\mathbf{x}(t)$, which represents the position of the individual before the change, and the linear transformation $\mathbf{A}(k)$ in the k dynamic environment is a rotation matrix composed by successive simple rotations in random planes. Defining $\rho = \theta/180$, where θ is the rotation angle given in degrees, the parameter ρ can be employed to control the degree of change for the DOP. If $\rho = 0.0$, the problem stays stationary, while if $\rho = 1.0$, the extreme changes occur. In this way, the speed and degree of the environmental change can be controlled by changing, repectively, the values of τ and ρ.

B. Stationary Test Problems

Three stationary problems are selected as test suite for the algorithms. The DOPs are constructed from these stationary problems using the dynamic problem generator described in Section III-A. The test functions, which should be minimized during the optimization process, are presented in Table I. In function f_3, the matrix $\mathbf{M}$ is obtained by the orthogonalization of a random matrix uniformly distributed in the unit hypersphere.

While the function f_1 is unimodal, functions f_2 and f_3 are multimodal. Functions f_2 and f_3 are the axis parallel and rotated Rastrigin functions respectively. The axis parallel Rastrigin function is separable, while the rotated Rastrigin function is not separable.

C. Experimental Design

In order to compare the algorithms, each one was executed 30 times (with 30 different random seeds) for each test function presented in Table I for different values of τ and ρ. The first two values of τ (100 and 800) imply changing the fitness function in an early and a medium stage of the optimization process respectively. The last value, $\tau = 2000$, implies changing the fitness function in a late stage of the optimization process. Environments with three different values of ρ were generated in this paper. These values represent different change levels: light shifting ($\rho = 0.05$), medium variation ($\rho = 0.3$), and severe change ($\rho = 0.7$).

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 for the EP algorithms with mutation generated from anisotropic distributions and, in order to have comparable results, the initial value of $\sigma_i(j)$ is equal to $3.0\sqrt{m}$ for the EP algorithms with mutation generated from isotropic distributions. The initial q-Gaussian parameter q in qGEP and IqGEP is equal to 1.0, i.e., the initial q-Gaussian distribution reproduces the Gaussian distribution. In qGEP and IqGEP, the minimum and maximum values of the q-Gaussian parameter q are 0.9 and 2.5 respectively, and $\tau_a = \frac{5}{\sqrt{m}}$ in Eq. 2. In all algorithms, the minimum allowed value for the mutation strength parameter $\sigma_i(j)$ is 0.01.

D. Experimental Results

The experimental results of the mean fitness of the best individual in the last generation before the changes averaged over 30 runs are presented in tables II and III, respectively, for the EPs with mutation generated from anisotropic and isotropic distributions. The statistic comparisons between the algorithms, which are carried out by t-test with 58 degrees of freedom at a 0.1 level of significance, are also presented. Figs. 1, 2, and 3 show the averaged results of fitness, Euclidean norm of the mutation strength parameter vector, and the mean distribution parameter q of the current best individual for algorithms GEP, CEP, and qGEP in the experiments with $\tau = 2000$. From these tables and figures, several results can be observed and are analyzed as follows.

One can observe that, in the three approaches (with Gaussian, Cauchy, and q-Gaussian mutations), the mutation strength parameters generally increase when the environmental changes occur. Increasing the mutation strength parameters implies that mutation distributions have a larger second order moment (one can remember that the mutation strength

TABLE II

RESULTS OF THE MEAN BEST-OF-GENERATION FITNESS BEFORE THE CHANGES AND RELEVANT STATISTICAL COMPARISONS (INSIDE THE PARENTHESES) FOR THE ALGORITHMS WITH MUTATION GENERATED FROM ANISOTROPIC DISTRIBUTIONS. 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	Dynamics		Algorithm		
	τ	ρ	GEP	CEP	qGEP
f_1	100	0.05	1.9811E+003 (-)	2.1114E+003 (-)	2.6766E+003
	100	0.30	6.5480E+003 (∼)	4.9329E+003 (-)	6.6921E+003
	100	0.70	1.0938E+004 (-)	1.0719E+004 (-)	1.3180E+004
	800	0.05	0.8334E-001 (-)	1.1454E-001(-)	6.2527E-001
	800	0.30	2.6604E+001 (+)	0.0177E+001 (-)	0.4696E+001
	800	0.70	3.9915E+002 (+)	0.0040E+002 (-)	0.4857E+002
	2000	0.05	0.2583E-002 (-)	2.6469E-002 (+)	0.2740E-002
	2000	0.30	0.2871E-002 (-)	2.8524E-002 (+)	0.2971E-002
	2000	0.70	0.3137E-002 (+)	2.9835E-002 (+)	0.3058E-002
f_2	100	0.05	1.8593E+001 (-)	2.7924E+001 (+)	2.1181E+001
	100	0.30	5.6235E+001 (+)	3.2245E+001 (∼)	3.3516E+001
	100	0.70	6.3172E+001 (+)	3.5776E+001 (+)	3.2575E+001
	800	0.05	1.2023E+001 (+)	0.2622E+001 (-)	0.3655E+001
	800	0.30	5.1576E+001 (+)	0.2633E+001 (-)	0.4732E+001
	800	0.70	7.4204E+001 (+)	0.2642E+001 (-)	0.5017E+001
	2000	0.05	1.1964E+001 (+)	0.2540E+001 (∼)	0.2523E+001
	2000	0.30	4.3656E+001 (+)	0.2544E+001 (-)	0.2670E+001
	2000	0.70	4.6781E+001 (+)	0.2543E+001 (∼)	0.2594E+001
f_3	100	0.05	2.6638E+001 (-)	4.0565E+001 (+)	2.9780E+001
	100	0.30	4.6086E+001 (+)	4.5033E+001 (+)	4.1717E+001
	100	0.70	4.5310E+001 (+)	4.7300E+001 (+)	4.1751E+001
	800	0.05	1.5009E+001 (∼)	1.5116E+001 (∼)	1.4198E+001
	800	0.30	9.3379E+001 (+)	4.7740E+001 (+)	3.4237E+001
	800	0.70	1.4862E+002 (+)	0.5633E+002 (+)	0.3036E+002
	2000	0.05	1.4791E+001 (∼)	1.3793E+001 (∼)	1.4219E+001
	2000	0.30	9.8225E+001 (+)	4.8846E+001 (+)	3.4941E+001
	2000	0.70	1.4498E+002 (+)	0.6107E+002 (+)	0.2958E+002

TABLE III

RESULTS OF THE MEAN BEST-OF-GENERATION FITNESS BEFORE THE CHANGES AND RELEVANT STATISTICAL COMPARISONS FOR THE ALGORITHMS WITH MUTATION GENERATED FROM ISOTROPIC DISTRIBUTIONS. THE RESULT REGARDING ALG. X IS SHOWN AS "+", "−", OR "∼" WHEN ALG. IQGEP IS, RESPECTIVELY, SIGNIFICANTLY BETTER THAN, SIGNIFICANTLY WORSE THAN, OR STATISTICALLY EQUIVALENT TO ALG. X.

Problem	Dynamics		Algorithm		
	τ	ρ	IGEP	ICEP	IqGEP
f_1	100	0.05	2.8987E+003 (∼)	2.8929E+003 (∼)	2.6082E+003
	100	0.30	6.9995E+003 (∼)	8.4478E+003 (+)	7.2625E+003
	100	0.70	1.2715E+004 (∼)	1.4516E+004 (+)	1.2923E+004
	800	0.05	1.3799E+001 (∼)	3.5996E+001 (∼)	2.3717E+001
	800	0.30	1.0431E+003 (∼)	0.7352E+003 (-)	0.9822E+003
	800	0.70	4.5721E+003 (∼)	4.1629E+003 (∼)	4.6143E+003
	2000	0.05	0.0221E-003 (∼)	1.1338E-003 (∼)	246.97E-003
	2000	0.30	0.3424E-001 (∼)	0.2052E-001 (∼)	2.5674E-001
	2000	0.70	7.3356E-001 (∼)	9.3955E-001 (∼)	7.9539E-001
f_2	100	0.05	1.9651E+001 (-)	2.2362E+001 (∼)	2.1975E+001
	100	0.30	4.0055E+001 (∼)	3.5213E+001 (-)	3.8951E+001
	100	0.70	4.6373E+001 (+)	3.9093E+001 (∼)	4.1714E+001
	800	0.05	1.1101E+001 (∼)	1.1710E+001 (∼)	1.2104E+001
	800	0.30	5.4840E+001 (+)	4.1870E+001 (∼)	3.7298E+001
	800	0.70	8.9868E+001 (+)	5.2495E+001 (+)	4.1956E+001
	2000	0.05	1.1678E+001 (∼)	1.1113E+001 (∼)	1.2203E+001
	2000	0.30	6.7978E+001 (+)	3.7464E+001 (∼)	3.3995E+001
	2000	0.70	1.0020E+002 (+)	0.5044E+002 (+)	0.4107E+002
f_3	100	0.05	2.3642E+001 (∼)	2.6707E+001 (+)	2.4608E+001
	100	0.30	4.1902E+001 (+)	3.8126E+001 (∼)	3.6294E+001
	100	0.70	4.1246E+001 (+)	3.8540E+001 (∼)	3.6739E+001
	800	0.05	1.5027E+001 (∼)	1.3414E+001 (∼)	1.3806E+001
	800	0.30	6.3718E+001 (+)	4.8625E+001 (+)	4.3253E+001
	800	0.70	1.0150E+002 (+)	0.4887E+002 (+)	0.3301E+002
	2000	0.05	1.3454E+001 (∼)	1.4550E+001 (∼)	1.4752E+001
	2000	0.30	7.9785E+001 (+)	4.8606E+001 (+)	3.8141E+001
	2000	0.70	1.2311E+002 (+)	0.4622E+002 (+)	0.3461E+002

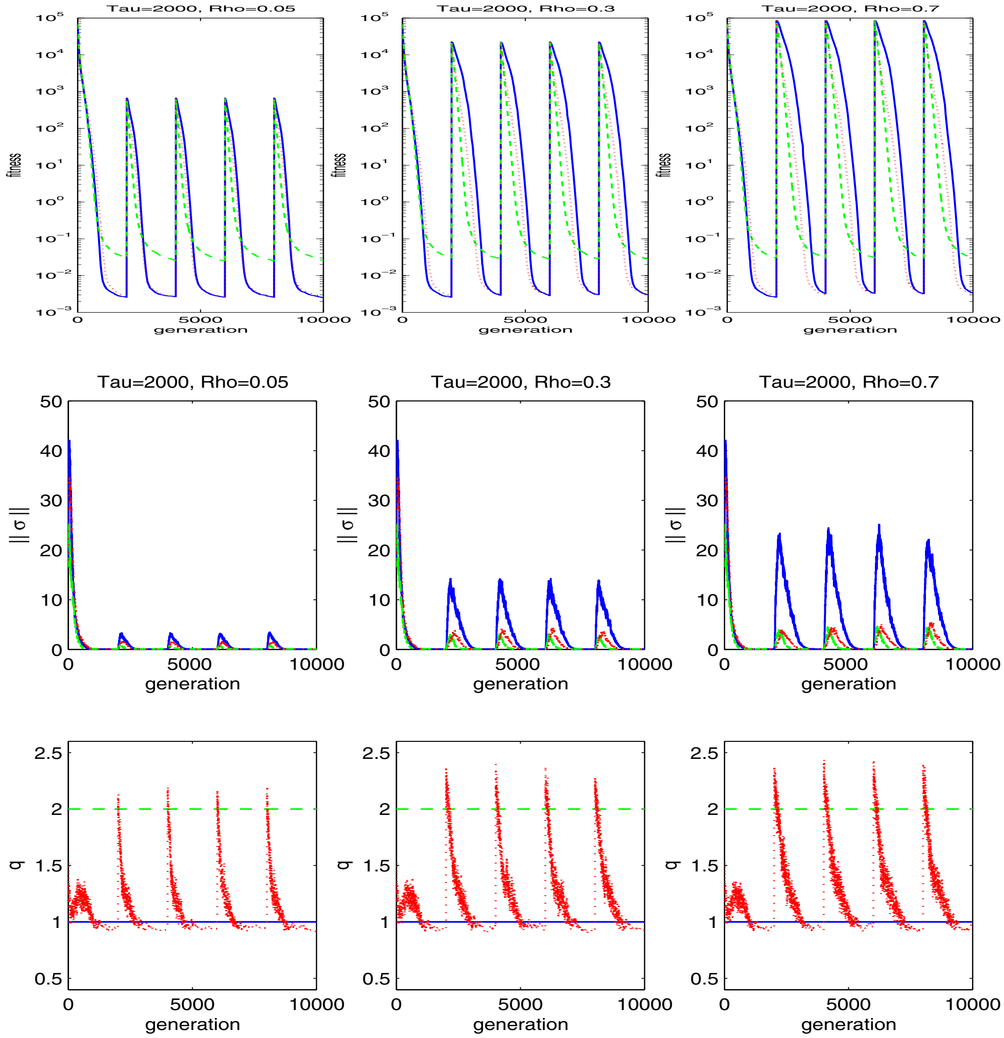

Fig. 1. Mean fitness, mean Euclidean norm of the mutation strength parameter vector, and mean distribution parameter q of the current best individual on DOP f_1 with $\tau = 2000$. Alg. GEP: solid line; Alg. CEP: dotted line; Alg. qGEP: dashed line.

parameters define the standard deviation in the Gaussian mutation distribution). When the mutation strength parameters increase after an environmental change, larger jumps occur more often, which can help the population to reach faster new optima. Additionally, in the EP with q-Gaussian mutation, the parameter q increases after an environmental change, which results in distributions with heavier tails. In this way, self-adaptation in EP (or ES) is very interesting to DOPs as it provides an intrinsic mechanism to adaptation to the changes of the problem.

Let us now analyze the results of the algorithms with Gaussian and Cauchy mutations. For the unimodal function f_1, while the algorithms with Cauchy mutation (Alg. CEP and ICEP) generally present a faster converge, the algorithms with Gaussian mutation (Alg. GEP and IGEP) provide better results regarding the best-of-generation fitness before the changes, mainly for $\tau = 2000$. These results can be explained because, in general, larger steps occur more often when the Cauchy mutation is used, which allows a faster convergence in the initial steps, but a slow convergence to the optimum in a late stage of the optimization process as a smaller number of candidate solutions (when compared to the algorithms with Gaussian mutation) are generated near the current best solution. However, for the multimodal functions, larger jumps produced by the Cauchy mutation generally allow the population to escape from local optima (see Figs. 2 and 3), mainly in the later stages of the evolution where the mutation strength parameters have converged to small values.

In the experiments with the q-Gaussian mutation, distributions with heavier tails (higher values of q) are used by the algorithm just after the changes and compact distributions are used in the later stages after the changes. One can observe

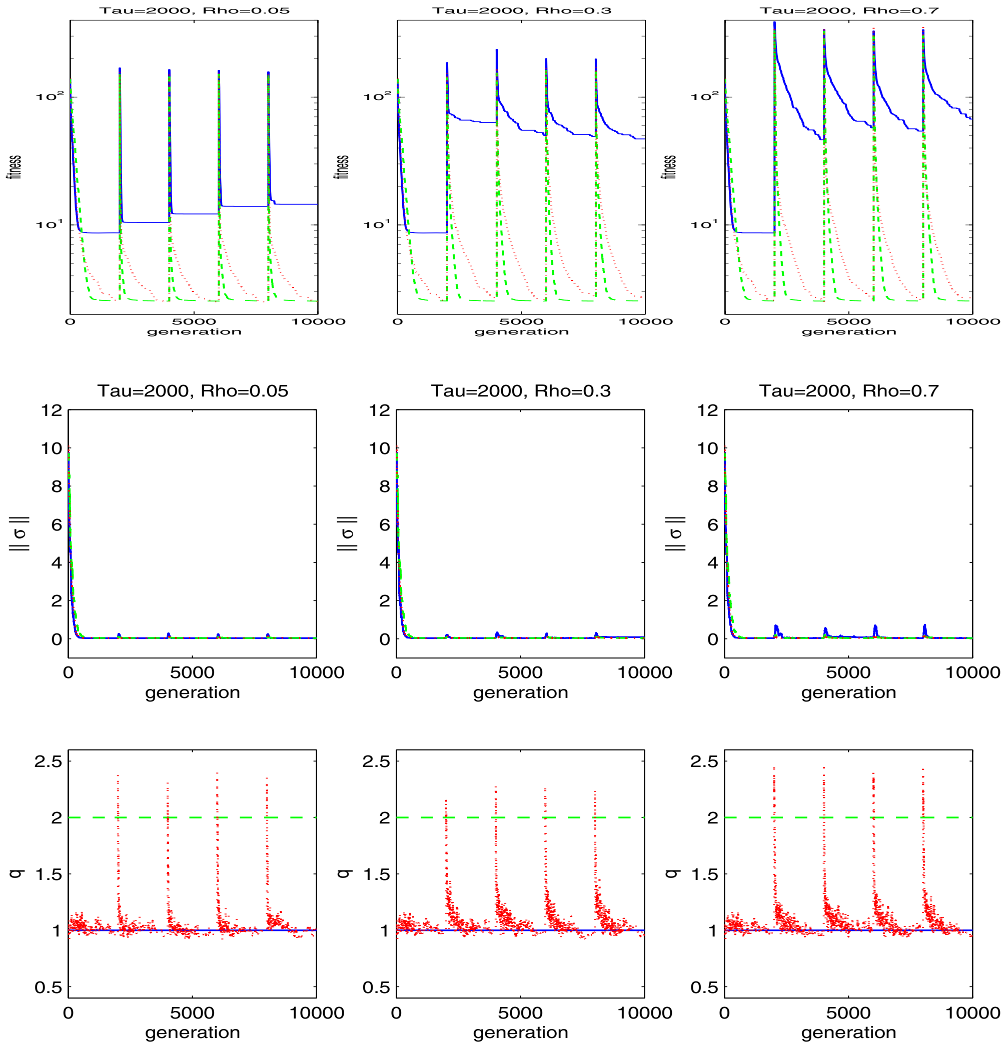

Fig. 2. Mean fitness, mean Euclidean norm of the mutation strength parameter vector, and mean distribution parameter q of the current best individual on DOP f_2 with $\tau = 2000$. Alg. GEP: solid line; Alg. CEP: dotted line; Alg. qGEP: dashed line.

that the mean value of q in Figs. 1, 2 and 3 reaches a value higher than 2 (Cauchy distribution) and decreases to a value close to 1 (Gaussian distribution). In this way, larger steps occur more often in the generations just after the changes, which helps the population to escape from local optima or to converge faster. In the later generations, the distribution with small values of q increases the local search, which helps the algorithm to reach the best local optima.

One can observe that a better performance of the EP with mutation generated from anisotropic Cauchy mutation (CEP) is generally reached on DOP f_2. This fact occurs because function f_2 is separable and more larger steps, which are generally parallel to the coordinate axes, were generated by the algorithm CEP. However, in the experiments with function f_3, the algorithm qGEP generally reaches a better performance, as the function is not separable and the Cauchy mutation generates a higher number of long jumps parallel to the coordinate axes.

Table IV presents the statistical comparisons between algorithms with mutation generated from isotropic and anisotropic distributions. It is observable that, while the performance of the algorithm with mutation generated from anisotropic Cauchy distribution (CEP) generally outperforms the algorithm with mutation generated from isotropic distribution (ICEP) in problem f_2 (separable function), it generally overperforms ICEP in problem f_3 (not separable function). Similar results are generally reached by the algorithms with q-Gaussian mutation. Despite a worse performance for larger values of τ, it is possible to observe that the fitness is continuously improved when the mutation generated from

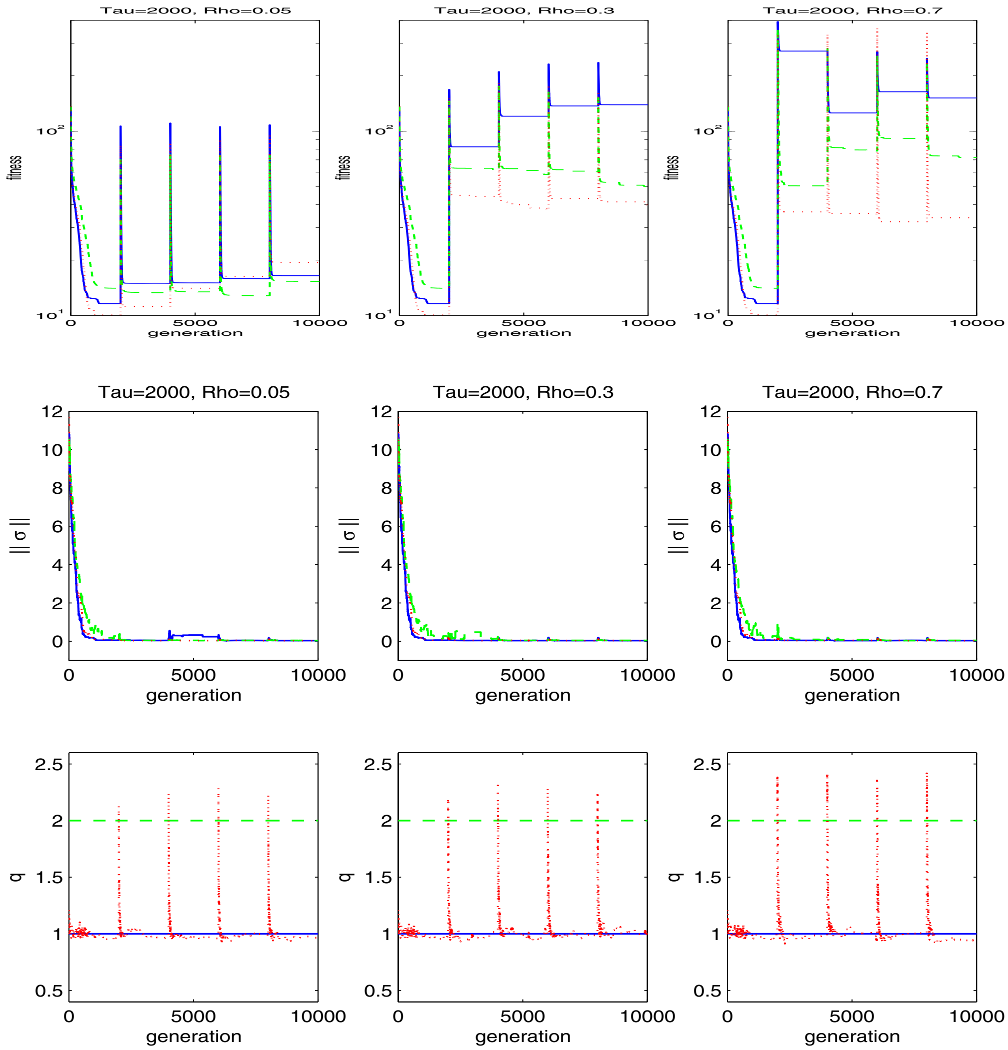

Fig. 3. Mean fitness, mean Euclidean norm of the mutation strength parameter vector, and mean distribution parameter q of the current best individual on DOP f_3 with $\tau = 2000$. Alg. GEP: solid line; Alg. CEP: dotted line; Alg. qGEP: dashed line.

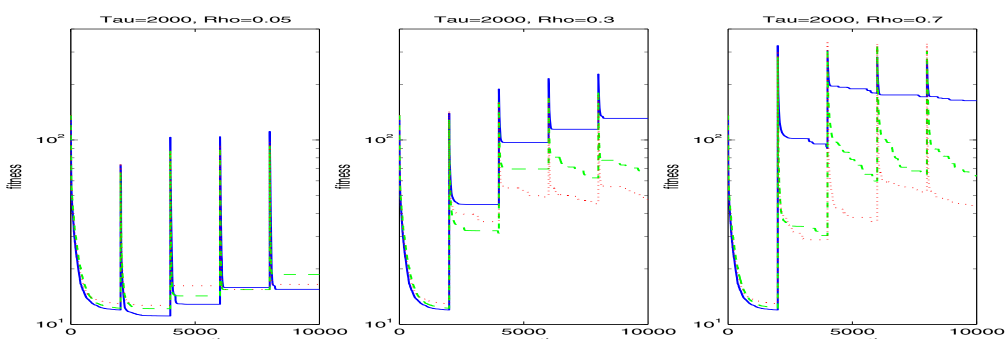

Fig. 4. Mean fitness of the current best individual on DOP f_3 with $\tau = 2000$. Alg. IGEP: solid line; Alg. ICEP: dotted line; Alg. IqGEP: dashed line.

TABLE IV

STATISTICAL COMPARISON OF THE BEST-OF-GENERATION FITNESS IN THE LAST GENERATION BEFORE THE CHANGES FOR THE ALGORITHMS WITH MUTATION GENERATED FROM ANISOTROPIC OR ISOTROPIC DISTRIBUTIONS. THE RESULT REGARDING ALG. X IS SHOWN AS "+", "−", OR "∼" WHEN ISOTROPIC ALG. X IS, RESPECTIVELY, SIGNIFICANTLY BETTER THAN, SIGNIFICANTLY WORSE THAN, OR STATISTICALLY EQUIVALENT TO THE ANISOTROPIC ALG. X.

Problem	Dynamics		Algorithm		
	τ	ρ	Alg. IGEP/GEP	Alg. ICEP/CEP	Alg. IqGEP/qGEP
f_1	100	0.05	-	-	∼
	100	0.3	∼	-	∼
	100	0.7	-	-	∼
	800	0.05	-	-	-
	800	0.3	-	-	-
	800	0.7	-	-	-
	2000	0.05	+	+	∼
	2000	0.3	-	+	∼
	2000	0.7	-	-	-
f_2	100	0.05	∼	+	∼
	100	0.3	+	-	-
	100	0.7	+	-	-
	800	0.05	∼	-	-
	800	0.3	∼	-	-
	800	0.7	-	-	-
	2000	0.05	∼	-	-
	2000	0.3	-	-	-
	2000	0.7	-	-	-
f_3	100	0.05	+	+	+
	100	0.3	∼	+	+
	100	0.7	+	+	+
	800	0.05	∼	∼	∼
	800	0.3	+	∼	-
	800	0.7	+	+	-
	2000	0.05	∼	∼	∼
	2000	0.3	+	∼	-
	2000	0.7	+	+	-

q-Gaussian isotropic distribution is used (Fig. 4).

IV. CONCLUSIONS AND FUTURE WORK

In this paper, the use of self-adaptation of the mutation distribution is proposed for DOPs. For this purpose, the isotropic and anisotropic q-Gaussian distributions are employed in the mutation operator. 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.

The experimental results indicate that this property can be useful for DOPs as, after the environmental changes, the increase of the parameter q caused by self-adaptation results in a higher number of long jumps (like the Cauchy mutation), which can help the population to escape from the local optima and/or converge faster to the optima. In the later stages after the environmental changes, the parameter q reaches small values, which improves the local search (like the Gaussian mutation). The experimental results also indicate that the use of mutation generated from anisotropic distributions is interesting in problems with separable fitness functions, but can cause performance degradation otherwise.

In the future, other control methods for the q parameter should be investigated, including self-organization [8]. Also, the proposed algorithm will be investigated on other continuous dynamic optimization problems.

REFERENCES

[1] H.-G. Beyer and H. S. Schwefel, "Evolution strategies: a comprehensive introduction," *Natural Computing*, vol. 1, pp. 3–52, 2002.

[2] C.-Y. Lee and X. Yao, "Evolutionary programming using mutations based on the levy probability distribution," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 1, pp. 1 – 13, 2004.

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

[4] 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.

[5] 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.

[6] R. Tinós and S. Yang, "Self-adaptation of mutation distribution in evolutionary algorithms," in *Proc. of the 2007 IEEE Congress on Evolutionary Computation*, 2007, pp. 79–86.

[7] ——, "Continuous dynamic problem generators for evolutionary algorithms," in *Proc. of the 2007 IEEE Congress on Evolutionary Computation*, 2007, pp. 236–243.

[8] ——, "Self-organizing random immigrants genetic algorithm for dynamic optimization problems," *Genetic Programming and Evolvable Machines*, vol. 8, no. 3, pp. 255–286, 2007.

[9] K. Weicker and N. Weicker, "On evolution strategy optimization in dynamic environments," in *Proc. of the 1999 Congress on Evolutionary Computation*, 2000, pp. 2039–2046.

[10] X. Yao, Y. Liu, and G. Lin, "Evolutionary programming made faster," *IEEE Trans. on Evolutionary Computation*, vol. 3, no. 2, pp. 82 – 102, 1999.