Письма в ЭЧАЯ. 2007. Т. 4, № 1(137). С. 119–131

FINDING TWO-DIMENSIONAL PEAKS

Z. K. Silagadze

Budker Institute of Nuclear Physics, Novosibirsk, Russia

Two-dimensional generalization of the original peak finding algorithm suggested earlier is given. The ideology of the algorithm emerged from the well-known quantum mechanical tunneling property which enables small bodies to penetrate through narrow potential barriers. We merge this «quantum» ideology with the philosophy of Particle Swarm Optimization to get the global optimization algorithm which can be called Quantum Swarm Optimization. The functionality of the newborn algorithm is tested on some benchmark optimization problems.

Представлен предложенный ранее алгоритм нахождения двумерного обобщения исходного пика. Идея алгоритма возникла из хорошо известного квантово-механического свойства туннелирования, которое позволяет небольшим частицам проникать через узкие потенциальные барьеры. Мы дополняем эту «квантовую» идеологию философией оптимизации роя частиц для получения глобального алгоритма оптимизации, который можно назвать оптимизацией квантового роя. Функциональность нового алгоритма проверяется на некоторых контрольных проблемах оптимизации.

PACS: 02.60.Pn

INTRODUCTION

Some time ago we suggested a new algorithm for automatic photopeak location in gamma-ray spectra from semiconductor and scintillator detectors [1]. The algorithm was inspired by quantum mechanical property of small balls to penetrate through narrow barriers and to find their way down to the potential wall bottom even in the case of irregular potential shape.

In one dimensional case the idea was realized by means of finite Markov chain and its invariant distribution [1]. States of this Markov chain correspond to channels of the original histogram. The only nonzero transition probabilities are those which connect a given state to its closest left and right neighbor states. Therefore, the transition probability matrix for our Markov chain has the form:

$$P = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & \cdot & \cdot & \cdot \\ P_{21} & 0 & P_{23} & 0 & 0 & \cdot & \cdot & \cdot \\ 0 & P_{32} & 0 & P_{34} & 0 & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & 1 & 0 \end{pmatrix}.$$

As for the transition probabilities, the following expressions were used:

$$P_{i,i\pm1} = \frac{Q_{i,i\pm1}}{Q_{i,i-1} + Q_{i,i+1}} \tag{1}$$

with

$$Q_{i,i\pm1} = \sum_{k=1}^{m} \exp\left[\frac{N_{i\pm k} - N_i}{\sqrt{N_{i\pm k} + N_i}}\right]. \tag{2}$$

Number m is a parameter of the model which mimics the (inverse) mass of the quantum ball, and therefore allows us to govern its penetrating ability.

The invariant distribution for the above-described Markov chain can be given by a simple analytic formula [2]:

$$u_2 = \frac{P_{12}}{P_{21}}u_1\ ,\ u_3 = \frac{P_{12}P_{23}}{P_{32}P_{21}}u_1\ ,\ \ldots\ ,\ u_n = \frac{P_{12}P_{23}\cdots P_{n-1,n}}{P_{n,n-1}P_{n-1,n-2}\cdots P_{21}}u_1,$$

where u_1 is defined from the normalization condition

$$\sum_{i=1}^{n} u_i = 1.$$

Local maximums in the original spectrum are translated into the very sharp peaks in the invariant distribution, and therefore their location is facilitated.

The algorithm proved helpful in uniformity studies of NaJ(Tl) crystals for the SND detector [3]. Another application of this «peak amplifier», to refine the amplitude fit method in ATLAS B_s-mixing studies, was described in [4]. In this paper we will try to extend the method also in the two-dimensional case.

1. TWO-DIMENSIONAL GENERALIZATION

The following two-dimensional generalization seems straightforward. For two-dimensional $n \times n$ histograms the corresponding Markov chain states will also form a two-dimensional array (i, j). Let $u_{ij}^{(k)}$ be a probability for the state (i, j) to be occupied after k-steps of the Markov process. Then

$$u_{lm}^{(k+1)} = \sum_{i,j=1}^{n} P_{ij;lm}u_{ij}^{(k)},$$

where $P_{ij;lm}$ is a transition probability from the state (i, j) to the state (l, m). We will assume that the only nonzero transition probabilities are those which connect a given state to its closest left, right, up or down neighbor states. Then the generalization of Eqs. (1) and (2) is almost obvious. Namely, for the transition probabilities we will take

$$\begin{aligned} P_{ij;i,j\pm1} &= \frac{Q_{ij;i,j\pm1}}{Q_{ij;i,j-1} + Q_{ij;i,j+1} + Q_{ij;i-1,j} + Q_{ij;i+1,j}}, \\ P_{ij;i\pm1,j} &= \frac{Q_{ij;i\pm1,j}}{Q_{ij;i,j-1} + Q_{ij;i,j+1} + Q_{ij;i-1,j} + Q_{ij;i+1,j}}, \end{aligned} \tag{3}$$

with

$$
\begin{aligned}
Q_{ij;i,j\pm1} &= \sum_{k=1}^{m}\sum_{l=-k}^{k} \exp\left[\frac{N_{i+l,j\pm k}-N_{ij}}{\sqrt{N_{i+l,j\pm k}+N_{ij}}}\right], \\
Q_{ij;i\pm1,j} &= \sum_{k=1}^{m}\sum_{l=-k}^{k} \exp\left[\frac{N_{i\pm k,j+l}-N_{ij}}{\sqrt{N_{i\pm k,j+l}+N_{ij}}}\right].
\end{aligned} \tag{4}
$$

We are interested in the invariant distribution u_{ij} for this Markov chain, such that

$$
\sum_{i,j=1}^{n} P_{ij;lm}u_{ij} = u_{lm}.
$$

Unfortunately, unlike one-dimensional case, this invariant distribution can not be given by a simple analytic formula. But there is a way out: having at hand the transition probabilities $P_{ij;lm}$, we can simulate the corresponding Markov process starting with some initial distribution $u_{ij}^{(0)}$. Then, after a sufficiently large number of iterations we will end with almost invariant distribution irrespective to the initial choice of $u_{ij}^{(0)}$. For example, in the role of $u_{ij}^{(0)}$ one can take the uniform distribution:

$$
u_{ij}^{(0)} = \frac{1}{n^2}.
$$

For practical realization of the algorithm, it is desirable to have the precise meaning of the words «sufficiently large number of iterations». In our first tests the following stopping criterion was used: one stops at k-th iteration if the averaged relative difference between $u_{ij}^{(k)}$ and $u_{ij}^{(k-1)}$ probability distributions is less than the desired accuracy ϵ:

$$
\sum_{u_{ij}^{(k)}\neq 0} 2\,\frac{|u_{ij}^{(k)}-u_{ij}^{(k-1)}|}{u_{ij}^{(k)}+u_{ij}^{(k-1)}}\,u_{ij}^{(k)} < \epsilon\,. \tag{5}
$$

The performance of the algorithm is illustrated by Fig. 1 for a 100×100 histogram representing three overlapping Gaussians with different widths. As expected, it works much like its one-dimensional cousin: the invariant probability distribution shows sharp peaks at locations where the initial data have quite broad local maximums. Note that in this concrete example one iteration variability $\epsilon = 10^{-3}$ was reached after 258 iterations for $m = 3$ and after 113 iterations for $m = 30$.

Convergence to the invariant distribution can be slow. In the example given above by Fig. 1 the convergence is indeed slow for small penetrating abilities. If we continue iterations for $m = 3$ further, the side peaks will slowly decay in favor of the main peak corresponding to the global maximum. In the case of $m = 3$ it takes too much iterations to reach the invariant distribution. However, as Fig. 1 indicates, the remarkable property to develop sharp peaks at locations of local maximums of the initial histogram is already revealed by $u_{ij}^{(k)}$ when a number of iterations k is in the order of 300.

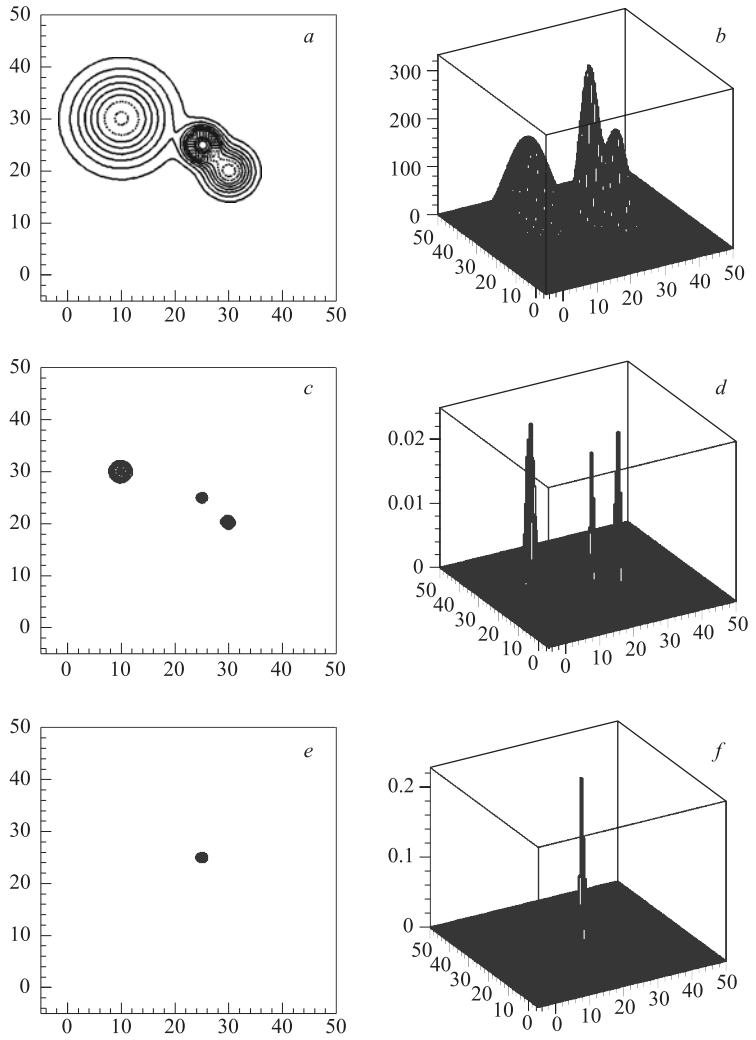

Fig. 1. *a*, *b*) The initial data in the contour and lego formats, respectively; *c*, *d*) the corresponding probability distribution after 258 iterations for the penetrating ability $m = 3$; *e*, *f*) the invariant distribution for the penetrating ability $m = 30$

One can make the algorithm to emphasize minimums, not maximums, by just reversing signs in the exponents:

$$\exp\left[\frac{N_{lm}-N_{ij}}{\sqrt{N_{lm}+N_{ij}}}\right] \longrightarrow \exp\left[-\frac{N_{lm}-N_{ij}}{\sqrt{N_{lm}+N_{ij}}}\right].$$

This is illustrated by Fig. 2. Here the initial histogram is generated by using a variant of the

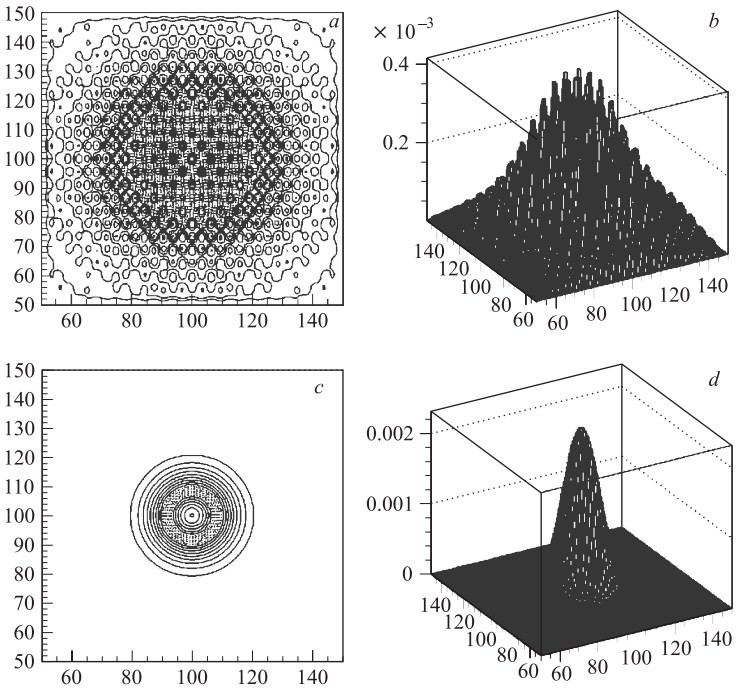

Fig. 2. The probability distribution for the Griewank function. *a*, *b*) histograms for $m = 3$; *c*, *d*) histograms for $m = 30$

Griewank function [5]

$$F(x, y) = \frac{(x - 100)^2 + (y - 100)^2}{4000} - \cos(x - 100)\cos\frac{y - 100}{\sqrt{2}} + 1. \qquad (6)$$

This function has the global minimum at a point $x = 100,\ y = 100$ and in the histogramed interval $50 \leqslant x \leqslant 150,\ 50 \leqslant y \leqslant 150$ exhibits nearly thousand local minimums. Many of them are still visible in the probability distribution for penetrating ability $m = 3$. But for $m = 30$ only one peak, corresponding to the global minimum, remains.

2. QUANTUM SWARM OPTIMIZATION

The above discussion was focused on two-dimensional histograms, while, in practice, a more common problem is to find global optimums of nonlinear functions. The algorithm in the form discussed so far is not suitable for this latter problem. However, it is possible to merge this ideology with that of Particle Swarm Optimization [6–8] to get a workable tool.

The Particle Swarm Optimization was inspired by intriguing ability of bird flocks to find spots with food, even though the birds in the flock had no previous knowledge of their location and appearance. «This algorithm belongs ideologically to that philosophical school that allows wisdom to emerge rather than trying to impose it, that emulates nature rather than trying to control it, and that seeks to make things simpler rather than more complex» [6].

This charming philosophy is indeed very attractive. So, we attempted to develop Quantum Swarm Optimization — when each particle in the swarm mimics the quantum behavior.

The algorithm that emerged goes as follows:

• Initialize a swarm of n_p particles at random positions in the search space $x_{\min} \leqslant x \leqslant x_{\max}$, $y_{\min} \leqslant y \leqslant y_{\max}$.

• Find a particle i_b in the swarm with the best position (x_b, y_b), such that the function $F(x, y)$ under investigation has the most optimal value for the swarm at (x_b, y_b).

• For each particle in the swarm find the distance from the best position $d = \sqrt{(x - x_b)^2 + (y - y_b)^2}$. For the best particle instead take the maximal value of these distances from the previous iteration (or, for the first iteration take $d = \sqrt{(x_{\max} - x_{\min})^2 + (y_{\max} - y_{\min})^2}$).

• Generate a random number r uniformly distributed in the interval $0 \leqslant r \leqslant 1$ and find the random step $h = rd$.

• Check for a better optimum the closest left, right, up and down neighbor states with the step h. If the result is positive, change i_b and (x_b, y_b) respectively. Otherwise,

• Move the particle to left, right, up or down by the step h according to the corresponding probabilities of such jumps:

$$p_L = \frac{q_L}{q_L + q_R + q_U + q_D}, \quad p_R = \frac{q_R}{q_L + q_R + q_U + q_D},$$
$$p_U = \frac{q_U}{q_L + q_R + q_U + q_D}, \quad p_D = \frac{q_D}{q_L + q_R + q_U + q_D}, \tag{7}$$

where

$$q_L = \sum_{y'=y_u,y,y_d} \exp\left(I_s \frac{F(x_d, y') - F(x, y)}{h}\right),$$
$$q_R = \sum_{y'=y_u,y,y_d} \exp\left(I_s \frac{F(x_u, y') - F(x, y)}{h}\right),$$
$$q_D = \sum_{x'=x_u,x,x_d} \exp\left(I_s \frac{F(x', y_d) - F(x, y)}{h}\right), \tag{8}$$
$$q_U = \sum_{x'=x_u,x,x_d} \exp\left(I_s \frac{F(x', y_u) - F(x, y)}{h}\right),$$

and

$$x_u = \min(x + h, x_{\max}), \quad x_d = \max(x - h, x_{\min}),$$
$$y_u = \min(y + h, y_{\max}), \quad y_d = \max(y - h, y_{\min}). \tag{9}$$

At last, $I_s = 1$, if optimization means to find the global maximum, and $I_s = -1$, if the global minimum is being searched.

• Do not stick at walls. If the particle is at the boundary of the search space, it jumps away from the wall with the probability equaled one (that is, the probabilities of other three jumps are set to zero).

• Check whether the new position of the particle leads to the better optimum. If yes, change i_b and (x_b, y_b) accordingly.

• Do not move the best particle if not profitable.

• When all particles from the swarm make their jumps, the iteration is finished. Repeat it at a prescribed time or until some other stopping criteria are met.

To test the algorithm performance, we tried it on some benchmark optimization test functions. For each test function and for each number of iterations one thousand independent numerical experiments were performed and the success rate of the algorithm was calculated. The criterion of success was the following:

$$|x_f - x_m| \leqslant \begin{cases} 10^{-3}\,|x_m|, & \text{if } |x_m| > 10^{-3}, \\ 10^{-3}, & \text{if } |x_m| \leqslant 10^{-3}, \end{cases}$$
$$|y_f - y_m| \leqslant \begin{cases} 10^{-3}\,|y_m|, & \text{if } |y_m| > 10^{-3}, \\ 10^{-3}, & \text{if } |y_m| \leqslant 10^{-3}, \end{cases} \tag{10}$$

where (x_m, y_m) is the true position of the global optimum and (x_f, y_f) is the position found by the algorithm. The results are given in the Table below. The test functions themselves are defined in the Appendix. Here we give only some comments on the algorithm performance.

Success rate of the algorithm in percentages for various test functions and for various numbers of iterations. Swarm size $n_p = 20$

Function name	Iterations							
	50	100	200	300	400	500	600	700
Chichinadze	35.5	97	100	100	100	100	100	100
Schwefel	99.4	99.5	99.8	99.3	99.2	99.8	100	99.6
Ackley	100	100	100	100	100	100	100	100
Matyas	88.9	100	100	100	100	100	100	100
Booth	100	100	100	100	100	100	100	100
Easom	93.6	100	100	100	100	100	100	100
Levy5	98.4	99.5	99.4	99.3	99	99	99.1	99.5
Goldstein–Price	100	100	100	100	100	100	100	100
Griewank	76.3	99.7	100	100	100	100	100	100
Rastrigin	100	100	99.8	99.9	100	99.9	99.9	100
Rosenbrock	43.6	90.4	99.8	100	100	100	100	100
Leon	13.8	52.1	82	91.6	97.6	99.1	99.6	99.8
Giunta	100	100	100	100	100	100	100	100
Beale	99.7	100	100	100	100	100	100	100
Bukin2	61.8	84.4	93.8	97.8	98.6	99.3	99.7	99.8
Bukin4	99.6	100	100	100	100	100	100	100
Bukin6	0.2	0.1	0	0.2	0	0.1	0.2	0.1
Styblinski–Tang	100	100	100	100	100	100	100	100
Zettl	100	100	100	100	100	100	100	100
Three Hump Camel	100	100	100	100	100	100	100	100
Schaffer	8.2	34.7	60.7	71.2	77.8	78.9	80.4	83.9
Levy13	100	100	100	100	100	100	100	100
McCormic	100	100	100	100	100	100	100	100

For some test problems, such as Chichinadze, Ackley, Matyas, Booth, Easom, Goldstein–Price, Griewank, Giunta, Beale, Bukin4, Styblinski–Tang, Zettl, Levy13, McCormic and Three Hump Camel Back, the algorithm is triumphant.

Matyas problem seems simple, because the function is only quadratic. However, it is very flat near the line $x = y$ and this leads to problems for many global optimization algorithms.

Easom function is an unimodal test function which is expected to be hard for any stochastic algorithms, because vicinity of its global minimum has a small area compared to the search space. Surprisingly, our algorithm performs quite well for this function and one needs only about 100 iterations to find the needle of the global minimum in a haystack of the search space.

Schwefel function is deceptive enough to cause search algorithms to converge in the wrong direction. This happens because the global minimum is geometrically distant from the next best local minima. In some small fraction of events our algorithm is also prone to converge in the wrong direction and, in these cases, the performance does not seem to be improved by further increasing the number of iterations. But the success rate is quite high. Therefore, in this case, it is more sensible to have two or more independent tries of the algorithm with rather a small number of iterations each.

Rastrigin function is a multimodal test function which has plenty of hills and valleys. Our algorithm performs even better for this function, but the success is not universal either.

Rosenbrock function is, on the contrary, unimodal. Its minimum is situated in a banana shaped valley with a flat bottom and is not easy to find. The algorithm needs more than 200 iterations to be successful in this case. Leon function is of the similar nature, with even more flat bottom, and the convergence in this case is correspondingly more slow.

Griewank, Levy5 and Levy13 are multimodal test functions. They are considered to be difficult for local optimizers because of the very rugged landscapes and a very large number of local optima. For example, Levy5 has 760 local minima in the search domain but only one global minimum, and Levy13 has 900 local minima. Test results reveal a small probability that our algorithm becomes stuck in one of the local minima for the Levy5 function.

Giunta function simulates the effects of numerical noise by means of a high frequency, low amplitude sine wave, added to the main part of the function. The algorithm is successful for this function.

Convergence of the algorithm is rather slow for Bukin2 function, and especially for the Schaffer function. This latter problem is hard because the highly variable data surface features many circular local optima, and, unfortunately, our algorithm often becomes stuck in the optima nearest to the global one.

At last, the algorithm fails completely for the Bukin6 function. This function has a long narrow valley which is readily identified by the algorithm. But the function values differ very little along the valley. Besides, the surface is non-smooth in the valley with numerous pitfalls. This problem seems hopeless for any stochastic algorithm based heavily on random walks, because one has to chance upon the very vicinity of the global optimum to be successful. The non-stochastic component of our algorithm (calculation of jump probabilities to mimic the quantum tunneling) turns out to be of little use for this particular problem.

CONCLUDING REMARKS

The Quantum Swarm Optimization algorithm presented above emerged while trying to generalize in the two-dimensional case a «quantum mechanical» algorithm for automatic location of photopeaks in the one dimensional histogram [1].

«Everything has been said before, but since nobody listens we have to keep going back and beginning all over again» [9]. After this investigation had been almost finished, we

discovered the paper [10] by Xie, Zhang and Yang with the similar idea to use the simulation of particle-wave duality in optimization problems. However, their realization of the idea is quite different.

Even earlier, Levy and Montalvo used the tunneling method for global optimization [11], but without referring to quantum behavior. Their method consisted in a transformation of the objective function, once a local minimum has been reached, which destroys this local minimum and allows one to tunnel classically to another valley.

We also found that the similar ideology to mimic nature's quantum behavior in optimization problems emerged in quantum chemistry and led to such algorithms as quantum annealing [12] and Quantum Path Minimization [13].

Nevertheless, the Quantum Swarm Optimization is conceptually rather different from these developments. We hope it is simple and effective enough to find an ecological niche in a variety of global optimization algorithms.

APPENDIX

Here we collect the test functions definitions, locations of their optimums and the boundaries of the search space. The majority of them was taken from [14–16], but we also provide the original reference when known.

Chichinadze function [16, 17]

$$F(x,y) = x^2 - 12x + 11 + 10\cos\frac{\pi}{2}x + 8\sin(5\pi x) - \frac{1}{\sqrt{5}}\exp\left(-\frac{(y-0.5)^2}{2}\right),$$

$$-30 \leqslant x, y \leqslant 30, \quad F_{\min}(x,y) = F(5.90133, 0.5) = -43.3159.$$

Schwefel function [18]

$$F(x,y) = -x\sin\sqrt{|x|} - y\sin\sqrt{|y|},$$

$$-500 \leqslant x, y \leqslant 500, \quad F_{\min}(x,y) = F(420.9687, 420.9687) = -837.9658.$$

Ackley function [19]

$$F(x,y) = 20[1 - \exp(-0.2\sqrt{0.5(x^2+y^2)})] - \exp(0.5[\cos(2\pi x) + \cos(2\pi y)]) + e,$$

$$-35 \leqslant x, y \leqslant 35, \quad F_{\min}(x,y) = F(0,0) = 0.$$

Matyas function [15]

$$F(x,y) = 0.26(x^2+y^2) - 0.48xy,$$

$$-10 \leqslant x, y \leqslant 10, \quad F_{\min}(x,y) = F(0,0) = 0.$$

Booth function [16]

$$F(x,y) = (x+2y-7)^2 + (2x+y-5)^2,$$

$$-10 \leqslant x, y \leqslant 10, \quad F_{\min}(x,y) = F(1,3) = 0.$$

Easom function [20]

$$F(x,y) = -\cos x \cos y \exp\left[-(x-\pi)^2 - (y-\pi)^2\right],$$

$$-100 \leqslant x, y \leqslant 100, \quad F_{\min}(x,y) = F(\pi,\pi) = -1.$$

Levy5 function [15]

$$F(x,y) = \sum_{i=1}^{5} i\cos\left[(i-1)x + i\right] \sum_{j=1}^{5} j\cos\left[(j+1)y + j\right] + \\ + (x+1.42513)^2 + (y+0.80032)^2,$$

$$-100 \leqslant x, y \leqslant 100, \quad F_{\min}(x,y) = F(-1.30685, -1.424845) = -176.1375.$$

Goldstein–Price function [15]

$$F(x,y) = \left[1 + (x+y+1)^2(19 - 14x + 3x^2 - 14y + 6xy + 3y^2)\right] \times \\ \times \left[30 + (2x-3y)^2(18 - 32x + 12x^2 + 48y - 36xy + 27y^2)\right],$$

$$-2 \leqslant x, y \leqslant 2, \quad F_{\min}(x,y) = F(0,-1) = 3.$$

Griewank function [5, 15]

$$F(x,y) = \frac{x^2+y^2}{200} - \cos x \cos\frac{y}{\sqrt{2}} + 1,$$

$$-100 \leqslant x, y \leqslant 100, \quad F_{\min}(x,y) = F(0,0) = 0.$$

Rastrigin function [21]

$$F(x,y) = x^2 + y^2 - 10\cos(2\pi x) - 10\cos(2\pi y) + 20,$$

$$-5.12 \leqslant x, y \leqslant 5.12, \quad F_{\min}(x,y) = F(0,0) = 0.$$

Rosenbrock function [15]

$$F(x,y) = 100(y - x^2)^2 + (1-x)^2,$$

$$-1.2 \leqslant x, y \leqslant 1.2, \quad F_{\min}(x,y) = F(1,1) = 0.$$

Leon function [22]

$$F(x,y) = 100(y - x^3)^2 + (1-x)^2,$$

$$-1.2 \leqslant x, y \leqslant 1.2, \quad F_{\min}(x,y) = F(1,1) = 0.$$

Giunta function [23]

$$F(x,y) = \sin\left(\frac{16}{15}x - 1\right) + \sin^2\left(\frac{16}{15}x - 1\right) + \frac{1}{50}\sin\left[4\left(\frac{16}{15}x - 1\right)\right] + \\ + \sin\left(\frac{16}{15}y - 1\right) + \sin^2\left(\frac{16}{15}y - 1\right) + \frac{1}{50}\sin\left[4\left(\frac{16}{15}y - 1\right)\right] + 0.6,$$

$$-1 \leqslant x, y \leqslant 1, \quad F_{\min}(x, y) = F(0.45834282, 0.45834282) = 0.0602472184.$$

Beale function [15]

$$F(x, y) = (1.5 - x + xy)^2 + (2.25 - x + xy^2)^2 + (2.625 - x + xy^3)^2,$$
$$-4.5 \leqslant x, y \leqslant 4.5, \quad F_{\min}(x, y) = F(3, 0) = 0.$$

Bukin2 function [24]

$$F(x, y) = 100(y - 0.01x^2 + 1) + 0.01(x + 10)^2,$$
$$-15 \leqslant x \leqslant -5, \ -3 \leqslant y \leqslant 3, \quad F_{\min}(x, y) = F(-10, 0) = 0.$$

Bukin4 function [24]

$$F(x, y) = 100y^2 + 0.01|x + 10|,$$
$$-15 \leqslant x \leqslant -5, \ -3 \leqslant y \leqslant 3, \quad F_{\min}(x, y) = F(-10, 0) = 0.$$

Bukin6 function [24]

$$F(x, y) = 100\sqrt{|y - 0.01x^2|} + 0.01|x + 10|,$$
$$-15 \leqslant x \leqslant -5, \ -3 \leqslant y \leqslant 3, \quad F_{\min}(x, y) = F(-10, 1) = 0.$$

Styblinski–Tang function [25]

$$F(x, y) = \frac{1}{2}\left[x^4 - 16x^2 + 5x + y^4 - 16y^2 + 5y\right],$$
$$-5 \leqslant x, y \leqslant 15, \quad F_{\min}(x, y) = F(-2.903534, -2.903534) = -78.332.$$

Zettl function [22]

$$F(x, y) = (x^2 + y^2 - 2x)^2 + 0.25x,$$
$$-5 \leqslant x, y \leqslant 5, \quad F_{\min}(x, y) = F(-0.0299, 0) = -0.003791.$$

Three Hump Camel back function [14]

$$F(x, y) = 2x^2 - 1.05x^4 + \frac{x^6}{6} + xy + y^2,$$
$$-5 \leqslant x, y \leqslant 5, \quad F_{\min}(x, y) = F(0, 0) = 0.$$

Schaffer function [26]

$$F(x, y) = 0.5 + \frac{\sin\sqrt{x^2 + y^2} - 0.5}{[1 + 0.001(x^2 + y^2)]^2},$$
$$-100 \leqslant x, y \leqslant 100, \quad F_{\min}(x, y) = F(0, 0) = 0.$$

Levy13 function [14]

$$F(x, y) = \sin^2(3\pi x) + (x - 1)^2\left[1 + \sin^2(3\pi y)\right] + (y - 1)^2\left[1 + \sin^2(2\pi y)\right],$$
$$-10 \leqslant x, y \leqslant 10, \quad F_{\min}(x, y) = F(1, 1) = 0.$$

McCormic function [27]

$$F(x, y) = \sin(x + y) + (x - y)^2 - 1.5x + 2.5y + 1,$$
$$-1.5 \leqslant x \leqslant 4, \ -3 \leqslant x \leqslant 4 \quad F_{\min}(x, y) = F(-0.54719, -1.54719) = -1.9133.$$

Acknowledgments. Support from the INTAS grant No. 00–00679 is acknowledged.

REFERENCES

1. *Silagadze Z. K.* A New Algorithm for Automatic Photopeak Searches // Nucl. Instr. Meth. A. 1996. V. 376. P. 451–454.

2. *Feller W.* An Introduction in Probability Theory and Its Applications. N. Y.: Wiley, 1966. V. 1.

3. *Achasov M. N. et al.* Medium Energy Calorimetry at SND: Techniques and Performances on Physics // Proc. of «Calorimetry in High Energy Physics», Lisbon, 1999. Singapore: World Sci., 2000. P. 105–120.

4. *Bannikov A. V., Chelkov G. A., Silagadze Z. K.* $B_s^0 \to D_s^- a_1^+$ ($D_s^- \to \phi\pi^-, D_s^- \to K^{*0}K^-$) Decay Channel in the ATLAS B_s^0 Mixing Studies. JINR Preprint E1-98-29. Dubna, 1998.

5. *Griewank A. O.* Generalized Descent for Global Optimization // J. Optim. Theory Appl. 1981. V. 34. P. 11–39.

6. *Kennedy J., Eberhart R.* Particle Swarm Optimization // Proc. of IEEE Conf. on Neural Networks, Perth, 1995. P. 1942–1948.

7. *Kennedy J., Eberhart R. C., Shi Y.* Swarm Intelligence. Morgan Kaufmann Publishers. San Francisco, 2001.

8. *Parsopoulos K. E., Vrahatis M. N.* Recent Approaches to Global Optimization Problems through Particle Swarm Optimization // Natural Computing. 2002. V. 1. P. 235–306.

9. *Gide A.* The Tractate of Narcissus (Theory of Symbol) / Transl. by A. Mangravite // The Book of Masks; Atlas Arkhive Two: French Symbolist & Decadent Writing of the 1890s. London: Atlas Press, 1994.

10. *Xie X. F., Zhang W. J., Yang Z. L.* Solving Numerical Optimization Problems by Simulating Particle-Wave Duality and Social Information Sharing // Intern. Conf. on Artificial Intelligence, Las Vegas, USA, 2002. P. 1163–1169.

11. *Levy A. V., Montalvo A.* The Tunneling Method for Global Optimization // SIAM J. Sci. Stat. Comp. 1985. V. 6. P. 15–29.

12. For a review and references see, for example, *Kadowaki T.* Study of Optimization Problems by Quantum Annealing. Ph.D. thesis. Department of Physics, Tokyo Institute of Technology, 1998.

13. *Liu P., Berne B. J.* Quantum Path Minimization: An Efficient Method for Global Optimization // J. Chem. Phys. 2003. V. 118. P. 2999–3005.

14. *Jansson C., Knüppel O.* A Global Minimization Method: The Multi-dimensional Case. Technical Report 92-1. TU Hamburg; Harburg, 1992.

15. *Jansson C., Knüppel O.* Numerical Results for a Self-validating Global Optimization Method. Technical Report 94-1. TU Hamburg; Harburg, 1994.

16. *Van Iwaarden R. J.* An Improved Unconstrained Global Optimization Algorithm. Ph.D. thesis. Univ. of Colorado at Denver. Denver, 1996.

17. *Chichinadze V.* The ψ-Transform for Solving Linear and Nonlinear Programming Problems // Automata. 1969. V. 5. P. 347–355.

18. *Schwefel H.-P.* Numerical Optimization of Computer Models. Chichester: Wiley & Sons, 1981.

19. *Ackley D. H.* A Connectionist Machine for Genetic Hillclimbing. Boston: Kluwer Acad. Publ., 1987.

20. *Easom E. E.* A Survey of Global Optimization Techniques. M. Eng. thesis. Univ. Louisville, KY, Louisville, 1990.

21. *Voigt H.-M., Born J., Santibanez-Koref I.* A Multivalued Evolutionary Algorithm. Technical Report TR-92-038. Intern. Comp. Sci. Inst. Berkeley, CA, 1992.

22. *Nagendra S.* Catalogue of Test Problems for Optimization Algorithm Verification. Technical Report 97-CRD-110. General Electric Company, 1997.

23. *Giunta A. A.* Aircraft Multidisciplinary Design Optimization Using Design of Experiments Theory and Response Surface Modeling Methods. MAD Center Report 97-05-01. Virginia Polytechn. Inst. & State Univ. Blacksburg, VA, 1997.

24. *Bukin A. D.* New Minimization Strategy for Non-smooth Functions. Budker Inst. of Nucl. Phys. Preprint BUDKER-INP-1997-79. Novosibirsk, 1997.

25. *Styblinski M., Tang T.* Experiments in Nonconvex Optimization: Stochastic Approximation with Function Smoothing and Simulated Annealing // Neural Networks. 1990. V. 3. P. 467–483.

26. *Whitley D. et al.* Evaluating Evolutionary Algorithms // Artificial Intelligence, 1996. V. 85. P. 245–276.

27. *Madsen K., Zilinskas J.* Testing Branch-and-Bound Methods for Global Optimization. IMM Technical Report 05. Techn. Univ. of Denmark. 2000.

Received on March 16, 2006.