

PHENOTYPIC EVOLUTION WITH A MUTATION BASED ON SYMMETRIC α -STABLE DISTRIBUTIONS

ANDRZEJ OBUCHOWICZ*, PRZEMYSŁAW PRĘTKI*

* Institute of Control and Computation Engineering
University of Zielona Góra, ul. Podgórna 50, 65–246 Zielona Góra, Poland
e-mail: {A.Obuchowicz, P.Pretki}@issi.uz.zgora.pl

Multidimensional Symmetric α -Stable ($S\alpha S$) mutations are applied to phenotypic evolutionary algorithms. Such mutations are characterized by non-spherical symmetry for $\alpha < 2$ and the fact that the most probable distance of mutated points is not in a close neighborhood of the origin, but at a certain distance from it. It is the so-called surrounding effect (Obuchowicz, 2001b; 2003b). For $\alpha = 2$, the $S\alpha S$ mutation reduces to the Gaussian one, and in the case of $\alpha = 1$, the Cauchy mutation is obtained. The exploration and exploitation abilities of evolutionary algorithms, using $S\alpha S$ mutations for different α , are analyzed by a set of simulation experiments. The obtained results prove the important influence of the surrounding effect of symmetric α -stable mutations on both the abilities considered.

Keywords: evolutionary algorithms, Lévy-stable distributions, global optimization, surrounding effect

1. Introduction

Most applications of Evolutionary Algorithms (EAs) which employ the floating point representation of population individuals use the Gaussian mutation as a mutation operator (Bäck and Schwefel, 1993; Fogel, 1994; Fogel *et al.*, 1966; Galar, 1985; Michalewicz, 1996; Rechenberg, 1965). A new individual x is obtained by adding a normally distributed random value to each entry of a selected parent y :

$$x_i = y_i + N(0, \sigma_i), \quad i = 1, \dots, n. \quad (1)$$

The choice is usually justified by the Central Limit Theorem. Mutations in nature are caused by a variety of physical and chemical factors that are not identifiable or measurable. These factors are considered as independent and identically distributed (i.i.d.) random perturbations. The Generalized Central Limit Theorem states that the only possible non-trivial limit of normalized sums of i.i.d. terms is Lévy-stable (Lévy, 1925), also called α -stable or just stable in the mathematical literature (Fang *et al.*, 1990; Nolan, 2003; Samorodnitsky and Taqqu, 1994; Zolotariev, 1986). If the Lindeberg condition is obeyed, i.e., the first two absolute moments are finite, then the Lévy-Stable Distribution (LSD) reduces to the Gaussian distribution. The lack of closed form formulas for probability density functions (pdfs) for all but three LSDs (Gaussian (GD), Cauchy (CD) and Lévy (LD) distributions) has been a major drawback in the use of LSDs by practitioners. Fortunately, there exist algorithmic formu-

las for simulating Lévy-stable variables (Weron, 2001) as well as computer programs to compute Lévy-stable densities, distribution functions and quantiles (Nolan, 2003).

In recent years, the application of the CD to mutation operators in evolutionary algorithms has drawn researchers' attention (Bäck *et al.*, 1997; Kappler, 1996; Obuchowicz, 2003b; 2003c; Rudolph, 1997; Yao and Liu, 1996; 1997; 1999). While the univariate CD has a unique definition, there exist at least two multivariate versions of the CD: the spherical CD and the non-spherical CD with independent univariate Cauchy random variables in each dimension (Fang *et al.*, 1990; Obuchowicz, 2001b; Shu and Hartley, 1987). In these cases, the normally distributed random value $N(0, \sigma_i)$ (1) is substituted by a random variable of the one-dimensional CD. The shape of the Cauchy pdf resembles that of the Gaussian one, but it approaches the axis very slowly, increasing the probability of the so-called macro-mutations and local optimum leaving. Rudolph (1997) analyses analytically the local convergence of simple (1+1)ES and (1+ λ)ES with Gaussian, spherical and non-spherical Cauchy mutations. It was proved that the order of local convergence is identical to Gaussian and spherical Cauchy distributions, whereas non-spherical Cauchy mutations lead to slower local convergence. Yao and Liu (1996; 1997; 1999) successfully apply the non-spherical Cauchy mutation to evolutionary programming and evolutionary strategy algorithms in the case of solving global optimization problems of multivariate and multi-modal functions. Obuchowicz (2001b; 2003b; 2003c) presents comparison results of the

saddle crossing ability of EAs with Gaussian, spherical and non-spherical Cauchy mutations, as well as the influence of the choice of the reference frame on the effectiveness of EAs in global optimization tasks.

The suggestion that the application of LSDs other than the GD and CD can be very attractive for evolutionary algorithms with the floating-point representation of individuals was first introduced by Gutowski (2001), but this idea has not been pursued so far. In that work the author considered only some properties of LSDs without implementations to any evolutionary algorithm.

The aim of this work is to compare the effectiveness of EAs with mutations based on LSDs in global multidimensional optimization tasks. Implemented EAs are based on two known types of evolutionary models: evolutionary search with soft selection (ESSS), proposed by Galar (1985; 1989), and evolutionary programming (EP), introduced by Fogel (1994) and Fogel *et al.* (1966). The main difference between these two types of EAs is that the standard deviation of mutation is adapted in EP but not in ESSS.

This work is organized as follows: First, LSDs are defined in Section 2 and their properties in the one and multivariate cases and the method of numerical generation of Lévy-stable variables are described. The next part (Section 3) describes EAs used in simulation experiments. The main part containing the experimental studies is presented in Section 4, where hill climbing, saddle crossing as well as a set of multidimensional and multimodal optimization problems are considered using EAs with and without the self-adaptation mechanism of the scale parameter. Finally, the work is concluded.

2. Lévy-Stable Distributions

2.1. Characteristic Function Representation of the LSD

Due to the lack of closed form formulas for densities, the LSD can be most conveniently described by its characteristic function (ch.f.) $\phi(k)$ – the inverse Fourier transform of the pdf. The ch.f. of the LSD is parameterized by a quadruple $(\alpha, \beta, \sigma, \mu)$ (Weron, 2001), where α ($0 < \alpha \leq 2$) is a stability index (tail index, tail exponent or characteristic exponent), β ($-1 \leq \beta \leq 1$) is a skewness parameter, σ ($\sigma > 0$) is a scale parameter and μ is a location parameter. There are a variety of formulas of the LSD ch.f. in the relevant literature. This fact is caused by a combination of historical evolution and numerous problems that have been analyzed using specialized forms of LSDs. The most popular formula of the ch.f. of $X \sim S_\alpha(\beta, \sigma, \mu)$, i.e., a Lévy-stable random variable

with parameters α, β, σ and μ is given by (Weron, 1996):

$$\phi(k) = \begin{cases} \exp\left(-\sigma^\alpha |k|^\alpha \left\{1 - i\beta \operatorname{sign}(k) \times \tan\left(-\frac{\pi\alpha}{2}\right)\right\} + i\mu k\right) & \text{if } \alpha \neq 1, \\ \exp\left(-\sigma |k| \left\{1 + i\beta \operatorname{sign}(k) \times \frac{2}{\pi} \log |k|\right\} + i\mu k\right) & \text{if } \alpha = 1, \end{cases} \quad (2)$$

or, in a form more convenient for numerical purposes (Nolan, 2003),

$$\phi_0(k) = \begin{cases} \exp\left(-\sigma^\alpha |k|^\alpha \left\{1 + i\beta \operatorname{sign}(k) \times \tan\left(\frac{\pi\alpha}{2}\right) [(\sigma |k|)^{1-\alpha} - 1]\right\} + i\mu_0 k\right) & \text{if } \alpha \neq 1, \\ \exp\left(-\sigma |k| \left\{1 - i\beta \operatorname{sign}(k) \times \frac{2}{\pi} \log(\sigma |k|)\right\} + i\mu_0 k\right) & \text{if } \alpha = 1. \end{cases} \quad (3)$$

The $S_\alpha^0(\beta, \sigma, \mu_0)$ parameterization (3) is a variant of Zolotariev's (M)-parameterization (Zolotariev, 1986), with the ch.f. and pdf jointly continuous in all four parameters. The relation between location parameters in both representations is

$$\mu = \begin{cases} \mu_0 - \beta\sigma \tan\left(\frac{\pi\alpha}{2}\right) & \text{if } \alpha \neq 1, \\ \mu_0 - \frac{2\beta\sigma}{\pi} \log(\sigma) & \text{if } \alpha = 1. \end{cases}$$

Unfortunately, there are only three LSDs which have analytical forms of pdfs, i.e.,

the GD ($X \sim S_2(0, \sigma, \mu) = N(\mu, \sigma)$):

$$f_G(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \quad -\infty < x < \infty, \quad (4)$$

the CD ($X \sim S_1(0, \sigma, \mu) = C(\mu, \sigma)$):

$$f_C(x) = \frac{1}{\pi} \frac{\sigma}{\sigma^2 + (x-\mu)^2}, \quad -\infty < x < \infty, \quad (5)$$

and the LD ($X \sim S_{1/2}(1, \sigma, \mu) = Levy(\mu, \sigma)$):

$$f_L(x) = \sqrt{\frac{\sigma}{2\pi}} \frac{1}{(x-\mu)^{3/2}} \exp\left(-\frac{\sigma}{2(x-\mu)}\right), \quad \mu < x < \infty. \quad (6)$$

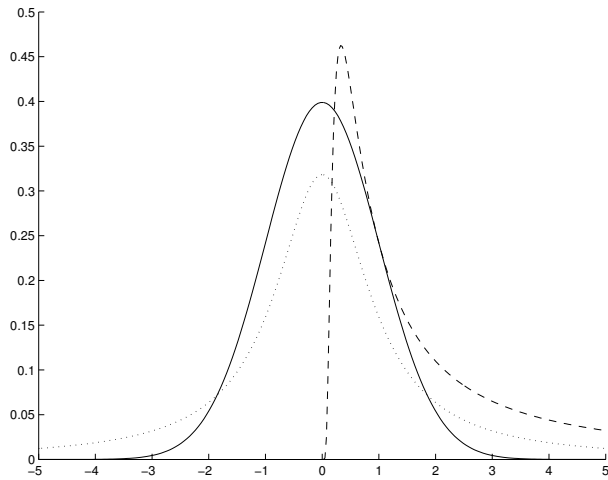


Fig. 1. Probability density functions of the standardized GD ($N(0, 1)$) – solid line, CD ($C(0, 1)$) – dotted line, and LD ($Levy(0, 1)$) – dashed line.

Figure 1 presents the pdfs of the standardized GD ($N(0, 1)$), CD ($C(0, 1)$), and LD ($Levy(0, 1)$).

2.2. Selected Properties of the LSD

The classical Central Limit Theorem says that the normalized sum of i.i.d. random variables with finite variance converges to a normal distribution. The Generalized Central Limit Theorem shows that if the finite variance assumption is dropped, the only possible resulting limits are stable.

Theorem 1. Generalized Central Limit Theorem (Lévy, 1925). Let X_1, X_2, X_3, \dots be an i.i.d. sequence of random variables. There exist constants $a_n > 0, b_n \in \mathbb{R}$

and a non-degenerate random variable Z with

$$a_n (X_1 + X_2 + \dots + X_n) - b_n \longrightarrow Z$$

if and only if Z is Lévy-stable for some $0 < \alpha \leq 2$.

A basic property of stable laws is that sums of Lévy-stable random variables are Lévy-stable. One of the effects connected with this fact is illustrated in Fig. 2, where the sum of two-dimensional Lévy-stable random vectors is presented. It is easy to see that the obtained graph has a ‘fractal’ nature. The existence of ‘long jumps’ as well as ‘short steps’ is independent on the graph scale. In the case of EAs this property makes it possible to obtain their good exploration and exploitation characteristics. In the independent case, the exact parameters of the sum of Lévy-stables random variables can be calculated using the propositions below (Nolan, 2003).

Proposition 1. The $S_\alpha^0(\beta, \sigma, \mu_0)$ parameterization (3) has the following properties:

1. If $X \sim S_\alpha^0(\beta, \sigma, \mu_0)$, then

$$\forall a \neq 0, b \in \mathbb{R},$$

$$aX + b \sim S_\alpha^0(\text{sign}(a)\beta, |a|\sigma, a\mu_0 + b).$$

2. The characteristic functions, densities and distribution functions are jointly continuous in all four parameters $(\alpha, \beta, \sigma, \mu_0)$.

3. If $X_1 \sim S_\alpha^0(\beta_1, \sigma_1, \mu_{01})$ and $X_2 \sim S_\alpha^0(\beta_2, \sigma_2, \mu_{02})$ are independent, then $X_1 + X_2 \sim S_\alpha^0(\beta, \sigma, \mu_0)$, where

$$\beta = \frac{\beta_1 \sigma_1^\alpha + \beta_2 \sigma_2^\alpha}{\sigma_1^\alpha + \sigma_2^\alpha}, \quad \sigma^\alpha = \sigma_1^\alpha + \sigma_2^\alpha,$$

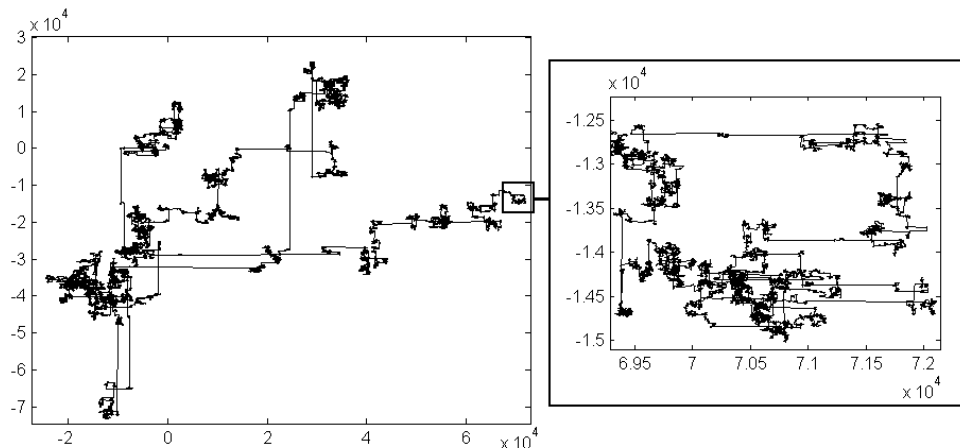


Fig. 2. Fractal nature of the sum of two-dimensional Lévy-stable random vectors ($\mathbf{X}_i \sim (S_{1/2}(0, 1, 0), S_{1/2}(0, 1, 0)) \mid i = 1, 2, \dots$).

$$\mu_0 = \begin{cases} \mu_{01} + \mu_{02} + \tan\left(\frac{\pi\alpha}{2}\right) [\beta\sigma - \beta_1\sigma_1 - \beta_2\sigma_2] & \text{if } \alpha \neq 1, \\ \mu_{01} + \mu_{02} + \frac{2}{\pi} [\beta\sigma \ln(\sigma) - \beta_1\sigma_1 \ln(\sigma_1) - \beta_2\sigma_2 \ln(\sigma_2)] & \text{if } \alpha = 1. \end{cases}$$

The formula $\sigma^\alpha = \sigma_1^\alpha + \sigma_2^\alpha$ above is a generalization of the rule for adding variances of independent random variables. It holds for both parameterizations.

Proposition 2. *The $S_\alpha(\beta, \sigma, \mu)$ parameterization (2) has the following properties:*

1. *If $X \sim S_\alpha(\beta, \sigma, \mu)$, then*

$$\forall a \neq 0, b \in \mathbb{R},$$

$$aX + b \sim \begin{cases} S_\alpha(\text{sign}(a)\beta, |a|\sigma, a\mu + b) & \text{if } \alpha \neq 1, \\ S_1(\text{sign}(a)\beta, |a|\sigma, a\mu + b - \beta\sigma \frac{2}{\pi} a \ln(a)) & \text{if } \alpha = 1. \end{cases}$$

2. *The characteristic functions, densities and distribution functions are continuous away from $\alpha = 1$, but discontinuous in any neighborhood of $\alpha = 1$.*

3. *If $X_1 \sim S_\alpha(\beta_1, \sigma_1, \mu_1)$ and $X_2 \sim S_\alpha(\beta_2, \sigma_2, \mu_2)$ are independent, then $X_1 + X_2 \sim S_\alpha(\beta, \sigma, \mu)$, where*

$$\beta = \frac{\beta_1\sigma_1^\alpha + \beta_2\sigma_2^\alpha}{\sigma_1^\alpha + \sigma_2^\alpha},$$

$$\sigma^\alpha = \sigma_1^\alpha + \sigma_2^\alpha, \quad \mu = \mu_1 + \mu_2.$$

A consequence of heavy tails of LSDs is that not all moments exist. In most statistical problems, the first moment $E(X)$ and variance $\text{Var}(X) = E(X^2) - (E(X))^2$ are routinely used to describe a distribution. In the case of LSDs such a representation is not useful since we have

$$\forall 0 < \alpha < 2,$$

$$E(X^p) = \int_{-\infty}^{\infty} x^p f(x) dx < +\infty \iff 0 < p < \alpha. \quad (7)$$

Thus, the second moment exists only for $\alpha = 2$, the first moment exists for $1 < \alpha \leq 2$ and is equal to the location parameter μ (2).

2.3. Simulation of Lévy-Stable Variables

If $U, U_1, U_2 \sim U(0, 1)$ are uniformly distributed random variables on the interval $(0, 1)$, then there are simple ways to generate stable random variables:

- for the normal case, the formulas

$$X_1 = \mu + \sigma \sqrt{-2 \ln(U_1)} \cos(2\pi U_2),$$

$$X_2 = \mu + \sigma \sqrt{-2 \ln(U_1)} \sin(2\pi U_2)$$

give two independent $N(\mu, \sigma)$ random variables,

- for the Cauchy case, the formula

$$X = \sigma \tan(\pi(U - 1/2)) + \mu$$

is $C(\mu, \sigma)$, and

- for the Lévy case, the formula

$$X = \sigma \frac{1}{Z^2} + \mu$$

is $Levy(\mu, \sigma)$, where $Z \sim N(0, 1)$.

In the general case, the complexity of the problem of simulating sequences of Lévy-stable random variables results from the fact that there is no analytical form for the inverse of the cumulative distribution function (cdf) apart from the GD, CD and LD. The first breakthrough was made by Kanter (1975), who gave a direct method for simulating $S_\alpha(1, 1, 0)$ random variables, for $\alpha < 1$. In general cases the following result of Chambers *et al.* (1976) gives a method for simulating any Lévy-stable random variable (Nolan, 2003).

Theorem 2. Simulating Lévy-stable random variables.

Let V and W be independent with $V \sim U(-\frac{\pi}{2}, \frac{\pi}{2})$, W exponentially distributed with the mean 1, $0 < \alpha \leq 2$.

1. *The symmetric random variable*

$$Z = \begin{cases} \frac{\sin(\alpha V)}{(\cos(V))^{1/\alpha}} \left[\frac{\cos((\alpha - 1)V)}{W} \right]^{(1-\alpha)/\alpha} & \text{if } \alpha \neq 1, \\ \tan(V) & \text{if } \alpha = 1 \end{cases}$$

has an $S_\alpha(0, 1, 0) = S_\alpha S$ distribution.

2. *In the non-symmetric case, for any $-1 \leq \beta \leq 1$, define $B_{\alpha, \beta} = \arctan(\beta \tan(\pi\alpha/2))/\alpha$ when $\alpha \neq 1$. Then*

$$Z = \begin{cases} \frac{\sin(\alpha(B_{\alpha, \beta} + V))}{(\cos(\alpha B_{\alpha, \beta}) \cos(V))^{1/\alpha}} \times \left[\frac{\cos(\alpha B_{\alpha, \beta} + (\alpha - 1)V)}{W} \right]^{(1-\alpha)/\alpha} & \text{if } \alpha \neq 1, \\ \frac{2}{\pi} \left[\left(\frac{\pi}{2} + \beta V \right) \tan(V) - \beta \ln \left(\frac{\frac{\pi}{2} W \cos(V)}{\frac{\pi}{2} + \beta V} \right) \right] & \text{if } \alpha = 1 \end{cases}$$

has an $S_\alpha(\beta, 1, 0)$ distribution.

It is easy to get V and W from independent uniform random variables $U_1, U_2 \sim U(0, 1)$: set $V = \pi(U_1 - \frac{1}{2})$ and $W = -\ln(U_2)$. Given the formulas for the simulation of standard Lévy-stable random variables (Theorem 2), a Lévy-stable random variable $X \sim S_\alpha(\beta, \sigma, \mu)$ for all admissible values of the parameters α, β, σ and μ has the form

$$X = \begin{cases} \sigma Z + \mu & \text{if } \alpha \neq 1, \\ \sigma Z + \frac{2}{\pi} \beta \sigma \ln(\sigma) + \mu & \text{if } \alpha = 1, \end{cases} \quad (8)$$

where $Z \sim S_\alpha(\beta, 1, 0)$.

2.4. Multivariate LSD

In this work only non-skewed ($\beta = 0$) LSDs will be applied to mutation operators of EAs. It is easy to see that the representations (2) and (3) are equivalent in this case. Thus, $Z \sim S_\alpha(0, \sigma, \mu) = S_\alpha S(\sigma, \mu)$ (symmetric α -stable) can be expressed by

$$Z = \mu + \sigma X, \quad (9)$$

where $X \sim S_\alpha S(1, 0) = S_\alpha S$ has the standardized symmetric α -stable distribution. The ch.f. of X is given by

$$\phi(k) = \exp(-|k|^\alpha). \quad (10)$$

For $\alpha = 1$, it is a ch.f. of the standardized CD, and for $\alpha = 2$, it becomes the ch.f. of the standardized GD. If $\mathbf{X} = (X_j \sim S_\alpha S \mid j = 1, 2, \dots, n) \sim \mathbf{S}_\alpha \mathbf{S}$ is a sample from a stable law, its ch.f. is given by

$$\phi(\mathbf{k}) = \exp(-\|\mathbf{k}\|_\alpha^\alpha), \quad (11)$$

where $\|\mathbf{a}\|_\alpha = (\sum_{j=1}^n |a_j|^\alpha)^{1/\alpha}$ denotes the l_α -norm.

If the ch.f. of \mathbf{X} is of the form (11), we say that \mathbf{X} possesses an α -symmetric multivariate distribution (Fang *et al.*, 1990). For $\alpha = 2$, the 2-symmetric multivariate distribution reduces to a spherical distribution. In other cases ($\alpha < 2$) the α -symmetric multivariate distribution is only invariant under the group of permutations. Let \mathcal{P} be the permutation group, i.e., if $\mathbf{H} \in \mathcal{P}$, then $\mathbf{H}^T \mathbf{H} = \mathbf{I}$ and the elements of \mathbf{H} are only 0 or 1. If $\mathbf{X} \sim \mathbf{S}_\alpha \mathbf{S}$, then $\mathbf{H} \mathbf{X} \sim \mathbf{S}_\alpha \mathbf{S}$. Figures 3–7 present selected 2-D and 3-D pdfs of α -symmetric multivariate distributions. The lack of spherical symmetry influences the relation between the effectiveness of an EA in a multimodal optimization task and reference frame selection. This fact, called the *symmetry effect*, was studied by Obuchowicz (2003b; 2003c), who analysed the non-spherical Cauchy mutation applied to the ESSS (Galar, 1985) and Evolutionary Programming (EP) (Fogel, 1994; Fogel *et al.*, 1966) algorithms. In (Obuchowicz, 2003b; 2003c) the 5D Rastrigin and Ackley functions were chosen as testing functions. Both functions considered are multimodal, but Rastrigin’s function

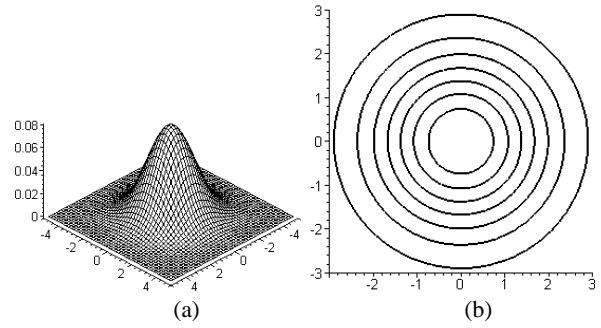


Fig. 3. 2-D probability density function (a) and its contour map (b) of $\mathbf{S}_\alpha \mathbf{S}$ for $\alpha = 2$ (GD).

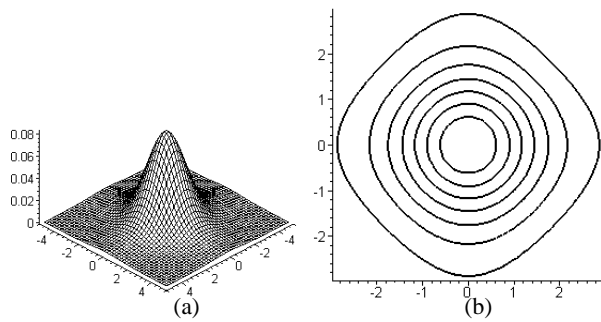


Fig. 4. 2-D probability density function (a) and its contour map (b) of $\mathbf{S}_\alpha \mathbf{S}$ for $\alpha = 1.5$.

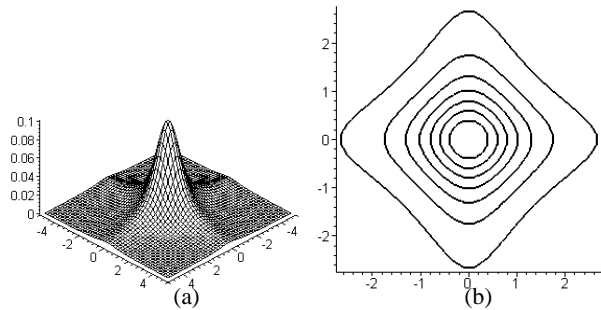


Fig. 5. 2-D probability density function (a) and its contour map (b) of $\mathbf{S}_\alpha \mathbf{S}$ for $\alpha = 1$ (CD).

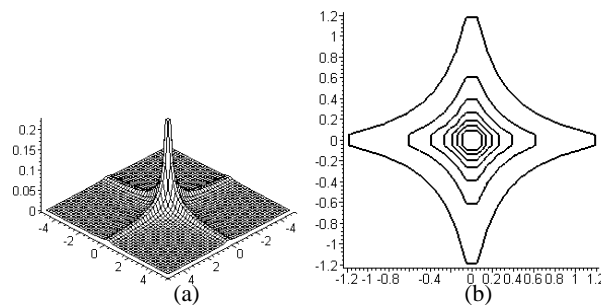


Fig. 6. 2-D probability density function (a) and its contour map (b) of $\mathbf{S}_\alpha \mathbf{S}$ for $\alpha = 0.5$.

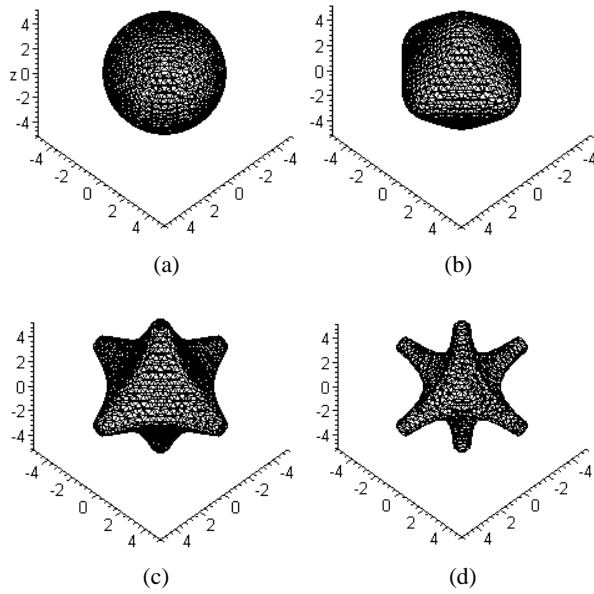


Fig. 7. Surfaces of equal values of 3-D pdfs ($f_\alpha(\mathbf{x}) = 0.001$) of $\mathbf{S}\alpha\mathbf{S}$ for $\alpha = 2$ (a), $\alpha = 1.5$ (b), $\alpha = 1$ (c), and $\alpha = 0.5$ (d).

characterizes the higher amplitude of changes and its valleys are deeper. Local optima of both functions are located in the nodes of the 5D-cubic net whose edges are parallel to the axes of the reference frame. Additionally, two other functions were taken into consideration. They were obtained from the 5D Rastrigin and Ackley functions after a rotation of the reference frame in the plane (x_1, x_2) through an angle equal to $\pi/4$, and in the plane (x_2, x_3) through an angle equal to $\pi/4$, too. The dependence of the EP algorithm with the Cauchy mutation on the choice of the reference frame manifested in slower convergence to the global optimum of the rotated testing functions in comparison with the non-rotated functions.

Another problem which seems to be imperceptible in the studies of α -symmetric multivariate distributions applied to mutation operations is related to the probability that the distance from the mutated point \mathbf{x} and its offspring \mathbf{y} will be in the range $\|\mathbf{x} - \mathbf{y}\| \in [r, r + dr]$. Figure 8 shows histograms of the distances between the origin and 10^6 points mutated with chosen distributions $\mathbf{S}\alpha\mathbf{S}$ for some space dimensions n . Although the pdfs of α -symmetric multivariate distributions have their maxima at origins, it is easy to prove (c.f. (Obuchowicz, 2001a; 2001b)) that the most probable distance of the offspring is near zero only in the case of a one-dimensional mutation. In the case of an n -dimensional mutation, the most probable distance increases with n , in the case of a Gaussian mutation it is proportional to the norm of the standard deviation vector and to $\sqrt{n-1}$. This fact, called the *surrounding effect* (Obuchowicz, 2001b), formed the basis for a simulation analysis of the ESSS and EP algo-

ritms with Gaussian and Cauchy mutations (Obuchowicz, 2003b; 2003c). The surrounding effect decreases the exploitation properties of an EA while increasing the dimension of the search space. This fact is also observed in the experiments described in Section 4.

3. Evolutionary Algorithms Used in Simulations

Two classes of evolutionary algorithms are used in simulation experiments. The first one is based on probably the simplest selection-mutation model of the Darwinian evolution, proposed and implemented (the ESSS algorithm) by Galar (1985). The searching process is executed in a multi-dimensional real space, on which some non-negative function, called the *fitness*, is defined. At the beginning, a population of points is randomly chosen from the searching space, and is iteratively changed by selection and mutation operators. As a selection operator the well-known proportional selection is used. Selected elements are mutated by adding a normally distributed random vector with a constant standard deviation vector. The second class is the well-known evolutionary programming model proposed by Fogel (1992) and Fogel *et al.* (1966; 1991). Apart from the different selection technique, the EP algorithm, unlike ESSS, possesses the adaptation mechanism of standard deviations of the mutation operator.

3.1. Evolutionary Search with Soft Selection

The assumptions described above can be formalized by the following algorithm: A real, n -dimensional search space (an adaptation landscape) \mathbb{R}^n is given. A non-negative function Φ to be maximized, called the *fitness*, is also defined on this adaptation landscape. First, an initial population $P(0)$ of η elements is randomly generated, e.g., by adding η times a normally distributed random vector to a given initial point $\mathbf{x}_0^0 \in \mathbb{R}^n$. The fitness index $q_k^0 = \Phi(\mathbf{x}_k^0)$ is calculated for each element \mathbf{x}_k^0 of the population. The searching process consists in generating a sequence of η -element populations. A new population $P(t+1)$ is created based only on the previous population $P(t)$. In order to generate a new element \mathbf{x}_k^{t+1} , a *parent* element is selected and mutated. Both selection and mutation are random processes. Each element \mathbf{x}_k^t can be chosen as a parent with a probability proportional to its fitness q_k^t (the well-known *roulette method*). A new element \mathbf{x}_k^{t+1} is obtained by adding a normally distributed random value to each entry of the selected parent:

$$(\mathbf{x}_k^{t+1})_i = (\mathbf{x}_k^t)_i + N(0, \sigma) \quad i = 1, \dots, n, \quad (12)$$

where the standard deviation σ is a parameter to be selected.

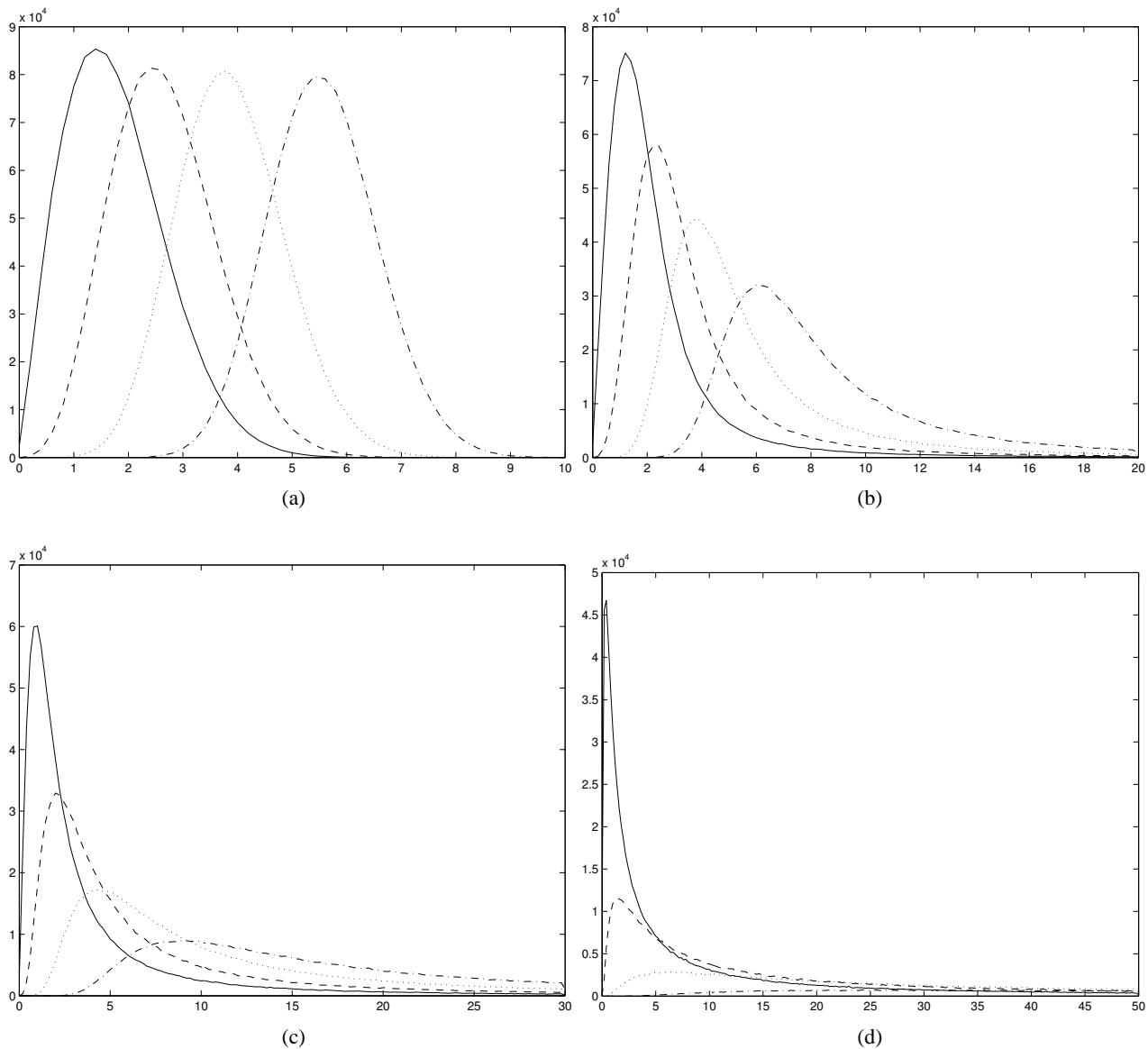


Fig. 8. Histograms of the distances between the origin and 10^6 points mutated with distributions $S\alpha S$ for $\alpha = 2$ (a), $\alpha = 1.5$ (b), $\alpha = 1$ (c), and $\alpha = 0.5$ (d) ($n = 2$ – solid line, $n = 4$ – dashed line, $n = 8$ – dotted line, and $n = 16$ – dash-dotted line).

Numerical tests of the ESSS algorithm (Galar, 1989) proved essential advantages of soft selection in comparison with *hard selection*, in which only the best individuals are chosen, and only local optima are attained. The ESSS algorithm does not constitute an optimization algorithm in the sense of reaching extrema with desired accuracy. The evolution process is not asymptotically convergent to an optimum, and the interpolation effectiveness of soft selection is rather weak. Evolution leads next generations to an elevated response surface rather than to maxima. In spite of that, search advantages of the ESSS algorithm suggest that this algorithm can be of real practical use in numerical packages for global optimization, especially when combined with local optimization algorithms.

Let $ESSS_\alpha$ denote the ESSS algorithm with the mutation based on the $S\alpha S$ distribution with a given α (Tab. 1). Four types of the ESSS algorithm, $ESSS_2$, $ESSS_{1.5}$, $ESSS_1$, and $ESSS_{0.5}$, are considered in this work. The pdfs and cdfs of $S\alpha S$ distributions used in the experiments are presented in Fig. 9.

The roulette method used as the selection operator possesses one main disadvantage. If disproportions between the element fitness values are much lower than the fitness values in the current population, then the distribution of the roulette method is similar to the uniform distribution. Thus, an evolutionary process is almost independent of the fitness function and reduces to some kind of random walk. In this work, the approach proposed in

Table 1. Outline of the ESSS $_{\alpha}$ algorithm

| |
|---|
| <p><i>Input data</i></p> <p>η – population size;</p> <p>t_{\max} – maximum number of iterations (epochs);</p> <p>σ – standard deviation of mutation;</p> <p>$\Phi : \mathbb{R}^n \rightarrow \mathbb{R}_+$ – non-negative fitness function, n – number of features;</p> <p>\mathbf{x}_0^0 – initial point.</p> <p>1. Initialize</p> <p>(a) $P(0) = (\mathbf{x}_1^0, \mathbf{x}_2^0, \dots, \mathbf{x}_\eta^0) : (\mathbf{x}_k^0)_i = (\mathbf{x}_0^0)_i + S\alpha S(0, \sigma); \quad i = 1, 2, \dots, n; \quad k = 1, 2, \dots, \eta;$</p> <p>(b) $q_0^0 = \Phi(\mathbf{x}_0^0).$</p> <p>2. Repeat</p> <p>(a) <i>Estimation</i></p> <p>$\Phi(P(t)) = (q_1^t, q_2^t, \dots, q_\eta^t),$ where $q_k^t = \Phi(\mathbf{x}_k^t), \quad k = 1, 2, \dots, \eta.$</p> <p>(b) <i>Choice of the best element in the history</i></p> <p>$(\mathbf{x}_0^t, \mathbf{x}_1^t, \mathbf{x}_2^t, \dots, \mathbf{x}_\eta^t) \rightarrow \mathbf{x}_0^{t+1}$ such that $q_0^{t+1} = \max\{q_k^t\}, \quad k = 0, 1, \dots, \eta.$</p> <p>(c) <i>Selection</i></p> <p>$(h_1, h_2, \dots, h_\eta),$ where $h_k = \min \left\{ h : \frac{\sum_{l=1}^h q_l^t}{\sum_{l=1}^{\eta} q_l^t} > \zeta_k \right\}$</p> <p>and $\{\zeta_k\}_{k=1}^{\eta}$ are random numbers uniformly distributed in $[0, 1).$</p> <p>(d) <i>Mutation</i></p> <p>$P(t) \rightarrow P(t+1);$</p> <p>$(\mathbf{x}_k^{t+1})_i = (\mathbf{x}_{h_k}^t)_i + S\alpha S(0, \sigma), \quad i = 1, 2, \dots, n; \quad k = 1, 2, \dots, \eta.$</p> <p>Until $t > t_{\max}.$</p> |
|---|

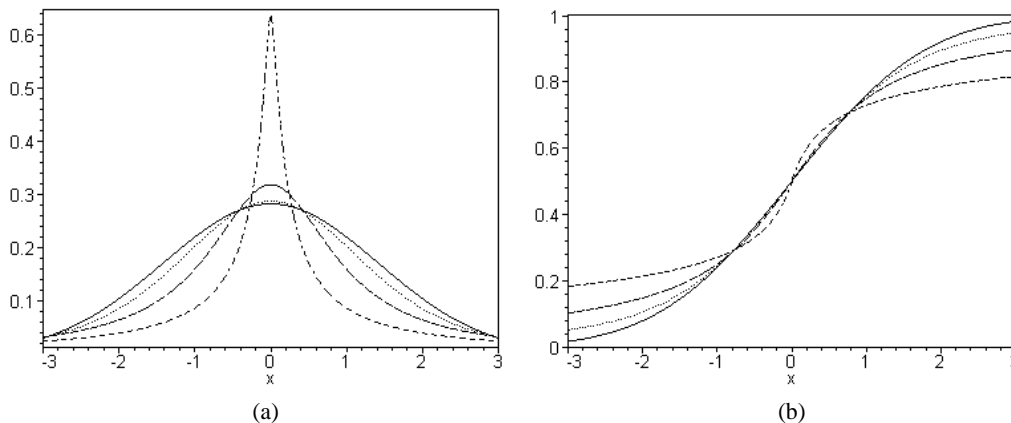


Fig. 9. Pdfs (a) and cdfs (b) of $S\alpha S$ distributions with $\alpha = 2$ (the Gaussian distribution) – solid line, $\alpha = 1.5$ – dotted line, $\alpha = 1$ (the Cauchy distribution) – dashed line, $\alpha = 0.5$ – dash-dotted line.

(Obuchowicz, 2003a) is applied. Let $f : \mathbb{R}^n \rightarrow \mathbb{R}$ be a function to be minimized. Then the fitness function is defined as follows:

$$\Phi(\mathbf{x}) = f_{\max}^t - f(\mathbf{x}) + \frac{1}{\eta^2}, \quad (13)$$

where $f_{\max}^t = \max f(\mathbf{x}_k^t \mid k = 1, \dots, \eta)$ is the maximal value of f over all elements in the current population. The last term in (13) is proportional to the probability of the worst element selection. This probability is very small but non-zero. The only limitation of this part is that it has to be significantly smaller than $1/\eta$ – the probability of element selection in the case of uniform random selection. The fitness function Φ is non-negative and its relative values in the current population make the roulette method effective.

3.2. Evolutionary Programming

Evolutionary programming was devised by L.G. Fogel *et al.* (1966) in the mid 1960s for the evolution of finite state machines in order to solve prediction tasks. EP was extended by Fogel (1992) and Fogel *et al.* (1991) to work on real-valued object variables based on normally distributed mutations. This algorithm is called the meta-EP (Fogel *et al.*, 1991) or the Classical EP (CEP) (Yao and Liu, 1999).

In meta-EP, an individual is represented by the pair $\mathbf{a} = (\mathbf{x}, \boldsymbol{\sigma})$, where $\mathbf{x} \in \mathbb{R}^n$ is a real-valued phenotype, $\boldsymbol{\sigma} \in \mathbb{R}_+^n$ is a self-adapted standard deviation vector for the Gaussian mutation. For initialization, EP assumes bounded subspaces $\Omega_x = \prod_{i=1}^n [u_i, v_i] \subset \mathbb{R}^n$ and $\Omega_\sigma = \prod_{i=1}^n [0, c] \subset \mathbb{R}_+^n$ with $u_i < v_i$ and $c > 0$. However, the search domain is extended to $\mathbb{R}^n \cup \mathbb{R}_+^n$ during the algorithm processing. As a mutation operator, a Gaussian mutation with a standard deviation vector assigned to an individual is used. All elements in the current population are mutated. Individuals from both parent and offspring populations participate in the process of selecting a new generation. For each individual \mathbf{a}_k , q individuals are chosen at random from $P(t) \cup P'(t)$ and compared with \mathbf{a}_k with respect to their fitness values. Here w_k is the number expressing how many of the q individuals are worse than \mathbf{a}_k . Then η individuals having the highest score w_k are selected from 2η parents and offspring to form a new population $P(t+1)$.

An analysis of the classical EP algorithm (Fogel, 1992) gives a proof of the global convergence with probability 1 for the resulting algorithm, and the result is derived by defining a Markov chain over the discrete state space that is obtained from reduction of the abstract search space \mathbb{R}^n to the finite set of numbers representable on a digital computer.

In order to introduce non-Gaussian mutations into the meta-EP algorithm, let EP_α denote the meta-EP al-

gorithm with the mutation based on the $\text{S}\alpha\text{S}$ distribution with a given α (Tab. 2). As a base meta-EP algorithm the description included in (Bäck and Schwefel, 1993; Yao and Liu, 1999) is chosen. However, Yao *et al.* (1999) introduce into their version of meta-EP (CEP) the self-adaptation schema of the standard deviation of the Gaussian mutation borrowed from another well-known phenotype evolution algorithm: evolutionary strategies (ES) (Bäck and Schwefel, 1993; Bäck *et al.*, 1997; Schwefel, 1981). This version of enriched meta-EP is used in this work. The self-adaptation scheme of the scale parameter (marked by the asterisk in Table 2) is an extension of that proposed for ES.

It is worth noticing that the application of the self-adaptation scheme influences the distribution of mutation. Figure 10 compares the pdfs of $\text{S}\alpha\text{S}(0, 1)$ and $\text{S}\alpha\text{S}(0, 1) \exp(\text{S}\alpha\text{S}(0, 1))$, the latter representing a simplified self-adapted random mutation. The mass of probability density is more concentrated around the central point and the tails are slightly heavier in the case of the $\text{S}\alpha\text{S}(0, 1) \exp(\text{S}\alpha\text{S}(0, 1))$ distribution. This fact can manifest itself by increased the numbers of ‘short steps’ and ‘long jumps’ at the cost of ‘mean steps’.

4. Simulation Experiments

4.1. Study of Hill Climbing Using (1+1)ES

4.1.1. Problem Statement

Before the ESSS_α and EP_α algorithms are used, let us consider the simple modification (1+1) ES_α of the evolutionary strategy (1+1)ES (Rechenberg, 1965). The population at the iteration t is reduced to only one element \mathbf{x}^t , from which an offspring \mathbf{y}^t is generated by a mutation operator. The mutation is defined by

$$\mathbf{y}^t = \mathbf{x}^t + \sigma \mathbf{Z}, \quad (14)$$

where $\mathbf{Z} \sim \text{S}\alpha\text{S}$ for a given α , σ is an input parameter. A better element (in the sense of the fitness function) from the parent \mathbf{x}^t and the offspring \mathbf{y}^t is chosen as a new base element \mathbf{x}^{t+1} , i.e.,

$$\mathbf{x}^{t+1} = \begin{cases} \mathbf{x}^t & \text{if } \Phi(\mathbf{x}^t) > \Phi(\mathbf{y}^t), \\ \mathbf{y}^t & \text{otherwise.} \end{cases} \quad (15)$$

Replacing $t \leftarrow t + 1$, the operations (14) and (15) are repeated iteratively until a given stopping criterion is met.

The aim of this section is to analyze the exploitation effectiveness of the above (1+1) ES_α algorithm. Let us consider the spherical function

$$f_{sph}(\mathbf{x}) = \|\mathbf{x}\|^2 \quad (16)$$

Table 2. Outline of the EP_α algorithm

| |
|---|
| <p><i>I. Initiation</i></p> <p><i>A. Random generation</i></p> $P(0) = \{\mathbf{a}_k(0) = (\mathbf{x}_k(0), \boldsymbol{\sigma}_k(0)) \mid k = 1, 2, \dots, \eta\}.$ $\mathbf{x}_k(0) = \text{RANDOM}(\Omega_x), \boldsymbol{\sigma}_k(0) = \text{RANDOM}(\Omega_\sigma), \quad \Omega_x \subset \mathbb{R}^n, \Omega_\sigma \subset \mathbb{R}_+^n.$ <p><i>B. Estimation</i></p> $P(0) \rightarrow \Phi(P(0)) = \{q_k(0) = \Phi(\mathbf{x}_k(0)) \mid k = 1, 2, \dots, \eta\}.$ <p><i>C. $t = 1$.</i></p> <p><i>II. Repeat:</i></p> <p><i>A. Mutation</i></p> $P'(t) = m_{\tau, \tau'}(P(t)) = \{\mathbf{a}'_k(t) \mid k = 1, 2, \dots, \eta'\}.$ $x'_{ki}(t) = x_{ki} + S\alpha S_i(0, \sigma_{ki}), \sigma'_{ki} = \sigma_{ki} \exp(\tau' S\alpha S(0, 1) + \tau S\alpha S_i(0, 1)), (*) \quad i = 1, 2, \dots, n,$ <p>$S\alpha S_i(0, 1)$ indicates that the random number is generated anew for each component i.</p> <p><i>B. Estimation</i></p> $P'(t) \rightarrow \Phi(P'(t)) = \{q'_k(t) = \Phi(\mathbf{x}'_k(t)) \mid k = 1, 2, \dots, \eta'\}.$ <p><i>C. Selection of a new generation</i></p> $P(t+1) = s_{\theta_n}^n(P(t) \cup P'(t)) = \{\mathbf{a}_k(t+1) \mid k = 1, 2, \dots, \eta\}.$ $\forall \mathbf{a}_k \in P(t) \cup P'(t), \quad \mathbf{a}_k \rightarrow \{\mathbf{a}_{kl} = \text{RANDOM}(P(t) \cup P'(t)) \mid l = 1, 2, \dots, q\},$ $w_k = \sum_{l=1}^q \theta(\Phi(\mathbf{x}_k) - \Phi(\mathbf{x}_{kl})), \theta(\alpha) = \begin{cases} 0 & \text{for } \alpha < 0 \\ 1 & \text{for } \alpha \geq 0 \end{cases}$ <p>$P(t+1) \leftarrow \eta$ individuals with the best score w_k.</p> <p><i>D. $t = t + 1$.</i></p> <p><i>Until</i> $(\iota(P(t)) = \text{true})$.</p> |
|---|

as an objective function to be minimized. The experiment consists in starting the $(1+1)ES_\alpha$ algorithm many times from different starting points and calculating the percentage of successful mutation operations ζ , i.e., we are interested in percentages of mutations resulting in better offspring than their base points.

4.1.2. Experiment and Results

The simulations were performed for the 4D sphere function (16). Four starting points were selected: $\mathbf{a}_1 = (100, 0, 0, 0)$, $\mathbf{a}_2 = (100/\sqrt{2}, 100/\sqrt{2}, 0, 0)$, $\mathbf{a}_3 = (100/\sqrt{3}, 100/\sqrt{3}, 100/\sqrt{3}, 0)$, and $\mathbf{a}_4 = (50, 50, 50, 50)$. It is worth noticing that $\|\mathbf{a}_i\| = 100$, $i = 1, 2, 3, 4$. Four algorithms from the $(1+1)ES_\alpha$ class are used in this experiment: $(1+1)ES_2$, $(1+1)ES_{1.5}$, $(1+1)ES_1$,

and $(1+1)ES_{0.5}$. The scale parameter for all algorithms is the same: $\sigma = 0.1$. Each algorithm is started from each starting point 500 times.

Figure 11 shows the percentage of successful mutations obtained for all algorithms used and all starting points. Let $\zeta_{\alpha, i}$ be the percentage of successful mutations of $(1+1)ES_\alpha$ ($\alpha = 2, 1.5, 1, 0.5$) started from the point \mathbf{a}_i ($i = 1, 2, 3, 4$).

Observation 1. *It is easy to see that $\zeta_{2, i} \approx 50\%$ does not depend on the starting point.*

Observation 2. *If α decreases, then $\zeta_{\alpha, i}$ rapidly decreases and the disproportion between the results for different starting points increases. The percentage $\zeta_{\alpha, i}$ is smaller for starting points located far from the axes of the reference frame.*

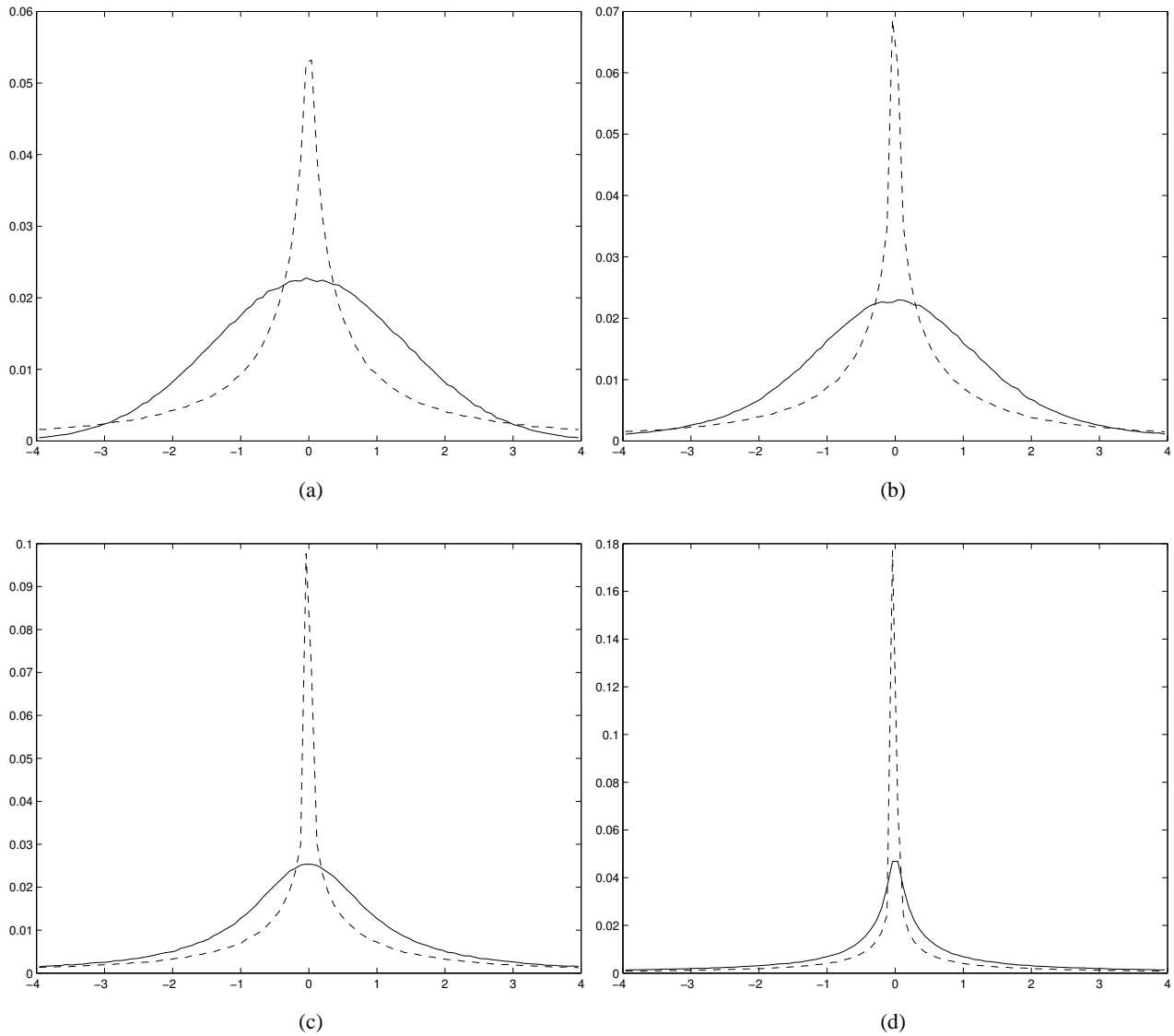


Fig. 10. Pdfs of $S_{\alpha}S(0,1)$ (solid line) and $S_{\alpha}S(0,1) \exp(S_{\alpha}S(0,1))$ (dashed line) for $\alpha = 2$ (a), $\alpha = 1.5$ (b), $\alpha = 1$ (c), and $\alpha = 0.5$ (d).

Let $\bar{t}_{\alpha,i}$ be the average number of iterations needed to locate the optimum (the stopping criterion $f_{sph} < 0.5$) taken over 500 runs of the $(1+1)ES_{\alpha}$ algorithm ($\alpha = 2, 1.5, 1, 0.5$) started from the point \mathbf{a}_i ($i = 1, 2, 3, 4$) (see Fig. 12).

Observation 3. *It is surprising that, in spite of the dependence of $\zeta_{\alpha,i}$ on α and \mathbf{a}_i , both $\bar{t}_{2,i}$ and $\bar{t}_{0.5,i}$ seem to be almost independent of the starting point. Moreover, one can suspect that $\bar{t}_{0.5,4} < \bar{t}_{0.5,1}$ (when $\zeta_{0.5,4} \ll \zeta_{0.5,1}$, see Observation 2); however, the difference between both the numbers is in the limit of the statistical error. The largest dependence of $\bar{t}_{\alpha,i}$ on \mathbf{a}_i is obtained for $\alpha = 1.5$.*

4.1.3. Conclusions

The independence of $\zeta_{2,i}$ from the selecting starting point (see Observation 1) follows from the fact that the GD possesses a spherical symmetry and the level curve of its pdf (described by $\sigma = 0.1$) is much smaller than the level curve of the f_{sph} slope.

The results described in Observation 2 are caused by the α -symmetry of $S_{\alpha}S$ used in mutation, which prefers directions parallel to the axes of the reference frame. Thus, when \mathbf{a}_1 , which is located on the axis of the reference frame, is chosen as a starting point, then the mutation operator possesses the highest probability of

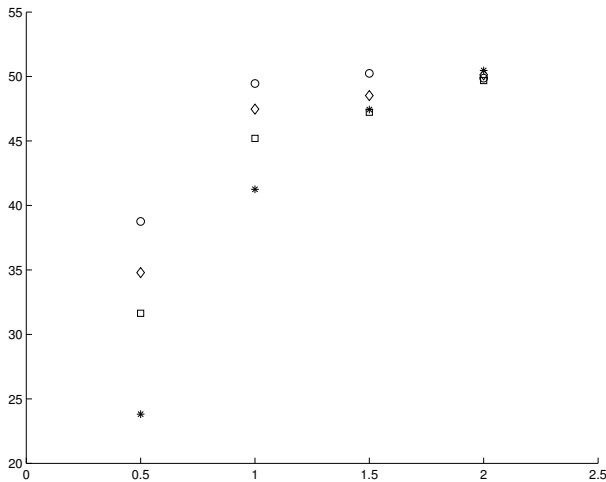


Fig. 11. Percentage of successful mutations in (1+1)ES $_{\alpha}$ started from different points (a_1 – stars, a_2 – squares, a_3 – diamonds, a_4 – circles) vs. α .

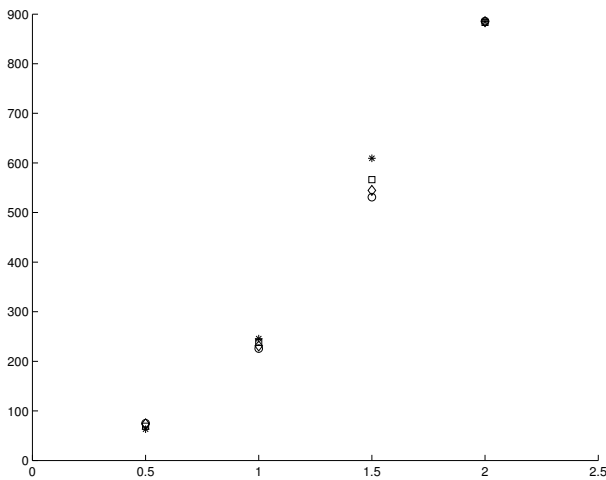


Fig. 12. Average number of iterations needed to locate the optimum neighborhood ($f_{sph} < 0.5$) for (1+1)ES $_{\alpha}$ started from different points (a_1 – stars, a_2 – squares, a_3 – diamonds, a_4 – circles) vs. α .

allocating offspring in an area of better values of f_{sph} . This probability rapidly decreases when the order of the diagonal on which the starting point is allocated increases. The disproportion of the results for given (1+1)ES $_{\alpha}$ and a different starting point increases with a decrease in α .

Observation 3 suggests that there are two competing mechanisms influencing $\bar{t}_{\alpha,i}$. The first, described above, is the relation between the selection of the starting point and $\zeta_{\alpha,i}$. The second is connected with heavy tails of $S\alpha S$. For low values of α , however, most of the mutations result in worse offspring, but an average “jump” of a successful mutation is much longer than in the case of higher α .

4.2. Optimization of Unimodal Functions

4.2.1. Problem Formulation

Consider two unimodal functions: the sphere function $f_{sph}(\mathbf{x})$ (16) and generalized Rosenbrock function

$$f_{GR}(\mathbf{x}) = \sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right], \quad (17)$$

where n is the search space dimension.

The ESSS $_{\alpha}$ and EP $_{\alpha}$ ($\alpha = 2, 1.5, 1, 0.5$) algorithms are considered in this section. The goal of this experiment is to analyze exploitation effectiveness of the examined algorithms in the sense of the best point convergence to an optimum.

4.2.2. Experiment and Results

Four algorithms from the ESSS $_{\alpha}$ class are used in this experiment: ESSS $_2$, ESSS $_{1.5}$, ESSS $_1$, and ESSS $_{0.5}$. All of them are applied in order to adapt to the 2D, 5D, 10D and 20D landscapes defined by the functions $f_{sph}(\mathbf{x})$ (16) and $f_{GR}(\mathbf{x})$ (17). The initial population is obtained by $\eta = 20$ mutations of the starting point $\mathbf{x}_0^0 = (30/\sqrt{n}, \dots, 30/\sqrt{n})$ ($\|\mathbf{x}_0^0\| = 30$). The scale parameter for all algorithms is the same: $\sigma = 0.05$. Each algorithm is started 500 times for each set of initial parameters. The maximum number of epochs is set as $t_{max} = 5000$ for $f_{sph}(\mathbf{x})$ (16) and $t_{max} = 10000$ for $f_{GR}(\mathbf{x})$ (17). Figures 13 and 14 present the convergence of the best element in the current population to the optimum for different α and n in the case of $f_{sph}(\mathbf{x})$ and $f_{GR}(\mathbf{x})$, respectively.

Observation 4. In both cases of $f_{sph}(\mathbf{x})$ and $f_{GR}(\mathbf{x})$, ESSS $_{\alpha}$ algorithms with low values of α more quickly converge to the optimum, but they localize the optimum with worse accuracy, i.e., the population becomes stabilized on higher values of the objective function. The accuracy decreases while the search space dimension increases. In the limit of the lowest α and high dimension, the stabilized population places their elements further from the optimum point than \mathbf{x}_0^0 (it is clearly seen for ESSS $_{0.5}$).

Before the cause of the observed facts is explained, let us introduce the following helpful experiment: Consider the sphere function $f_{sph}(\mathbf{x})$ (16). All parameters are the same as in the previous experiment except for the starting point which is now located at the optimum point $\mathbf{x}_0^0 = (0, \dots, 0)$. Figures 15 and 16 illustrate how far from the optimum point the population fluctuates in its stable state.

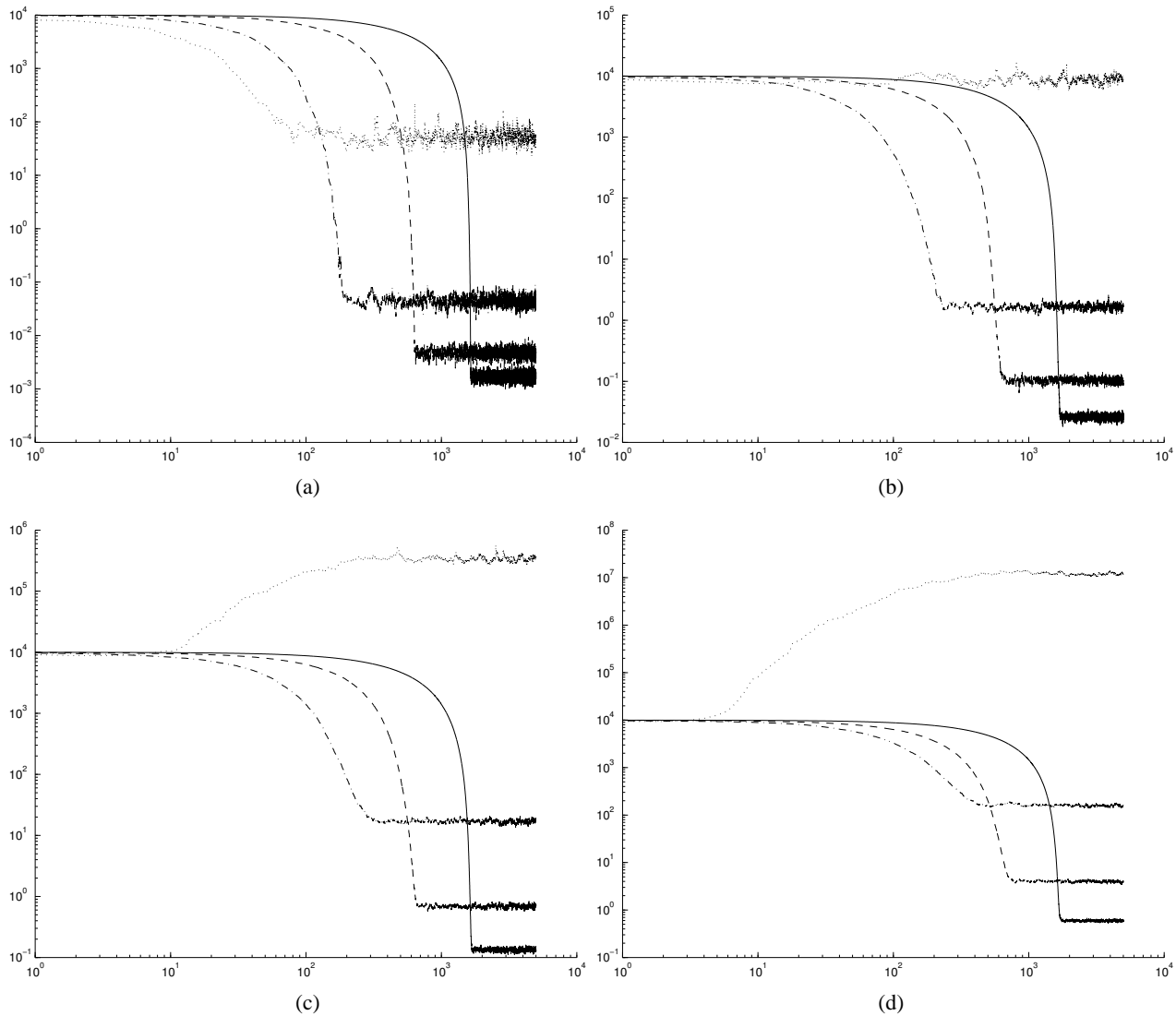


Fig. 13. Spherical function $f_{sph}(x)$ (16) optimized by the $ESSS_\alpha$ algorithm. The best value of $f_{sph}(x)$ in the population vs. iterations (results averaged over 500 algorithm runs) in the 2D (a), 5D (b), 10D (c), and 20D (d) search spaces. ($\alpha = 2$ – solid line, $\alpha = 1.5$ – dashed line, $\alpha = 1$ – dash-dotted line, and $\alpha = 0.5$ – dotted line).

Observation 5. *It is easy to see that the population in the stable state fluctuates around the optimum further and further with an increase in the space dimension. These distances rapidly grow with a decrease in the index of stability α .*

In order to analyse whether the same effect as those described in Observations 4 and 5 will be obtained in the case of EAs with the self-adaptation mechanism of the scale parameter, let us consider the following experiment: Four algorithms from the EP_α class are used: EP_2 , $EP_{1.5}$, EP_1 , and $EP_{0.5}$. All of them are applied in order to adapt to the 2D and 20D landscapes defined by the functions $f_{sph}(x)$ (16). The initial population of $\eta = 20$ individuals is selected from $\Omega_x = \prod_{i=1}^n [-10, 10]$ and $\Omega_\sigma = \prod_{i=1}^n [0, 0.05]$. The number of sparring partners

is $q = 5$. The maximum number of epochs is set as $t_{max} = 1000$. Each algorithm is started 50 times for each set of initial parameters. Figure 17 presents the convergence of the best element in the current population to the optimum for different α and n (the results are averaged over 50 algorithm runs). The relations between the average scale parameter in the current population vs. iterations for a chosen algorithm run are shown in Figs. 18 and 19.

Observation 6. *The convergence of the EP_α algorithm to the local optimum (Fig. 17) is quite different than in the case of $ESSS_\alpha$ (Fig. 13). The EP_α algorithm with high values of α converges to the optimum more quickly. Unlike $ESSS_\alpha$, where some stable state is detected, the population in EP_α (slower and slower with time) continually converges to the local optimum.*

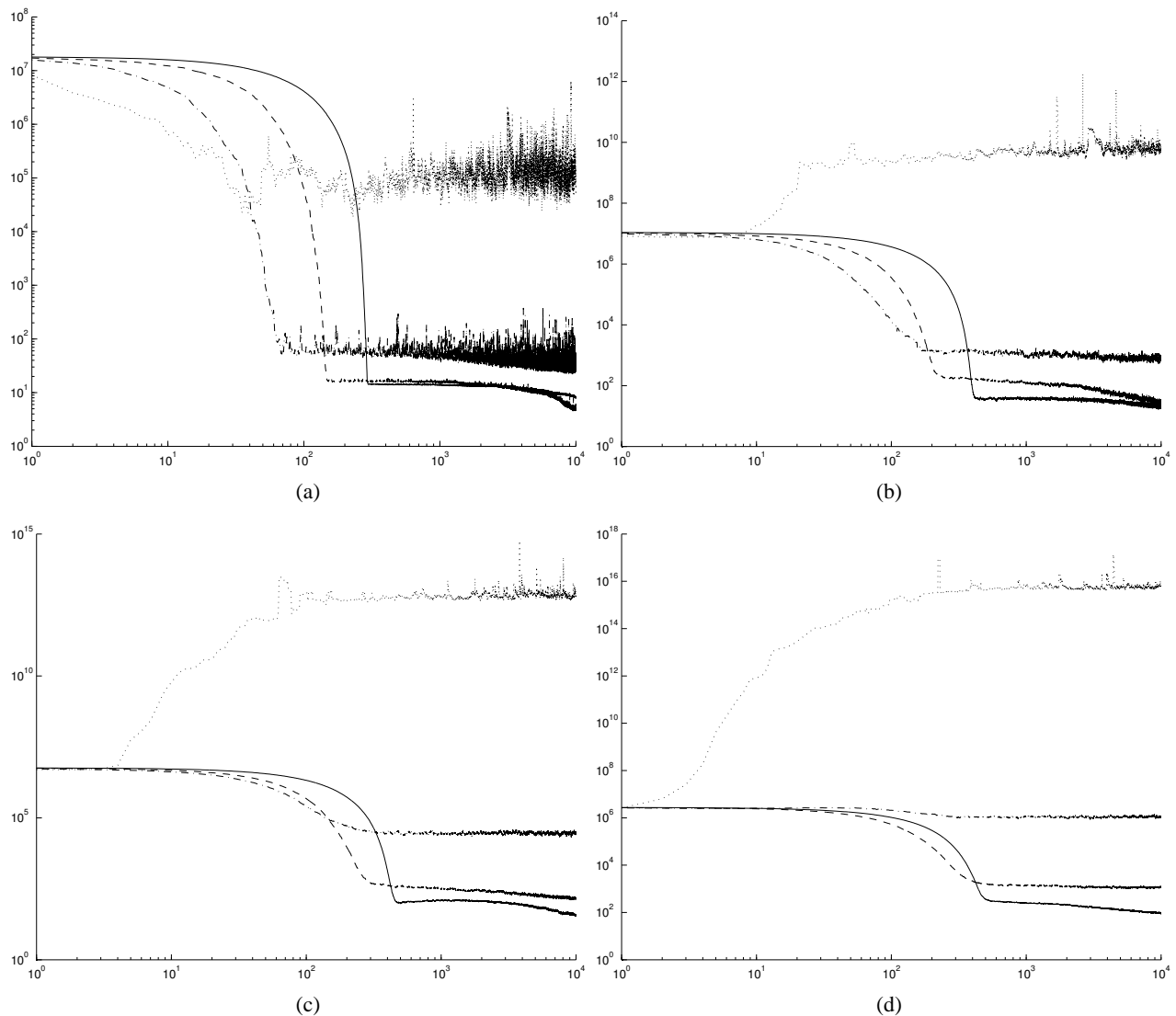


Fig. 14. Generalized Rosenbrock function $f_{GR}(x)$ (17) optimized by the $ESSS_{\alpha}$ algorithm. The best value of $f_{GR}(x)$ in the population vs. iterations (results averaged over 500 algorithm runs) in the 2D (a), 5D (b), 10D (c), and 20D (d) search spaces. ($\alpha = 2$ – solid line, $\alpha = 1.5$ – dashed line, $\alpha = 1$ – dash-dotted line, and $\alpha = 0.5$ – dotted line).

Observation 7. *Two facts can be observed in Figs. 18 and 19. The average scale parameter (taken over all elements in the actual population) rapidly decreases to very small values in the case of low α and high dimensions, and shows more chaotic behavior in comparison with high α .*

4.2.3. Conclusions

The results presented in Observations 4 and 5 are caused by two mechanisms. The first one is connected with the existence of heavy tails of the $S\alpha S$ pdfs; they improve the convergence to the optimum neighborhood with a decrease in α . But the optimum approximation error becomes worse with a decrease in α because of the sur-

rounding effect described in Section 2.4. This effect of multivariate LSDs does not allow us to locate offspring in the close vicinity of the the base point. Some kind of *dead surrounding*, where the allocation of offspring is almost impossible, is created. The radius of the *dead surrounding* increases with the search space dimension n and a decrease in the index of stability α .

The effect of *dead surrounding* is reduced by the self-adaptation mechanism of a scale parameter of mutation (Observations 6 and 7). But the necessity of dead surrounding reduction results in a rapid decrease in a scale parameter to very small values, especially in the case of low α and high dimensions. This fact, among other things, causes a low convergence to the optimum of the EP_{α} algorithm with low values of α .

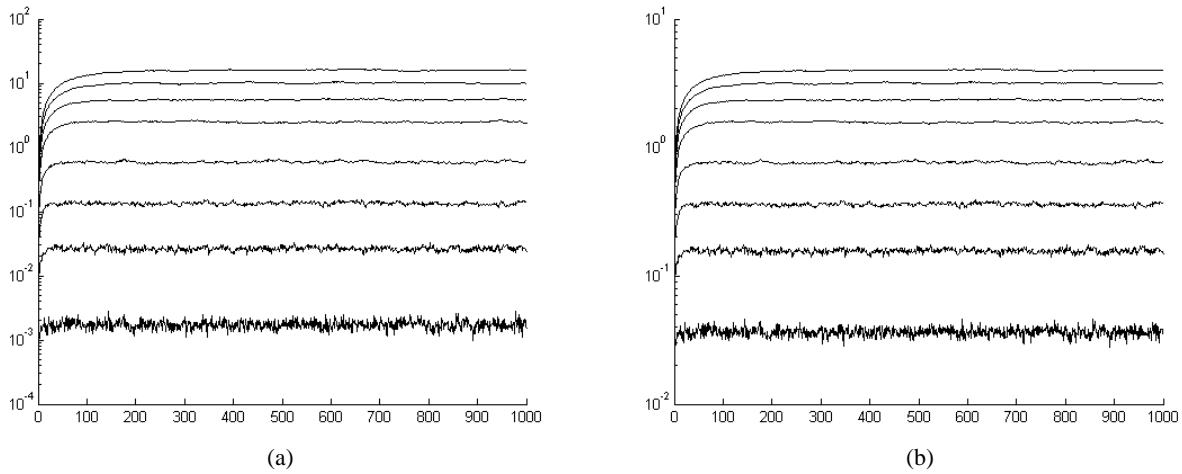


Fig. 15. ESSS₂ process for $f_{sph}(\mathbf{x})$ started at the optimum point. The best value of the objective function in the current population (a), and the distance between the best point in the population and the optimum point (b) vs. iterations (results averaged over 100 algorithms runs, $n = 2, 5, 10, 20, 40, 60, 80, 100$ from the bottom to top curves, respectively).

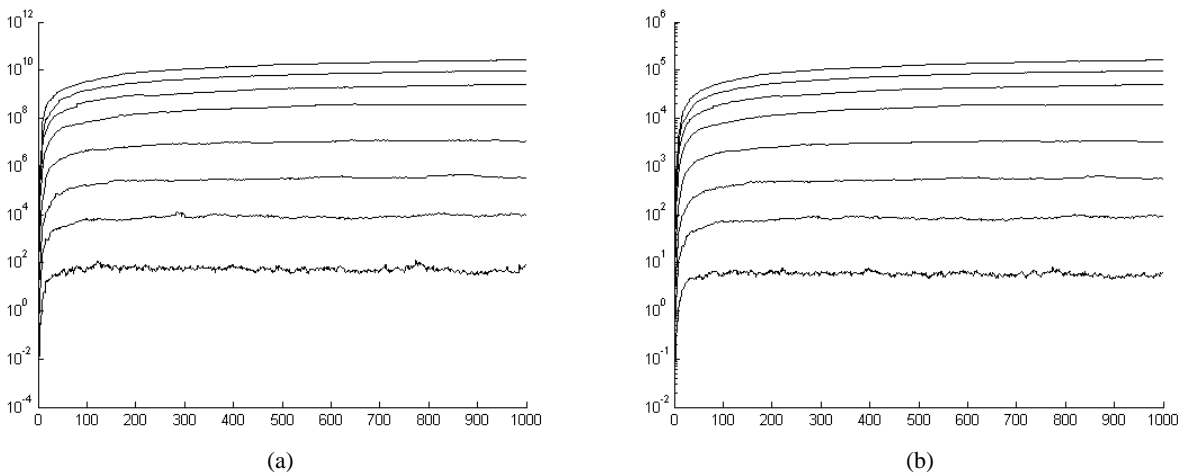


Fig. 16. ESSS_{0.5} process for $f_{sph}(\mathbf{x})$ started at the optimum point. The best value of the objective function in the current population (a), and the distance between the best point in the population and the optimum point (b) vs. iterations (results averaged over 100 algorithms runs, $n = 2, 5, 10, 20, 40, 60, 80, 100$ from the bottom to top curves, respectively).

4.3. Saddle-Crossing Ability

4.3.1. Problem Statement

The saddle-crossing ability problem was defined by Galar (1989) and was then extensively studied for ESSS₂, ESSS₁, EP₂, and EP₁ (ESSS and EP algorithms with Gaussian and Cauchy mutations, respectively) by Obuchowicz (2003b; 2003c). As the fitness function the sum of two Gaussian peaks (Fig. 20(a)) is adapted:

$$\begin{aligned} \Phi_{sc}(\mathbf{x}) = & \exp\left(-5 \sum_{i=1}^n x_i^2\right) \\ & + \frac{1}{2} \exp\left(-5\left((1-x_1)^2 + \sum_{i=2}^n x_i^2\right)\right), \end{aligned} \quad (18)$$

where n is the landscape dimension. The lowest peak has its optimum at the point $(1, 0, \dots, 0)$. The global optimum is located at the point $(0, 0, \dots, 0)$.

The aim of this experiment is a simulation analysis of the saddle-crossing ability, which is measured by the number of iterations needed to cross a saddle between the lower and higher peaks. The starting population is allocated in the neighborhood of the lower optimum $\mathbf{x}_0^0 = (1, 0, \dots, 0)$. The searching process is stopped when at most half of the elements of the current population are located on the global peak higher than the lowest local optimum (it is called the *successful run*), i.e.,

$$\begin{aligned} \Phi_{sc}(\mathbf{x}_k) > \phi_{lim} > \Phi_{sc}(\mathbf{x}_0^0) \\ \text{for most } k \text{ of } k = 1, 2, \dots, \eta, \end{aligned} \quad (19)$$

or the maximum number of iterations t_{\max} is exceeded.

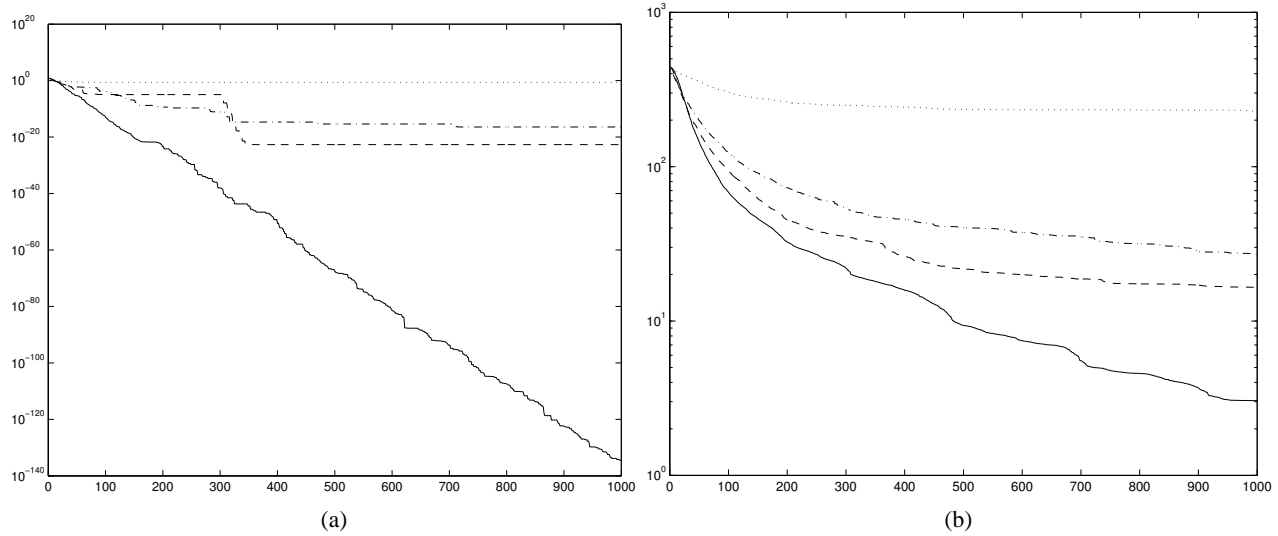


Fig. 17. Spherical function $f_{sph}(\mathbf{x})$ (16) optimized by the EP_α algorithm. The best value of $f_{sph}(\mathbf{x})$ in the population vs. iterations (results averaged over 50 algorithm runs) in the 2D (a) and 20D (b) search spaces. ($\alpha = 2$ – solid line, $\alpha = 1.5$ – dashed line, $\alpha = 1$ – dash-dotted line, and $\alpha = 0.5$ – dotted line).

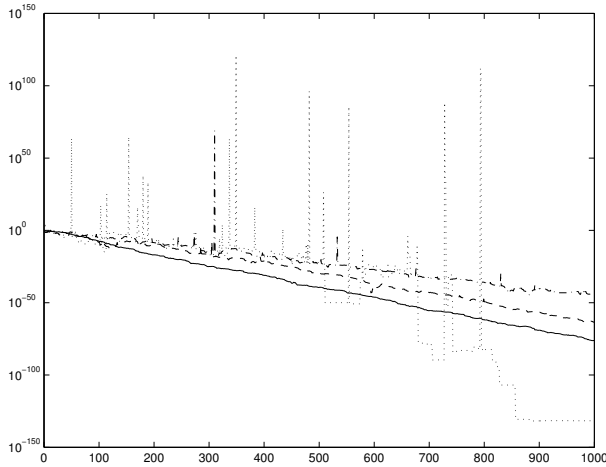


Fig. 18. Spherical function $f_{sph}(\mathbf{x})$ (16) optimized by the EP_α algorithm. The average scale parameter σ in the current population vs. iterations in the 2D search space. ($\alpha = 2$ – solid line, $\alpha = 1.5$ – dashed line, $\alpha = 1$ – dash-dotted line, and $\alpha = 0.5$ – dotted line).

Each algorithm is started a given number of samples. Two parameters are chosen as measures of algorithm effectiveness in this experiment. The first one is the percentage ζ of runs in which the process is stopped before t_{max} iterations, i.e., the condition (19) is fulfilled. The second one is the average number \bar{t} of iterations which is needed to cross the saddle by a given algorithm.

4.3.2. Experiment and Results

In the first part of this experiment, four algorithms from the $ESSS_\alpha$ class are applied: $ESSS_2$, $ESSS_{1.5}$, $ESSS_1$,

and $ESSS_{0.5}$. The following input parameters (the same for all algorithms) are used: the population size $\eta = 20$, the scale parameter $\sigma = 0.05$, the maximum number of iterations $t_{max} = 10^5$, the limit fitness $\phi_{lim} = 0.6$. Each algorithm is started 500 times for each set of initial parameters. The results are presented in Fig. 20(b).

Observation 8. *In the case of very low dimensions, the effectiveness of saddle crossing is best for the lowest value of $\alpha = 0.5$ and decreases when the index of stability α increases.*

Observation 9. *The high efficiency of $ESSS_\alpha$ with low values of α and low dimensions rapidly decreases when the search space dimension increases. In the limit of high dimensions, the ordering of $ESSS_\alpha$ algorithms with respect to their saddle-crossing ability is reversed.*

Observation 10. *The dependence between the $ESSS_\alpha$ saddle-crossing ability and the search space dimension n described in Observation 9 changes in the case of $ESSS_2$. Here \bar{t} for the $ESSS_2$ algorithm is very high for $n = 1$ and decreases with the space dimension to some level achieved for $n \approx 6$. Such an “equilibrium” is kept until $n \approx 14$. For $n > 14$, the dependence between \bar{t} and the search space dimension n is the same as that described in Observation 9.*

In the second part of this experiment, four algorithms from the EP_α class are applied: EP_2 , $EP_{1.5}$, EP_1 , and $EP_{0.5}$. The initial population of $\eta = 20$ individuals is selected from $\Omega_x = [0.8, 1.2] \times \prod_{i=2}^n [-0.2, 0.2]$ and $\Omega_\sigma = \prod_{i=1}^n [0, 0.01]$. The number of sparring partners is $q = 5$. The maximum number of epochs is set as

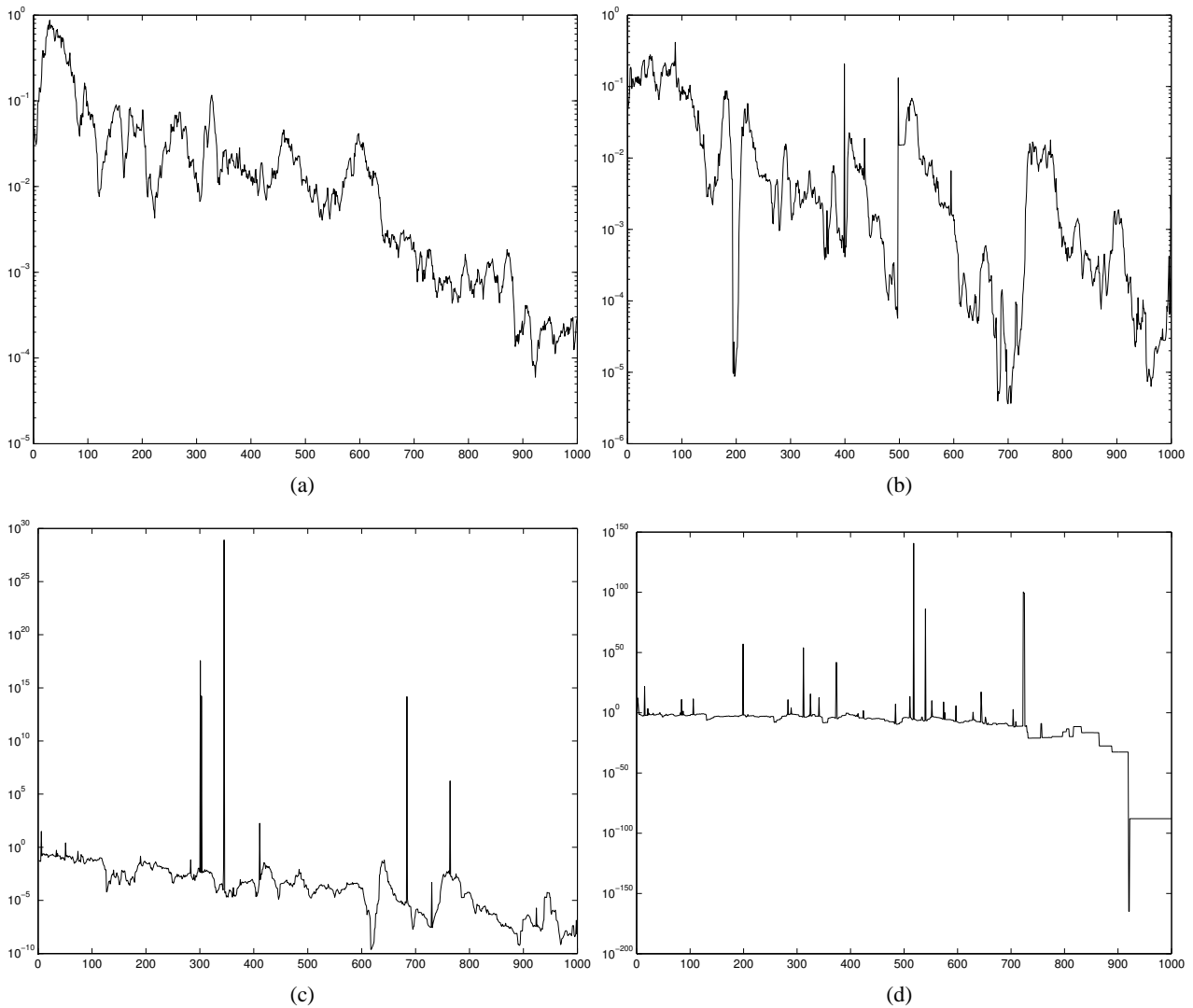


Fig. 19. Spherical function $f_{sph}(\mathbf{x})$ (16) optimized by the EP_α algorithm. The average scale parameter σ in the current population vs. iterations in the 20D search space for $\alpha = 2$ (a), $\alpha = 1.5$ (b), $\alpha = 1$ (c), and $\alpha = 0.5$ (d).

$t_{\max} = 1000$, the limit fitness $\phi_{lim} = 0.6$. Each algorithm is started 100 times for each set of initial parameters. The results are summarized in Fig. 21.

Observation 11. *The EP_2 algorithm has some problems with saddle crossing in t_{\max} (Fig. 21(a)). The percentage is $80\% < p_c \leq 100\%$ for $\alpha = 1.5$ and $\alpha = 1$. However, the efficiency of $EP_{0.5}$ is much lower than that of EP_1 .*

Observation 12. *When only the successful runs of the all algorithms are taken into consideration, \bar{t} is low for algorithms of low ζ – the EP_2 and $EP_{0.5}$ (Fig. 21(b)).*

4.3.3. Conclusions

It is not surprising that the effectiveness of saddle crossing is better for algorithms with a low value of α in the case

of low dimensions (Observation 8). Both optimum points in (18) are located on the same axis of the reference frame and the probability of $\mathbf{S}\alpha\mathbf{S}$ mutations in this direction increases with a decrease in α .

The range of mutations increases with n according to the *surrounding effect*. This fact allows $ESSS_2$ to become more effective with an increase in n for relatively low dimensions (Observation 10). This effect can be also visible for other algorithms from the $ESSS_\alpha$ class when smaller values of the scale parameter σ will be used.

The observed rapid decrease in saddle-crossing effectiveness for a higher dimension (Observation 9) is caused by two mechanisms. The proportion between the n -dimensional solid angle containing all directions of successful mutations (i.e., those which can produce offspring located in the global peak) and the full n -dimensional

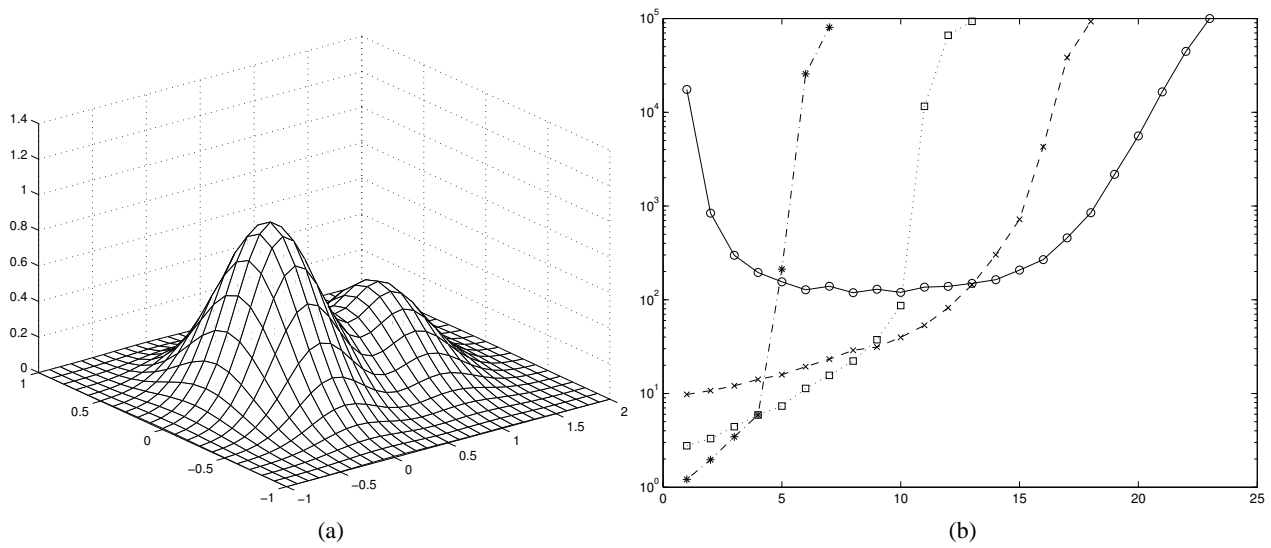


Fig. 20. Two-dimensional version of the sum of two Gaussian peaks (18) (a). The average number of iterations \bar{t} (taken over 500 algorithm runs) needed to cross the saddle vs. the dimension of the search space (b) (ES SS_2 – circles and solid line, ES $SS_{1.5}$ – crosses and dashed line, ES SS_1 – squares and dotted line, ES $SS_{0.5}$ – stars and dash-dotted line.)

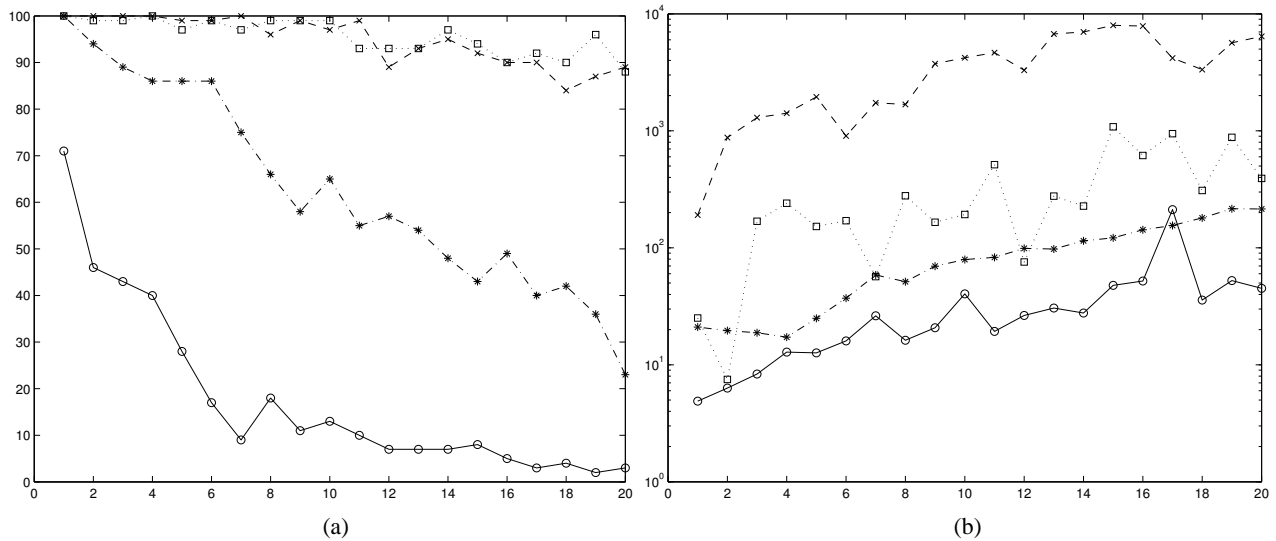


Fig. 21. Percentages ζ of the successful runs (a) and the mean number \bar{t} of iterations needed to cross the saddle taken over all successful runs (b) vs. the dimension of the search space, (EP $_2$ – circles and solid line, EP $_{1.5}$ – crosses and dashed line, EP $_1$ – squares and dotted line, EP $_{0.5}$ – stars and dash-dotted line.)

solid angle rapidly decreases when n increases. Then the probability of a successful mutation decreases, too. The second mechanism is connected with the *surrounding effect*. For relatively small dimensions this effect improves the saddle-crossing ability (Observation 10), but for n high enough the global optimum can be covered by the *dead surrounding* (see Section 4.2.3).

In the case of algorithms from the EP $_\alpha$ class, the problem of *dead surrounding* is overcome by the self-adaptation mechanism of the scale parameter. Another property becomes clear. Algorithms were started with

relatively small initial values of scale parameters (they are randomly selected from $\Omega_\sigma = \prod_{i=1}^n [0, 0.01]$). In the case of the EP $_2$ and EP $_{0.5}$ algorithms, either they cross a saddle in a relatively short time or they do not cross a saddle at all (Observations 11 and 12). If there are no successful mutations in the beginning of the algorithm run, then the self-adaptation and selection lead to decreasing values of σ , and the population is trapped around the local optimum. In the case of the EP $_2$ algorithm, the low number of successful mutations is caused by a low probability of macro-mutations. However, the mutation pdf in EP $_{0.5}$

possesses the heaviest tails in comparison with other examined algorithms, the $EP_{0.5}$ processing is characterized by the most rapid decrease of the scale parameter σ in time (Observation 7), and this is the cause of the untimely convergence to the local optimum.

4.4. Optimization of Selected Multimodal Functions

4.4.1. Problem Statement

The aim of this section is to analyze the convergence of the $ESSS_\alpha$ and EP_α algorithms to the global optimum of selected global optimization tasks. Three well-known multimodal global optimization benchmarks are chosen for this experiment:

(a) Ackley's function:

$$f_A(\mathbf{x}) = 20 + e - 20 \exp\left(-\frac{\|\mathbf{x}\|}{5n}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right), \quad (20)$$

(b) generalized Rastrigin function:

$$f_R(\mathbf{x}) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), \quad (21)$$

(c) generalized Griewank function:

$$f_G(\mathbf{x}) = \frac{\|\mathbf{x}\|^2}{4000} - \prod_{i=1}^n \frac{\cos(x_i)}{\sqrt{i}}. \quad (22)$$

Each of the above functions has its global optimum at $\mathbf{x}_{opt} = (0, \dots, 0)$ and $f_A(\mathbf{x}_{opt}) = f_R(\mathbf{x}_{opt}) = f_G(\mathbf{x}_{opt}) = 0$.

The stopping condition of all the algorithms considered in this experiment is

$$\min \{\|\mathbf{x}_i^t\| \mid i = 1, 2, \dots, \eta\} < 0.001, \quad (23)$$

or the maximum number of iterations t_{max} is exceeded.

4.4.2. Experiment and Results

In the first step of this experiment, four algorithms from the $ESSS_\alpha$ class are used: $ESSS_2$, $ESSS_{1.5}$, $ESSS_1$, and $ESSS_{0.5}$. All of them are applied in order to adapt to the 2D, 5D, 10D and 20D landscapes defined by the functions $f_A(\mathbf{x})$ (20), $f_R(\mathbf{x})$ (21), and $f_G(\mathbf{x})$ (22). The initial population is obtained by $\eta = 20$ mutations of the starting point $\mathbf{x}_0^0 = (30, 30, 0, \dots, 0)$. The scale parameter for all algorithms is the same: $\sigma = 0.05$. Each algorithm is started 50 times for each set of initial parameters. The

maximum number of epochs is set as $t_{max} = 100000$. Figures 22–24 present the convergence of the best element in the current population to the optimum for different α and n .

Observation 13. *In the case of all objective functions, the effectiveness of $ESSS_{0.5}$ is quite good for 2D. This algorithm crosses saddles faster than others; however, it does not localize the global optimum as precisely as $ESSS_{1.5}$ and $ESSS_1$. The effectiveness of $ESSS_{0.5}$ rapidly decreases when the landscape dimension n increases. The population stabilizes on a worse level of quality than the initial point \mathbf{x}_0^0 .*

Observation 14. *The behavior of the population in $ESSS_2$ is quite different than in $ESSS_{0.5}$. This population has serious problems while leaving the initial valley of the objective function. The saddle-crossing ability of $ESSS_2$ increases with an increase in n . However, it converges to the global optimum slower than $ESSS_{1.5}$ and $ESSS_1$, but localizes this point much better than others in the case of Ackley's and Rastrigin's 5D, 10D and 20D functions.*

Observation 15. *The $ESSS_{1.5}$ and $ESSS_1$ algorithms reconcile the properties described in Observations 13 and 14, and keep good efficiency in all cases. However, the downward tendency of their effectiveness appears for 20D landscapes.*

In the second step of this experiment, sixteen algorithms from the EP_α class are used ($\alpha = 2.0, 1.9, 1.8, \dots, 0.6, 0.5$). The initial population of $\eta = 20$ individuals is selected from $\Omega_x = \prod_{i=1}^2 [29, 31] \times \prod_{i=3}^n [-0.05, 0.05]$ and $\Omega_\sigma = \prod_{i=1}^n [0, 0.05]$. The number of sparring partners is $q = 5$. The maximum number of epochs is set as $t_{max} = 100000$. Each algorithm is started 50 times for each set of initial parameters. However, all algorithms are applied in order to adapt to the 2D, 5D, 10D and 20D landscapes defined by the functions $f_A(\mathbf{x})$ (20), $f_R(\mathbf{x})$ (21), and $f_G(\mathbf{x})$ (22). Only the results for Rastrigin's function $f_R(\mathbf{x})$ (21) are presented in Figs. 25–28, because the results obtained for all the functions considered demonstrate similar properties.

Observation 16. *The percentage of successful runs ζ decreases when the dimension of the search space increases. This fact is independent of α and the choice of the objective function.*

Observation 17. *The effectiveness of the classical evolutionary programming algorithm (EP_2) locates it among the worst algorithms out of all considered. The best results are obtained for $1 \leq \alpha \leq 1.5$ for each analysed dimension of the search space. For $\alpha < 1$, the percentage of successful runs decreases. The application of EP_α for $\alpha < 0.5$ does not lead to any successful run.*

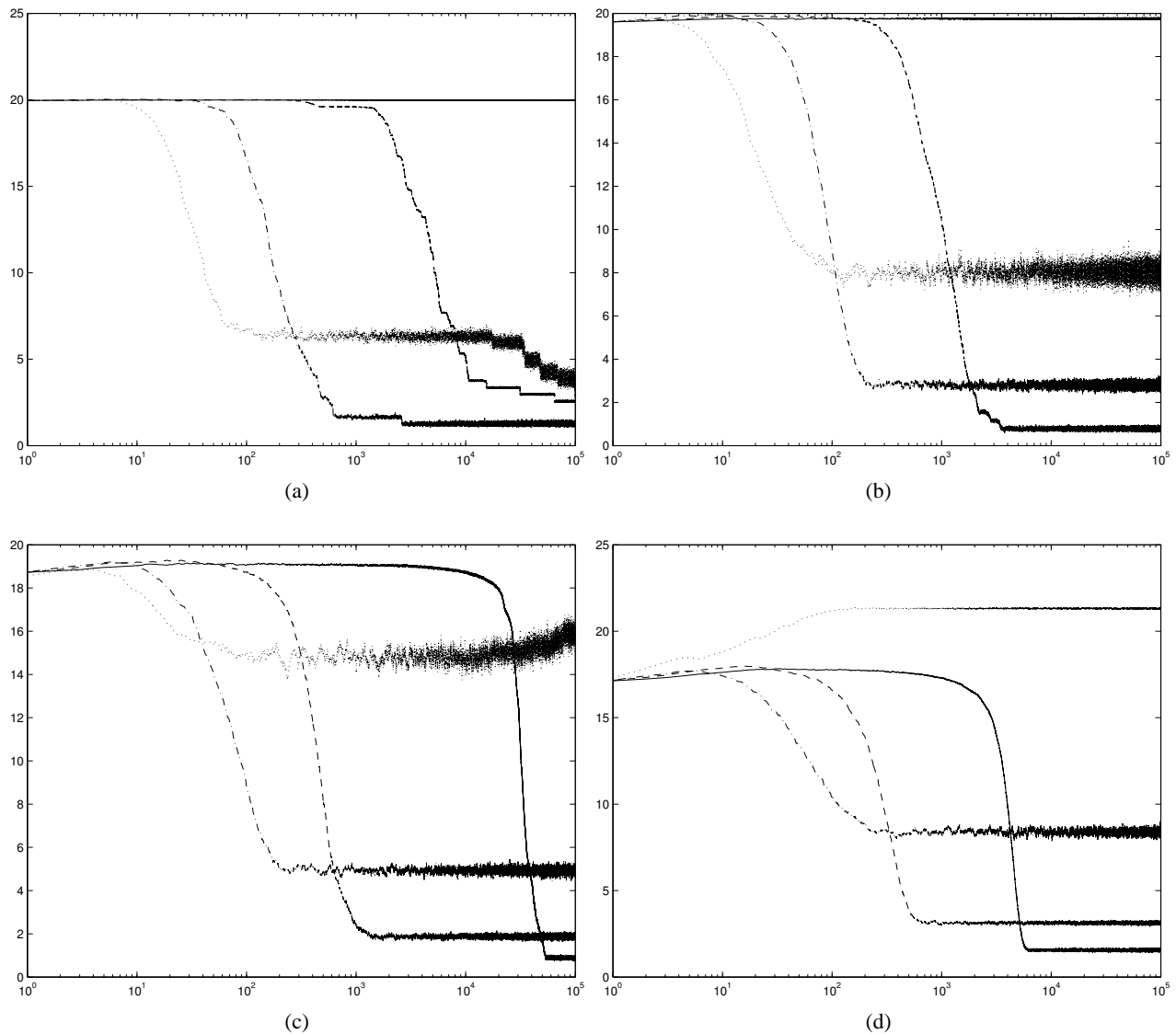


Fig. 22. Ackley's function $f_A(x)$ (20) optimized by the ESSS_α algorithm. The best value of $f_A(x)$ in the population vs. iterations (results averaged over 50 algorithms runs) in the 2D (a), 5D (b), 10D (c), and 20D (d) search spaces. ($\alpha = 2$ – solid line, $\alpha = 1.5$ – dashed line, $\alpha = 1$ – dash-dotted line, and $\alpha = 0.5$ – dotted line).

Observation 18. *This observation is similar to Observation 12, i.e., \bar{t} has low values for low ζ .*

4.4.3. Conclusions

Observations 13–15 show that the surrounding effect is one of the most important mechanisms influencing the efficiency of EAs in multimodal global optimization. The existence of the wide dead surrounding of $\text{ESSS}_{0.5}$ makes this algorithm unsuccessful. The same effect accelerates the saddle-crossing ability of ESSS_2 with an increase in n (Observation 14). All of the observations suggest that the range of the surrounding effect is very sensitive to changes in the landscape dimension n , the scale

parameter σ , and the index of stability α . Thus, the input parameter allocation process requires additional studies.

Like in previously considered experiments, the self-adaptation mechanism of EP_α algorithms allows us to eliminate dead surrounding. But the surrounding effect indirectly influences the effectiveness of algorithms of the EP_α class, especially in the case of low values of α (Observation 17). In these cases, the necessity of reducing dead surrounding results in a rapid decrease in σ to a very low values and the EP_α process is trapped around a local optimum. Thus, EP_α algorithms with low α either find the global optimum very quickly or cannot find it at all (Observation 18). In the case of high α , the effectiveness of EP_α is low because of the low probability of macro-mutations.

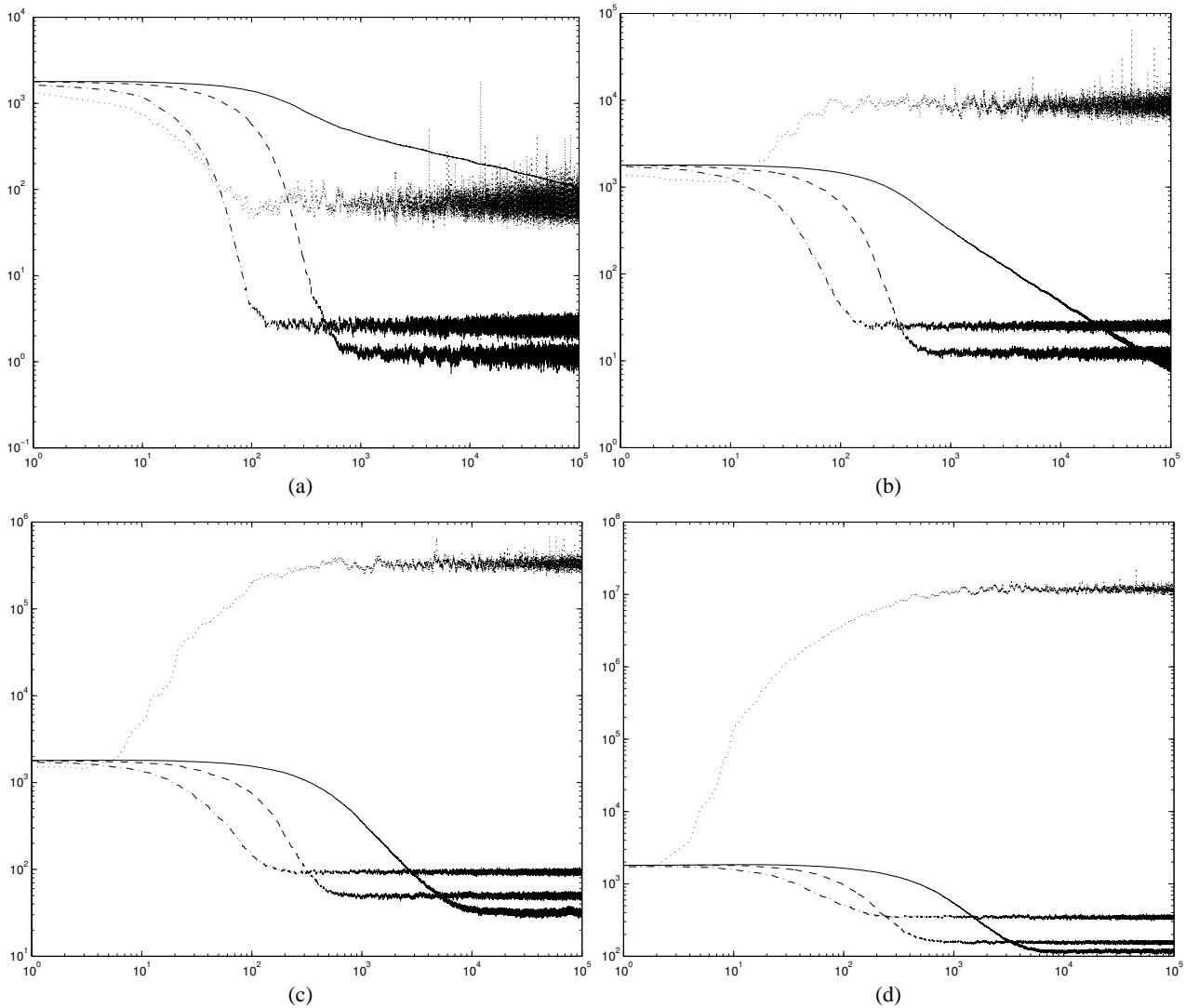


Fig. 23. Generalized Rastrigin function $f_R(\mathbf{x})$ (21) optimized by the ESSS_α algorithm. The best value of $f_R(\mathbf{x})$ in the population vs. iterations (results averaged over 50 algorithms runs) in the 2D (a), 5D (b), 10D (c), and 20D (d) search spaces. ($\alpha = 2$ – solid line, $\alpha = 1.5$ – dashed line, $\alpha = 1$ – dash-dotted line, and $\alpha = 0.5$ – dotted line).

4.5. Symmetry Problem

Let us consider the following series of three-dimensional fitness functions:

$$\Phi_l(\mathbf{x}) = \frac{1}{2} \exp(-5\|\mathbf{x}\|^2) + \exp(-5\|\mathbf{x} - \mathbf{m}_l\|^2), \quad l = 1, 2, 3, 4, \quad (24)$$

where

$$\begin{aligned} \mathbf{m}_1 &= (1, 0, 0, 0), \\ \mathbf{m}_2 &= (1/\sqrt{2}, 1/\sqrt{2}, 0, 0), \\ \mathbf{m}_3 &= (1/\sqrt{3}, 1/\sqrt{3}, 1/\sqrt{3}, 0), \\ \mathbf{m}_4 &= (1/2, 1/2, 1/2, 1/2) \end{aligned}$$

are global optimum locations. The lower local optimum is located in the center of the reference frame. The distances between both the local and global optima are the same in all Φ_l and equal to unity.

The goal is the same as in Section 4.3: to cross the saddle between both peaks when the initial population is chosen around the local optimum $\mathbf{x}_0^0 = (0, 0, 0)$. The searching process is stopped when the first element of the current population is located on the global peak higher than the lowest local optimum, i.e., $\exists k, \Phi_{sc}(\mathbf{x}_k) > \phi_{lim} > \Phi_{sc}(\mathbf{x}_0^0)$.

Two algorithms from the ESSS_α class are applied in this experiment: ESSS_1 and $\text{ESSS}_{0.5}$. The following input parameters (the same for all algorithms) are used: the population size $\eta = 20$, the scale parameter $\sigma = 0.02$,

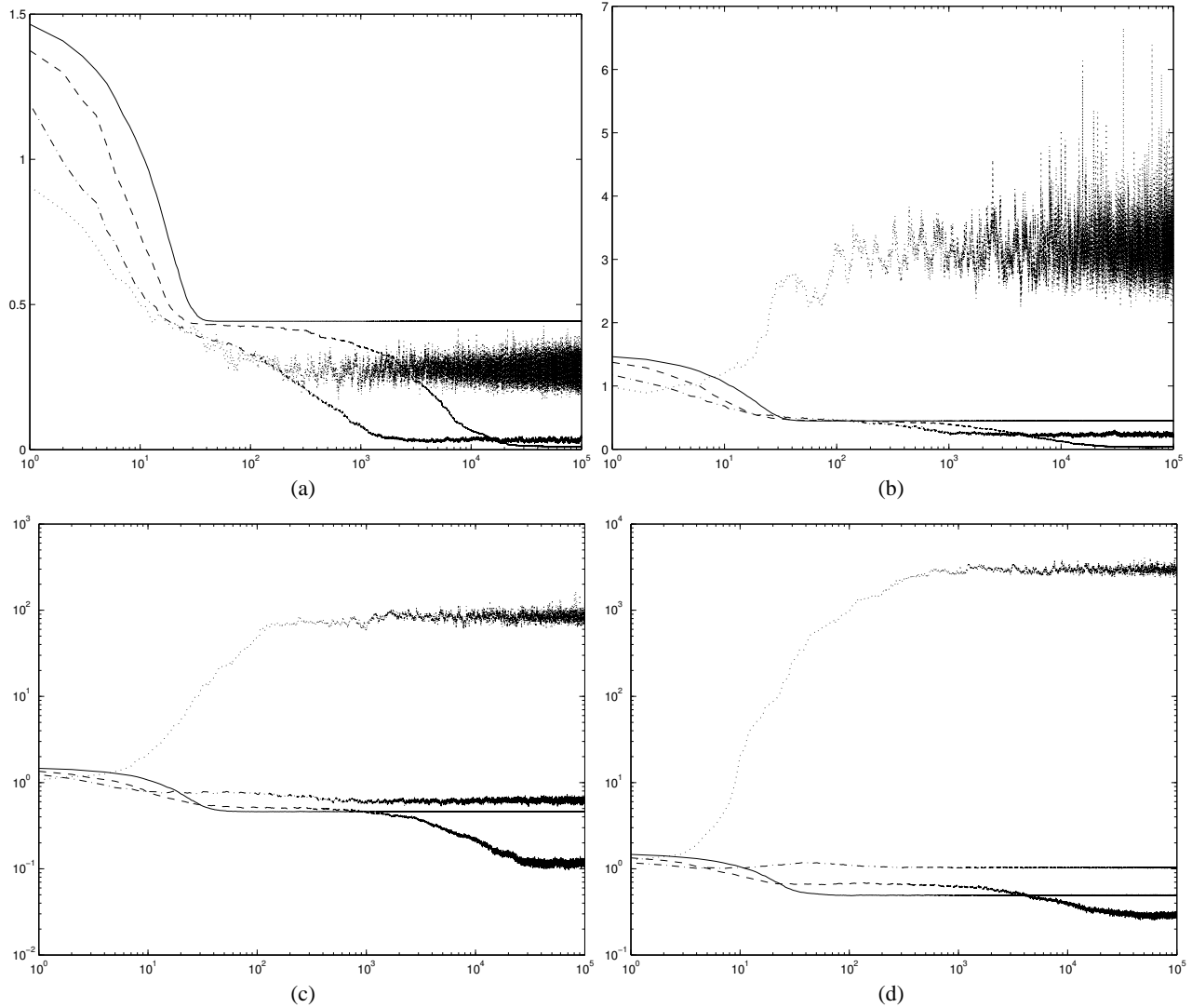


Fig. 24. Generalized Griewank function $f_G(x)$ (22) optimized by the $ESSS_\alpha$ algorithm. The best value of $f_G(x)$ in the population vs. iterations (results averaged over 50 algorithms runs) in the 2D (a), 5D (b), 10D (c), and 20D (d) search spaces. ($\alpha = 2$ – solid line, $\alpha = 1.5$ – dashed line, $\alpha = 1$ – dash-dotted line, and $\alpha = 0.5$ – dotted line).

the maximum number of iterations $t_{\max} = 10^5$, the limit fitness $\phi_{lim} = 0.6$. Each algorithm is started 300 times for each set of initial parameters. The results are presented in Fig. 29.

In the second step of this experiment, sixteen algorithms from the EP_α class ($\alpha = 2.0, 1.9, 1.8, \dots, 0.6, 0.5$) are used in order to allocate the global peak of Φ_l , $l = 1, 2, 3, 4$. The initial population of $\eta = 20$ individuals is selected from $\Omega_x = \prod_{i=1}^n [-0.05, 0.05]$ and $\Omega_\sigma = \prod_{i=1}^n [0, 0.05]$, the number of sparring partners being $q = 5$. The maximum number of epochs is $t_{\max} = 10000$, and the limit fitness is $\phi_{lim} = 0.6$. Each algorithm is started 100 times for each objective function. The results are presented in Figs. 30–33.

Observation 19. *It is easy to see that the efficiency of all algorithms considered strongly depends on the direction of locating the global peak.*

4.5.1. Conclusions

The last experiment reveals the influence of the selection of the reference frame on the global optimization effectiveness of evolutionary algorithms which use the $S\alpha S$ mutation with $\alpha \ll 2$.

5. Summary

The multi-dimensional Gaussian mutation is the most popular mutation technique in evolutionary algorithms

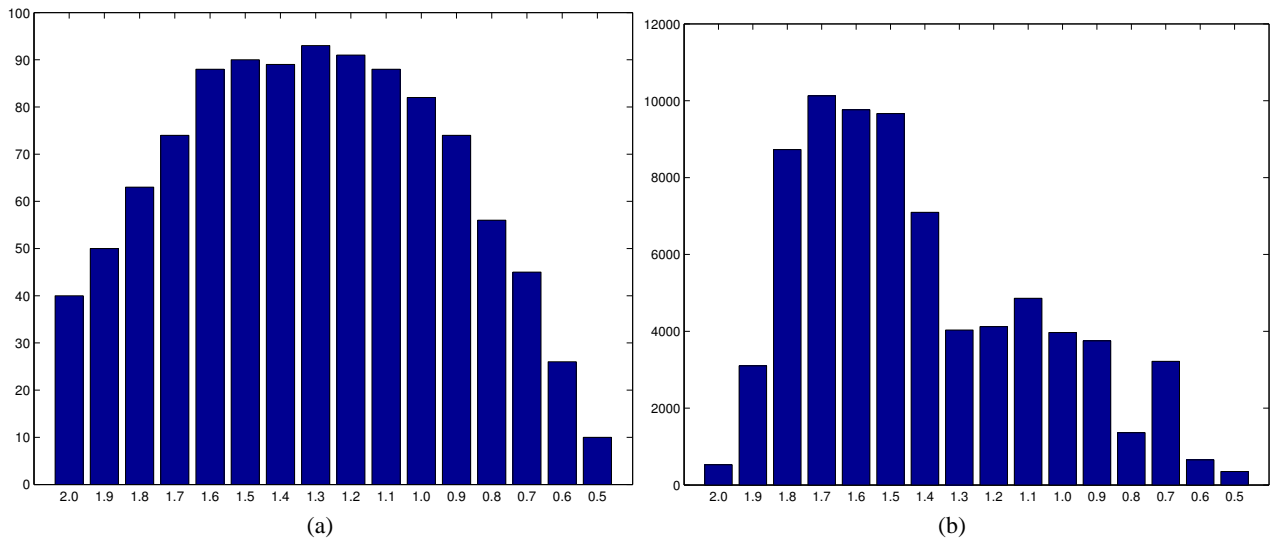


Fig. 25. Percentages ζ of the EP_α successful runs (a) and the mean number \bar{t} of iterations needed to allocate the global optimum taken over all successful runs (b) vs. the stability index α for the 2D Rastrigin function.

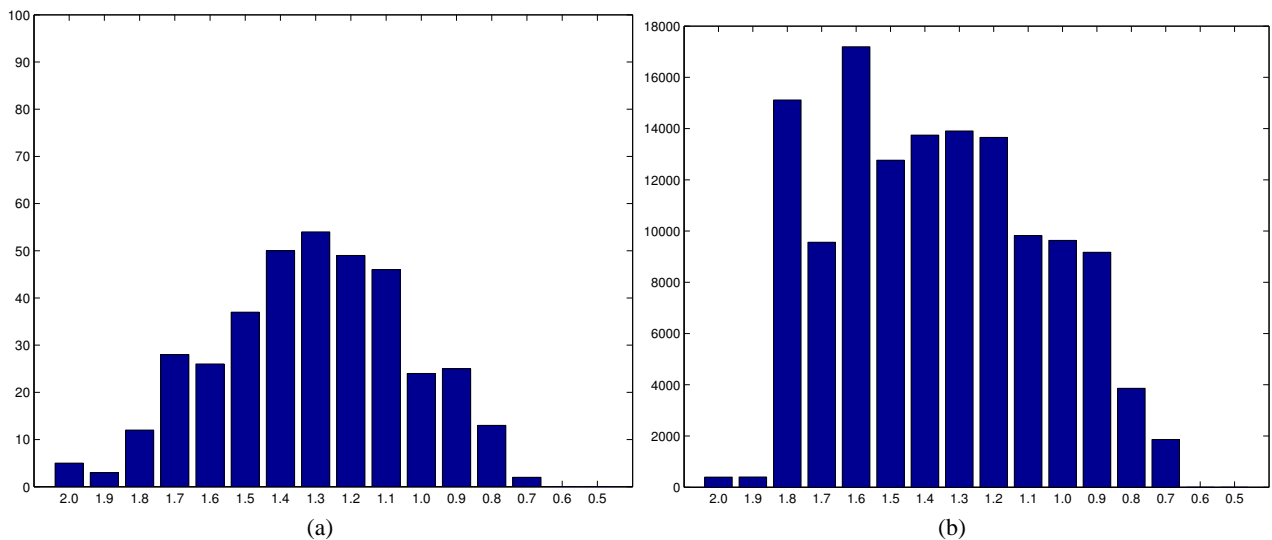


Fig. 26. Percentages ζ of the EP_α successful runs (a) and the mean number \bar{t} of iterations needed to allocate the global optimum taken over all successful runs (b) vs. the stability index α for the 5D Rastrigin function.

based on the floating point representation of individuals. In the case of an n -dimensional mutation, the most probable distance is equal to the norm of the standard deviation vector, which increases with the landscape dimension whenever the standard deviation of each entry is fitted (the *surrounding effect*). In recent years, the multi-dimensional Cauchy mutation has attracted a lot of attention. Evolutionary algorithms which use the Cauchy mutation seem to be more effective in comparison with algorithms with the Gaussian mutation in the case of most global optimization problems. But the multi-dimensional Cauchy density function obtained as a product of n independent one-dimensional Cauchy density functions is not isotropic (the *symmetry effect*). The convergence of the

density function to zero is different for different directions in the n -dimensional real space.

Both of the distributions mentioned above belong to the class of multivariate symmetric α -stable distributions. The success of the Cauchy mutation approach suggests that the application of other multivariate distributions from the $S_{\alpha S}$ class can be very attractive for evolutionary algorithms with the floating-point representation of individuals.

In this paper both of the properties under consideration, the *surrounding effect* and the *symmetry effect*, have been analysed using a set of simulating experiments. As examples of evolutionary algorithms, $(1+1)ES_\alpha$, $ESSS_\alpha$, and EP_α were used.

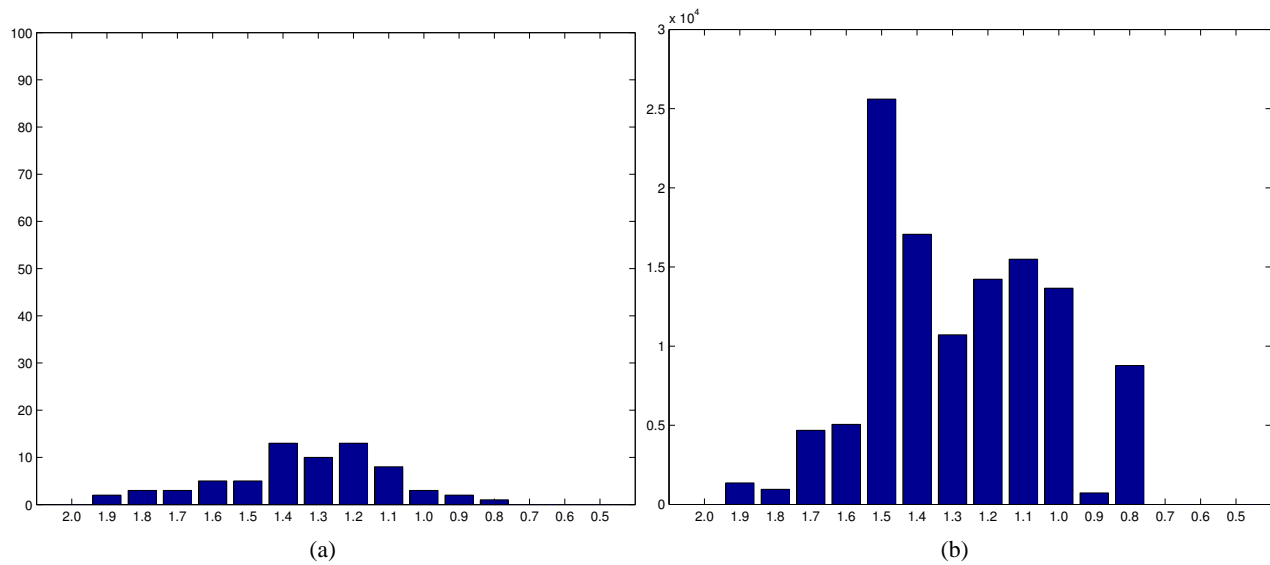


Fig. 27. Percentages ζ of the EP_α successful runs (a) and the mean number \bar{t} of iterations needed to allocate the global optimum taken over all successful runs (b) vs. the stability index α for the 10D Rastrigin function.

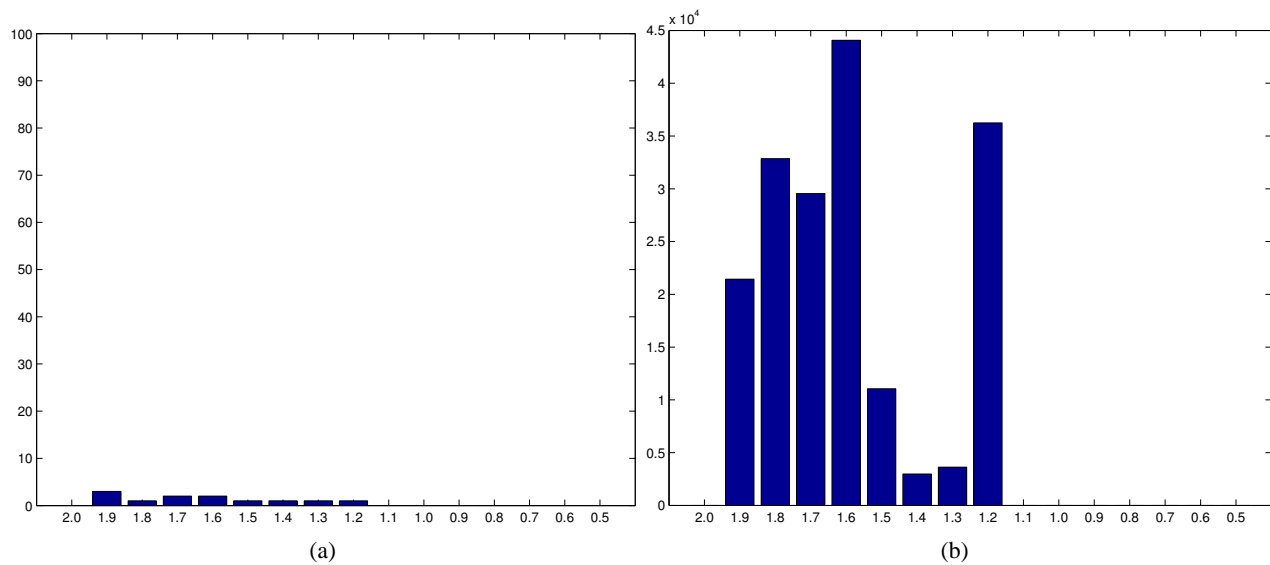


Fig. 28. Percentages ζ of the EP_α successful runs (a) and the mean number \bar{t} of iterations needed to allocate the global optimum taken over all successful runs (b) vs. the stability index α for the 20D Rastrigin function.

Convergence to the optimum of the sphere function was analysed in the first experiment, where four algorithms of the $(1+1)ES_\alpha$ class were tested. All algorithms were started from a set of initial points located in different directions from the optimum. The performed simulations prove the influence of the symmetry effect on the convergence rate of EAs based on $S\alpha S$ mutations with $\alpha < 2$. The strength of this influence is inversely proportional to the stability index α .

The search for the optima of unimodal functions was the goal of the second experiment, too. The emphasis was put on the dependence of the convergence of

selected $ESSS_\alpha$ algorithms on both the stability index α and the search space dimension n . As functions to be minimized, the sphere and Rosenbrock functions were chosen. However, the existence of the heavy tails of the $S\alpha S$ pdfs of low values of α allows the $ESSS_\alpha$ algorithm to converge quickly to the optimum neighborhood. The main mechanism influencing the algorithms' effectiveness was the surrounding effect. The probability of offspring allocation near the origin by the multivariate $S\alpha S$ mutation is low. Thus, the fluctuating population stabilizes at some distance from the optimum. This distance rapidly increases with the search space dimension.

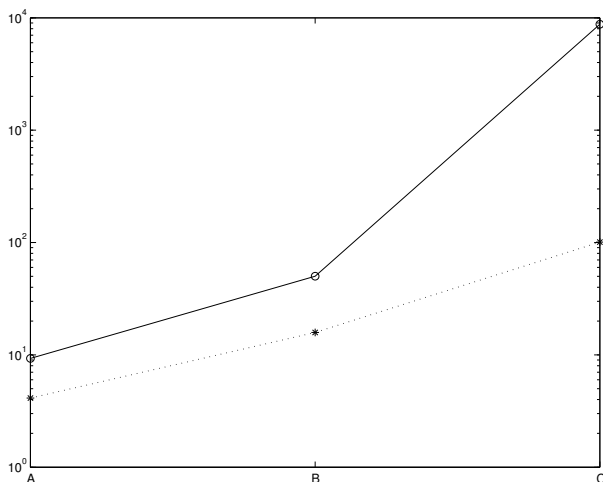


Fig. 29. Mean number of epochs needed to cross the saddle vs. the global optimum location (ESSS₁ – circles, ESSS_{0.5} – stars).

So, in the limit of the lowest α and high dimension the stabilized population locates its elements far from the optimum point. The surrounding effect is not directly observed when algorithms of the EP $_{\alpha}$ class are applied. It is reduced by the σ self-adaptation mechanism, which is implemented in these algorithms. The dead surrounding radius is proportional to σ . Thus, the self-adaptation mechanism controls both σ and the dead surrounding radius. This fact, among other things, causes slow convergence to the optimum of the EP $_{\alpha}$ algorithm with low values of α , for which wide dead surroundings have to be reduced by very low values of scale parameters σ .

The third experiment presents the influence of the landscape dimension on the exploration efficiency of the algorithms. The measure of the efficiency is the mean number of generations needed to cross a saddle between two Gaussian peaks. It is not surprising that, in the case of low dimensions, ESSS $_{\alpha}$ algorithms with a low value of α (with heavy tails of distributions used in mutations) are more effective than algorithms with a higher value of the stability index. The surrounding effect is the cause of two, seemingly contradictory, properties of ESSS $_{\alpha}$ algorithms. For low dimensions it accelerates the algorithms' saddle-crossing capability. But the efficiency of ESSS $_{\alpha}$ rapidly decreases when the landscape dimension increases. The Gaussian peaks of the fitness function become too narrow for these algorithms and offspring are located far from them. Because the surrounding effect appears strongly for low values of α , ESSS₂ with the standard Gaussian mutation becomes the most effective algorithm in the case of high search space dimensions. In the case of algorithms of the EP $_{\alpha}$ class, the worst results are obtained for the EP₂ and EP_{0.5} algorithms. However, the low efficiency of EP₂ algorithm is caused by a low probability of macro-

mutations, and the small number of successes of EP_{0.5} is caused by the rapid decrease in the scale parameter σ in time, and this algorithm untimely converges to the local optimum.

In the next experiment, all algorithms considered are applied to the global optimization problem of three well-known multivariate and multimodal optimization tasks: Ackley's, the generalized Rastrigin and generalized Griewank functions. The analysis of the obtained results proves that the surrounding effect is one of the most important mechanisms influencing the effectiveness of evolutionary algorithms, just like the symmetry effect, which is analysed in the last experiment. These simulations suggest that the most attractive algorithms for global optimization problem are EP $_{\alpha}$ for $1 \leq \alpha \leq 1.5$.

This work proves that S α S mutations are strongly influenced by two mechanisms: the surrounding effect and the heavy tails of their distributions. In the case of ESSS $_{\alpha}$ algorithms (without the self-adaptation of σ) the surrounding effect is a dominant mechanism affecting on the algorithm efficiency in a global optimization task, especially in the case of low values of α .

There is a strong dependence between the stability index α and the scale parameter σ in order to improve the effectiveness of EAs using S α S mutations. This relation, as well as the possibility of using the self-adaptation mechanism of α , need detailed studies, which are the goal for our further research.

References

- Bäck T. and Schwefel H.-P. (1993): *An overview of evolutionary computation*. — *Evol. Comput.*, Vol. 1, No. 1, pp. 1–23.
- Bäck T., Fogel D.B. and Michalewicz Z. (Eds.) (1997): *Handbook of Evolutionary Computation*. — Oxford: Oxford University Press, NY.
- Chambers J.M., Mallows C.L. and Stuck B.W. (1976): *A method for simulating stable random variables*. — *J. Amer. Statist. Assoc.*, Vol. 71, No. 354, pp. 340–344.
- Fang K.-T., Kotz S. and Ng S. (1990): *Symmetric Multivariate and Related Distributions*. — London: Chapman and Hall.
- Fogel L.J., Owens A.J. and Walsh A.J. (1966): *Artificial Intelligence through Simulated Evolution*. — New York: Wiley.
- Fogel D.B., Fogel L.J. and Atmar J.W. (1991): *Meta-evolutionary programming*. — *Proc. 25th Asilomar Conf. Signals, Systems & Computers*, San Jose, pp. 540–545.
- Fogel D.B. (1992): *An analysis of evolutionary programming*. — *Proc. 1st Annual Conf. Evolutionary Programming*, LA Jolla, CA: Evolutionary Programming Society, pp. 43–51.
- Fogel D.B. (1994): *An introduction to simulated evolutionary computation*. — *IEEE Trans. Neural Netw.*, Vol. 5, No. 1, pp. 3–14.

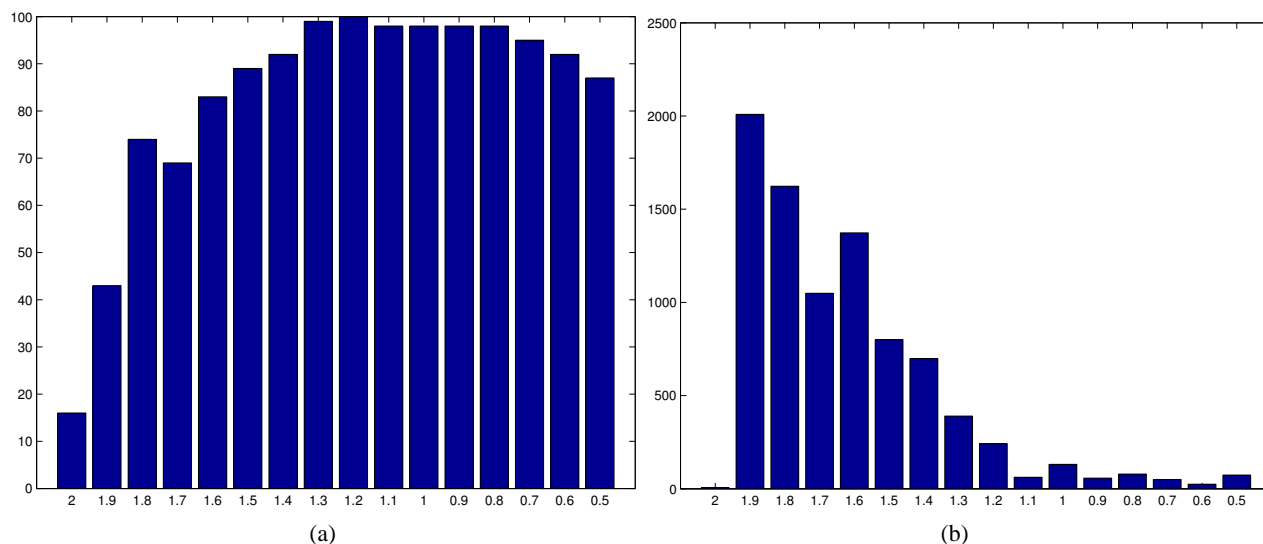


Fig. 30. Percentages ζ of the EP_α successful runs (a) and the mean number \bar{t} of iterations needed to allocate the global optimum taken over all successful runs (b) vs. the stability index α for $\Phi_1(x)$ (24).

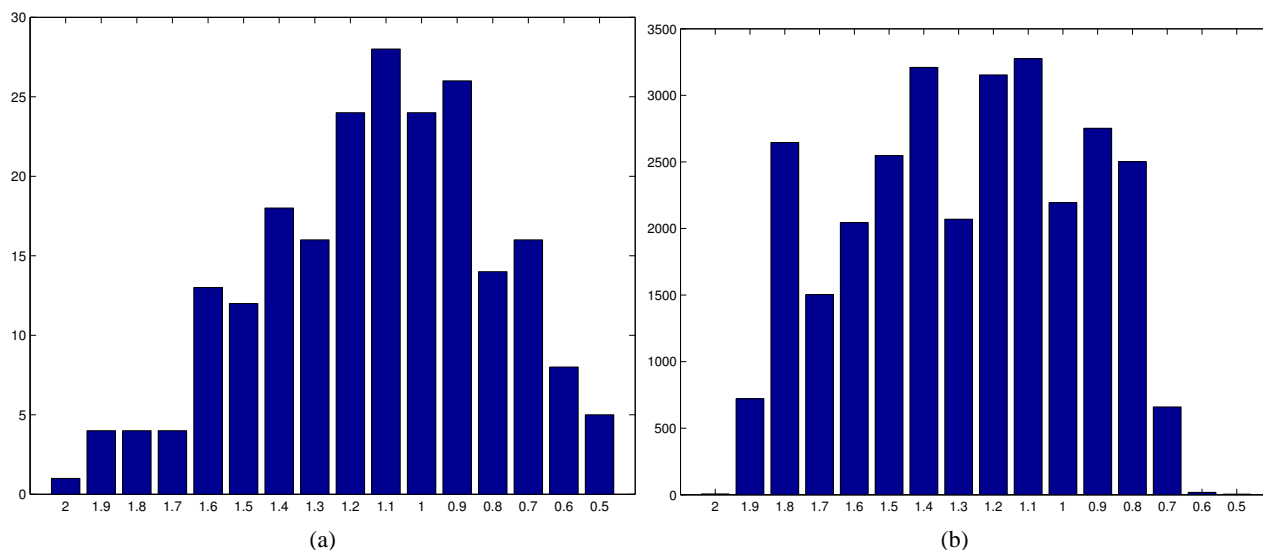


Fig. 31. Percentages ζ of the EP_α successful runs (a) and the mean number \bar{t} of iterations needed to allocate the global optimum taken over all successful runs (b) vs. the stability index α for $\Phi_2(x)$ (24).

Galar R. (1985): *Handicapped individua in evolutionary processes.* — Biol. Cybern., Vol. 51, pp. 1–9.

Galar R. (1989): *Evolutionary search with soft selection.* — Biol. Cybern., Vol. 60, pp. 357–364.

Gutowski M. (2001): *Lévy flights as an underlying mechanism for a global optimization algorithm.* — Proc. 5th Conf. Evolutionary Algorithms and Global Optimization, Jaszczybia Góra, Poland, pp. 79–86.

Kanter M. (1975): *Stable densities with change of scale and total variation inequalities.* — Ann. Probab., Vol. 3, No. 4, pp. 687–707.

Kappler C. (1996): *Are evolutionary algorithms improved by large mutation,* In: Problem Solving from Nature (PPSN IV (H.-M. Voigt, W. Ebeling, I. Rechenberg and H.-P. Schwefel, Eds.). — Berlin: Springer, pp. 388–397.

Lévy C. (1925): *Calcul des Probabilités.* — Paris: Gauthier Villars.

Michalewicz Z. (1996): *Genetic Algorithms + Data Structures = Evolution Programs.* — Berlin: Springer.

Nolan J.P. (2003): *Stable Distributions. Models for Heavy Tailed Data.* — Berlin: Springer.

Obuchowicz A. (2001a): *On the true nature of the multi-dimensional Gaussian mutation.* — In: Artificial Neural Networks and Genetic Algorithms (V. Kurkova, N.C. Steel, R. Neruda and M. Karny, Eds.). — Vienna: Springer, pp.248–251.

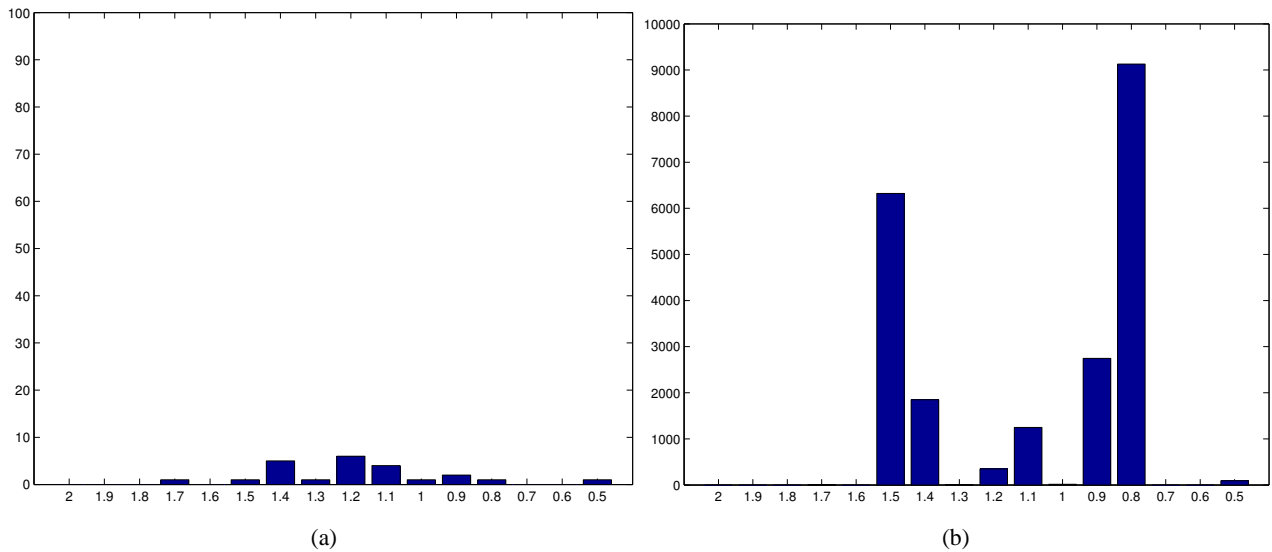


Fig. 32. Percentages ζ of the EP $_{\alpha}$ successful runs (a) and the mean number \bar{t} of iterations needed to allocate the global optimum taken over all successful runs (b) vs. the stability index α for $\Phi_3(x)$ (24).

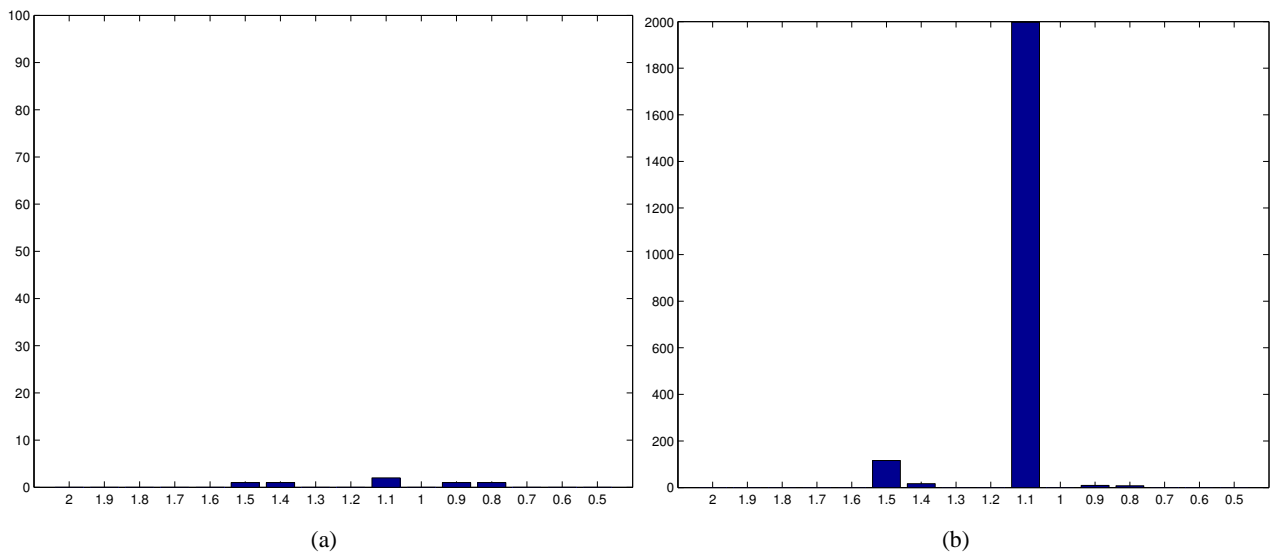


Fig. 33. Percentages ζ of the EP $_{\alpha}$ successful runs (a) and the mean number \bar{t} of iterations needed to allocate the global optimum taken over all successful runs (b) vs. the stability index α for $\Phi_4(x)$ (24).

Obuchowicz A. (2001b): *Mutli-dimensional Gaussian and Cauchy mutations*, In: Intelligent Information Systems (M. Kłopotek, M. Michalewicz, and S.T. Wierchoń, Eds.). — Heidelberg: Physica-Verlag, pp. 133–142.

Obuchowicz A. (2003a): *Population in an ecological niche: Simulation of natural exploration*. — Bull. Polish Acad. Sci., Tech. Sci., Vol. 51, No. 1, pp. 59–104.

Obuchowicz A. (2003b): *Multidimensional mutations in evolutionary algorithms based on real-valued representation*. — Int. J. Syst. Sci., Vol. 34, No. 7, pp. 469–483.

Obuchowicz A. (2003c): *Evolutionary Algorithms in Global Optimization and Dynamic System Diagnosis*. — Zielona Góra: Lubuskie Scientific Society.

Rechenberg I. (1965): *Cybernetic solution path of an experimental problem*. — Roy. Aircr. Establ., Libr. Transl. 1122, Farnborough, Hants., UK.

Rudolph G. (1997): *Local convergence rates of simple evolutionary algorithms with Cauchy mutations*. — IEEE Trans. Evolut. Comput., Vol. 1, No. 4, pp. 249–258.

Schwefel H.-P. (1981): *Numerical Optimization of Computer Models*. — Chichester: Wiley.

Samorodnitsky G. and Taqqu M.S. (1994): *Stable Non-Gaussian Random Processes*. — New York: Chapman & Hall.

Shu A. and Hartley R. (1987): *Fast simulated annealing*. — Phys. Lett. A, Vol. 122, Nos. 3/4, pp. 605–614.

- Weron R. (1996): *Correction to: On the Chambers-Mallows-Stuck method for simulating skewed stable random variables.* — Res. Rep., Wrocław University of Technology, Poland.
- Weron R. (2001): *Lévy-stable distributions revisited: tail index > 2 does not exclude the Lévy-stable regime.* — Int. J. Modern Phys. C, Vol. 12, No. 2, pp. 209–223.
- Yao X. and Liu Y. (1996): *Fast evolutionary programming.* In: *Evolutionary Programming V: Proc. 5th Annual Conference on Evolutionary Programming* (L.J. Fogel, P.J. Angeline, and T. Bäck, Eds.). — Cambridge, MA: MIT Press, pp. 419–429.
- Yao X. and Liu Y. (1997): *Fast evolutionary strategies.* — Contr. Cybern., Vol. 26, No. 3, pp. 467–496.
- Yao X. and Liu Y. (1999): *Evolutionary programming made faster.* — IEEE Trans. Evolut. Comput., Vol. 3, No. 2, pp. 82–102.
- Zolotariev A. (1986): *One-Dimensional Stable Distributions.* — Providence: American Mathematical Society.