Stochastic optimization algorithm with probability vector in mathematical function minimization and travelling salesman problem

JAN POHL, VÁCLAV JIRSÍK, PETR HONZÍK
Department of Control and Instrumentation
Brno University of Technology
Kolejní 2906/4, 612 00 Brno
CZECH REPUBLIC
xpohlj00@stud.feec.vutbr.cz, jirsik@feec.vutbr.cz, honzikp@feec.vutbr.cz

Abstract: In the paper there is introduced the newly developed optimization method the Stochastic Optimization Algorithm with Probability Vector (PSV). It is related to Stochastic Learning Algorithm with Probability Vector for artificial neural networks. Both algorithms are inspired by stochastic iterated function system SIFS for generating the statistically self similar fractals. The PSV is gradient method where the direction of individual future movement from the population is based stochastically. PSV was tested on mathematical function minimization and on the travelling sales man problem. The influence of the quantity of individuals upon the best achieved fitness function was also tested on the mathematical functions minimization.

Key Words: Stochastic, Optimization Algorithm, PSV, SIFS, Random Walk, Travelling Salesman Problem

1 Introduction

There are many optimization algorithms, e.g. gradient, stochastic, inspired by collective behaviour of biological individuals etc [1, 3, 4, 9 and 10]. The PSV algorithm that is introduced in this paper is a modified version of the stochastic learning algorithm [5, 6]. The algorithm is inspired by the stochastic iterated function system SIFS for generating the statistically self similar fractals [11, 12]. It is based on the group of separate individuals, which do not share information with each other. It is based on the similar principle as the stochastic hill climbing [2, 7] or random walk (RW).

2 Algorithm PSV

PSV algorithm uses individuals from the group in the similar way as in the stochastic hill climbing. There is defined a vector of transformations (1).

$$T = (t_1, t_2, \cdots, t_n) \tag{1}$$

Each of them can modify any parameter of an individual in a specific way. In every step the transformation is chosen randomly but with regard to the probability in the probability vector (2) that has to satisfy the conditions (3) and (4). This step is similar to the algorithm random walk (RW) [7].

$$P = (p_1, p_2, \cdots, p_n) \tag{2}$$

$$\sum_{i=1}^{n} |p_i| = 1 \tag{3}$$

$$p_i \in (-1,1) \wedge p_i \neq 0 \tag{4}$$

Each value in the probability vector can be interpreted as the probability (5) and the direction of movement (6).

$$|p_i| - probability \tag{5}$$

$$sign(p_i) - direction \tag{6}$$

If the chosen transformation is accepted, then the probability of this transformation increases. Transformation is accepted if the fitness function gives better result after its applying. On the other hand if the transformation gives worse results its probability decreases. Let's have an individual $W_k \in \boldsymbol{W}$. This individual consists of a set of parameters (7).

$$W_k = (w_1, w_2, \cdots, w_n) \tag{7}$$

At first the fitness function of the selected individual is evaluated. Secondly the roulette wheel selection is used to choose and apply the transformation from (1).

$$individual_{new} = t_i(individual) \quad (8)$$

Finally the fitness function of the new individual is evaluated. In the case the new individual has higher fitness function than the previous the related probability p_i is increased.

$$p_{i\_new} = p_i + sign(p_i)\alpha \quad (9)$$

Symbol α is coefficient used for increasing the probability. If the new individual has lower fitness function than the previous the related probability p_i is decreased.

$$p_{i\_new} = p_i + sign(p_i)(-\beta) \quad (10)$$

Symbol β is the coefficient used for decreasing the probability. Example of the basic transformations is (11).

$$w_{i\_new} = t_i(w_i) = w_i\left(1 + \frac{p_i\lambda}{w_i}\right) \quad (11)$$

Transformation (11) transforms parameter w_i of the individual W_k in specific direction and the coefficient λ is a learning (for artificial neural networks) or moving rate.

3 Function minimization and travelling salesman problem

We successfully tested the PSV method as the learning algorithm for artificial neural networks [5, 6]. In this paper was introduced the modified PSV algorithm. It is used to solve the travelling salesman problem and to find extremes of selected mathematical testing functions.

We used the following test functions:

1. 1st De Jong
2. Rodenbrock's saddle
3. 3rd De Jong
4. 4rd De Jong
5. Rastrigin's function
6. Schwefel's function
7. Griewangk's function
8. Sine envelope wave function
9. Stretched V sine wave function
10. Test function Ackley
11. Ackley's function
12. Michalewicz's function
13. Masters's cosine wave function

Optimization of the travelling salesman problem was tested on the circle and randomly deployed cities.

3.1 Function minimization

The examples of the tested functions are in Tab. 1 together with the best results found after 100 000 steps. Every function was tested with 50 individuals and different sets of the variables with parameters α, β, λ.

f.num.	var. x_1	var. x_2	fit.fun.
1	0.0009	0.0025	⇒0
2	0.9974	0.9955	⇒0
3	-0.0010	-0.0013	0.0024
4	-0.0005	-0.0013	⇒0
5	0.0107	-0.0054	0.0286
6	420.975	420.957	-837.965
7	-0.0916	0.0784	0.0057
8	-1.1773	-1.6985	-1.4915
9	-0.0013	0.0032	0.0668
10	-1.5148	-0.7686	-4.5888
11	0.0028	-0.0008	0.00537
12	2.2031	1.5708	-1.8013
13	0.0731	-0.0261	-0.9591

Tab. 1 Function results

On the figures in paragraph 3.1.1 - 3.1.8 there are examples of the some function from the Tab.1. Markers (ellipse and cross) represent estimated position of global minimum in the specific interval. The red dot represents the individual with the best fitness function.

3.1.1 Rastrigin's function

Tested with parameters α = 0.3, β= 0.4, λ= 0.1.

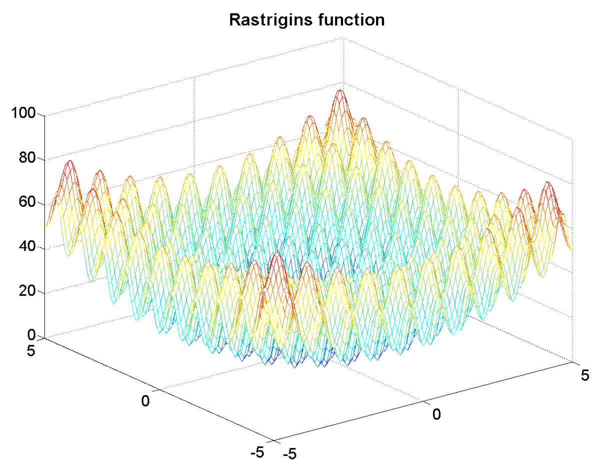

Fig.1 Rastrigin's function

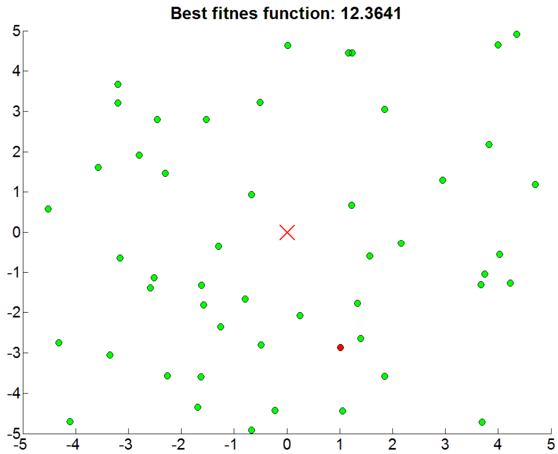

Fig.2 Rastrigin’s function starting position

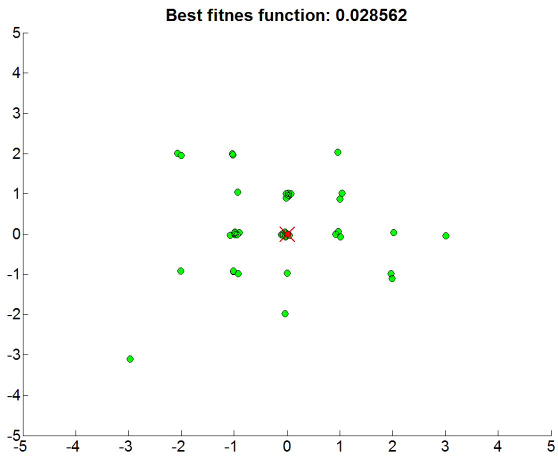

Fig.3 Rastrigin’s function position after 100 000 steps

3.1.2 Schwefel’s function

Tested with parameters α = 0.3, β= 0.4, λ= 0.1.

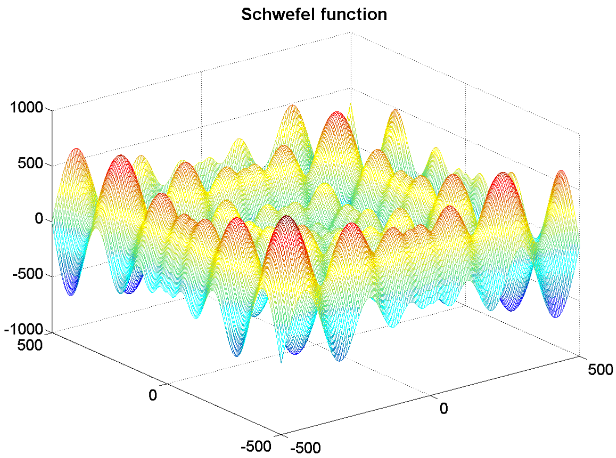

Fig.4 Schwefel function

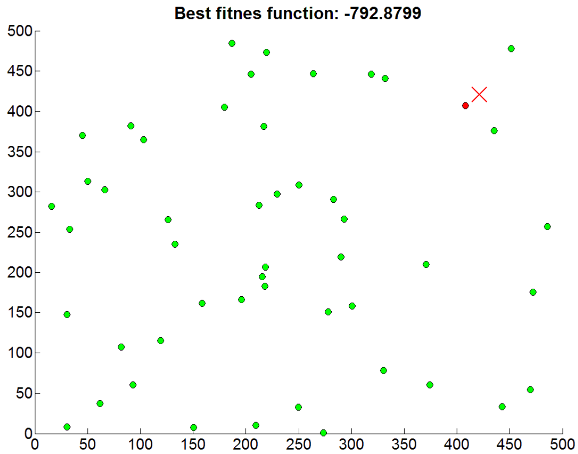

Fig.5 Schwefel function starting position

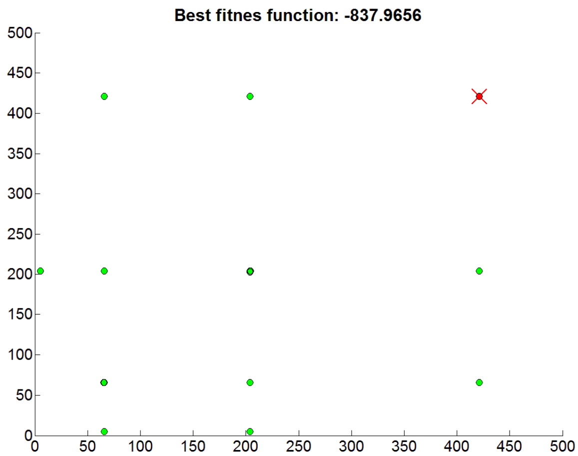

Fig.6 Schwefel function position after 100 000 steps

3.1.3 Griewangk’s function

Tested with parameters α = 0.3, β= 0.4, λ= 1.

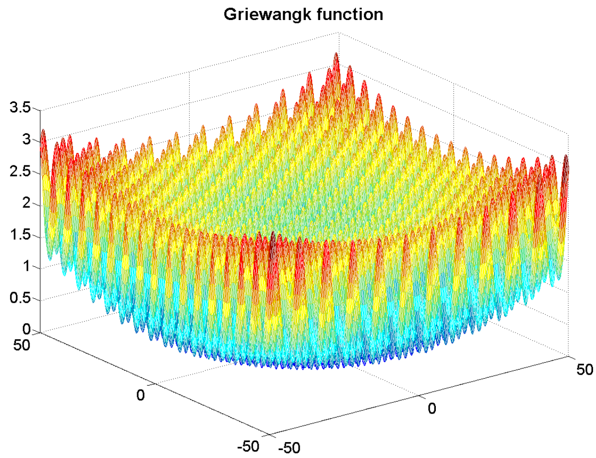

Fig.7 Griewangk’s function

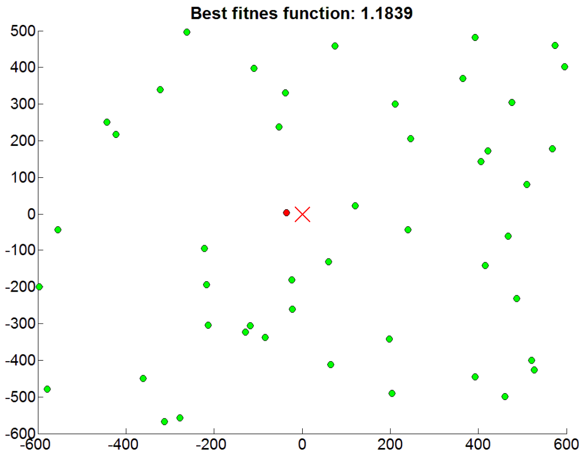

Fig.8 Griewangk's function starting position

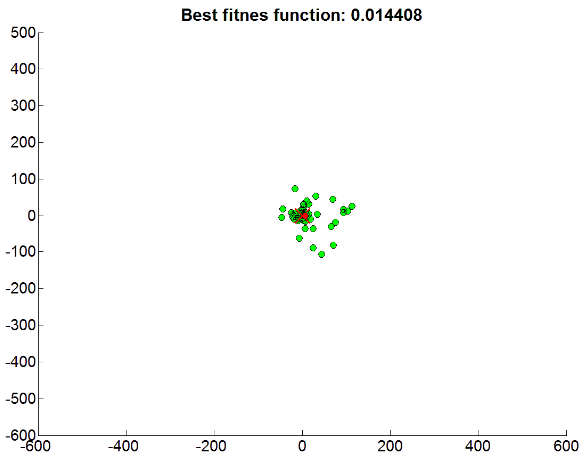

Fig.9 Griewangk's function position after 100 000 steps

3.1.4 Ackley function
Tested with parameters α = 0.1, β= 0.9, λ= 1.

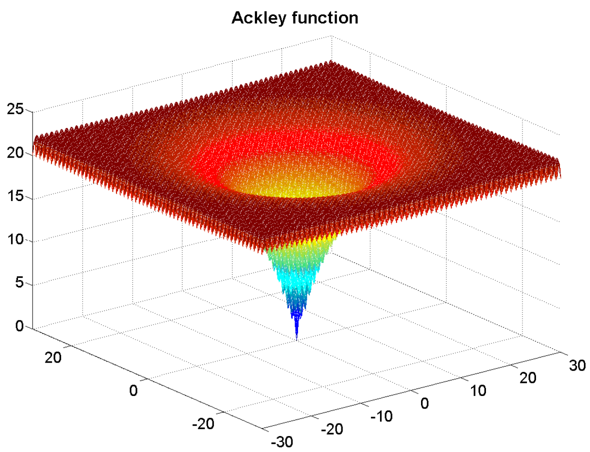

Fig.10 Ackley function

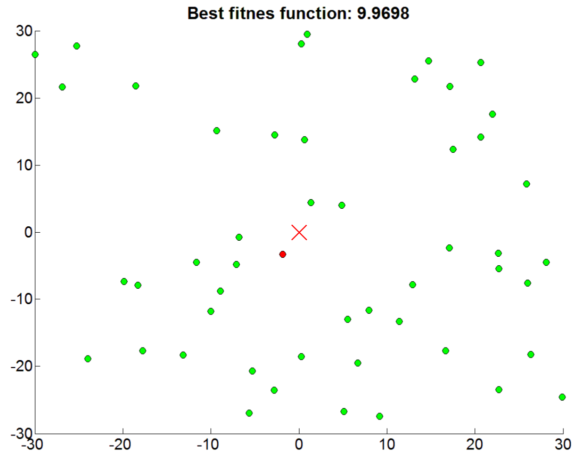

Fig.11 Ackley function starting position

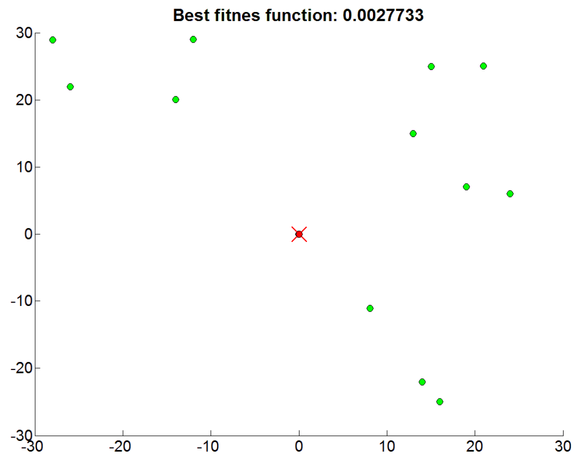

Fig.12 Ackley function position after 100 000 steps

3.1.5 Sine envelope wave function
Tested with parameters α = 0.3, β= 0.4, λ= 1.

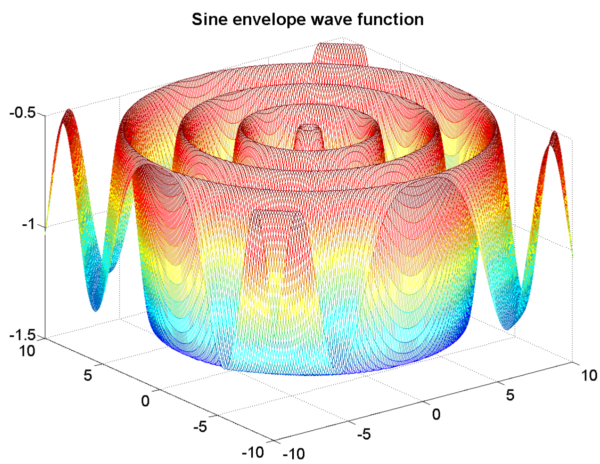

Fig.13 Sine envelope wave function

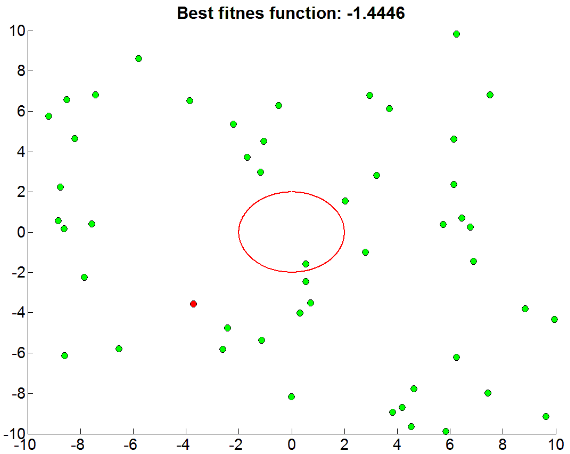

Fig.14 Sine envelope wave function starting position

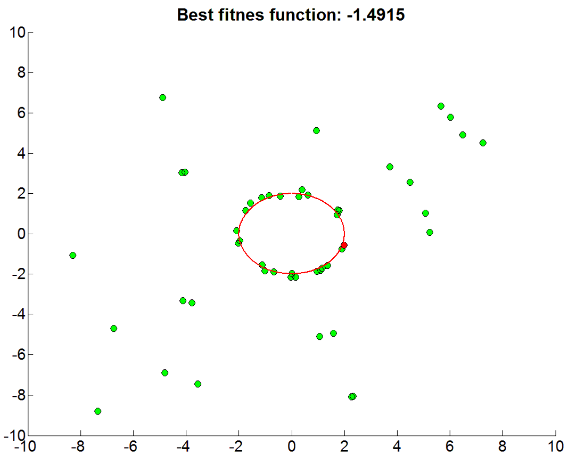

Fig.15 Sine envelope wave function position after 100 000 steps

3.1.6 Test function Ackley

Tested with parameters α = 0.3, β= 0.4, λ= 0.1.

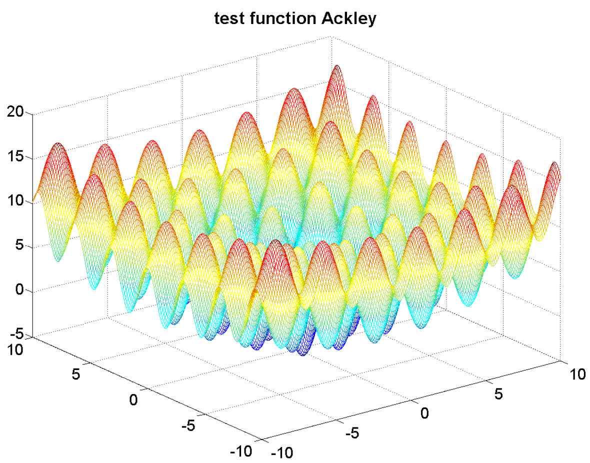

Fig.16 Test function Ackley

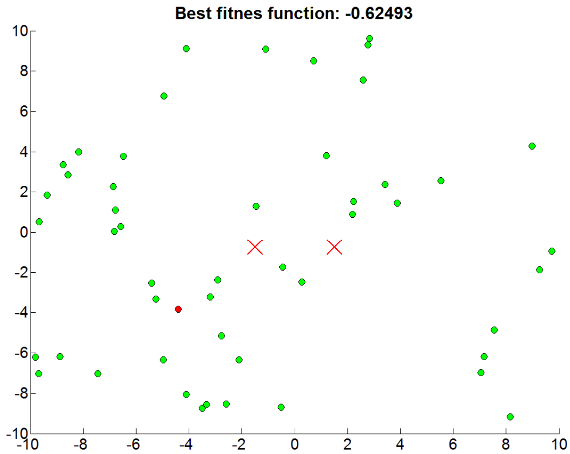

Fig.17 Test function Ackley starting position

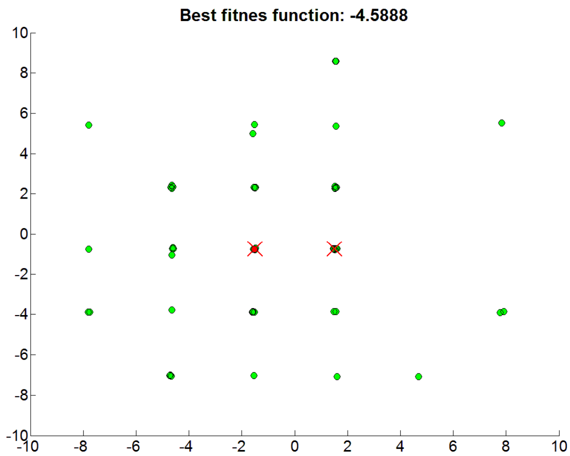

Fig.18 Test function Ackley position after 100 000 steps

3.1.7 Michalewicz function

Tested with parameters α = 0.3, β= 0.4, λ= 0.01.

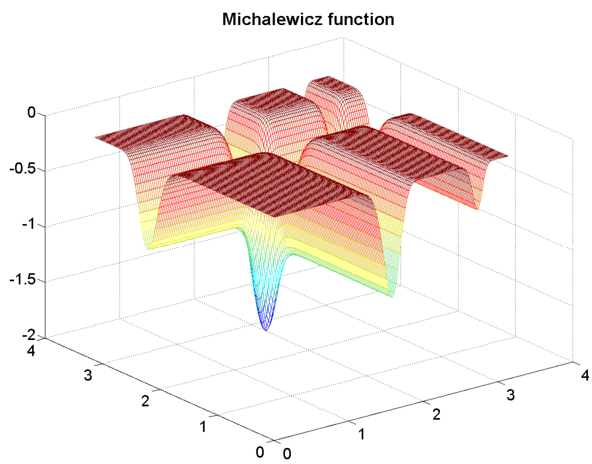

Fig.19 Michalewicz function

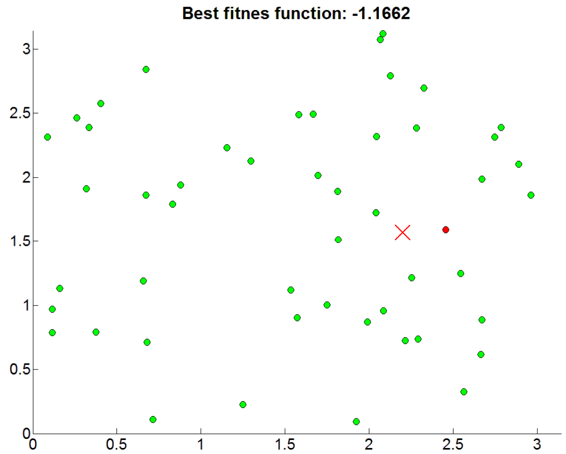

Fig.20 Michalewicz function starting position

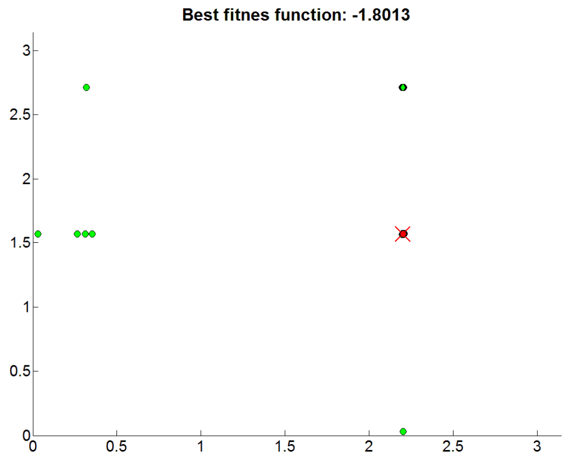

Fig.21 Michalewicz function position after 100 000 steps

3.1.8 Master's cosine wave function

Tested with parameters α = 0.3, β= 0.4, λ= 0.1.

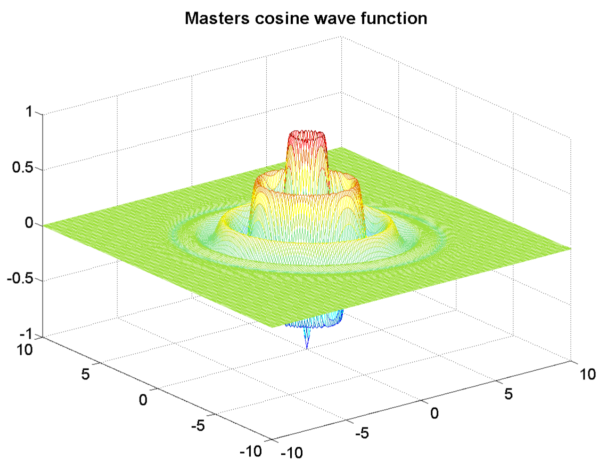

Fig.22 Master's cosine wave function

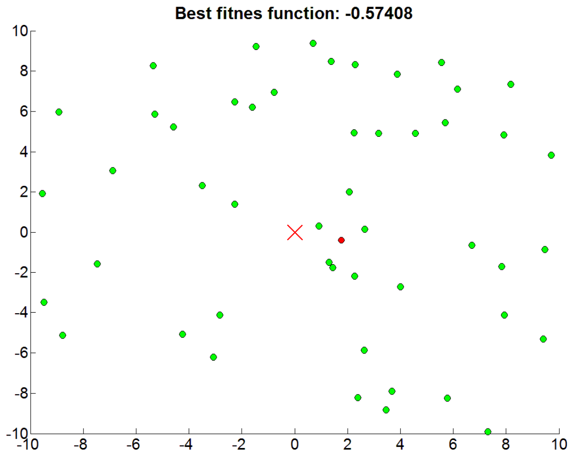

Fig.23 Masters's cosine wave function starting position

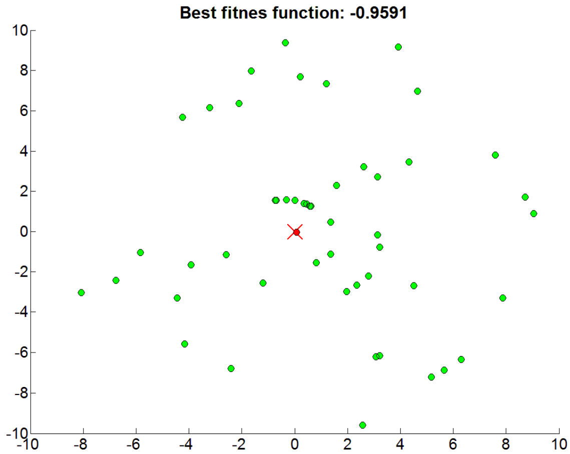

Fig.24 Master's cosine wave function position after 100 000 steps

3.2 Travelling salesman problem

The PSV algorithm was slightly modified for the purposes of the travelling salesman problem. Because the combination of the cities is discrete, the step λ equals 1 and the probability vector only determines the direction of the movement. Then the transformation (11) is modified to (12).

$$w_{i\_new} = t_i(w_i) = w_i\left(1 + \frac{sign(p_i)\lambda}{w_i}\right) \qquad (12)$$

Parameters of the algorithm are α = 0.3, β = 0.3, λ = 1, number of individuals = 5.

3.2.1 Random cities position

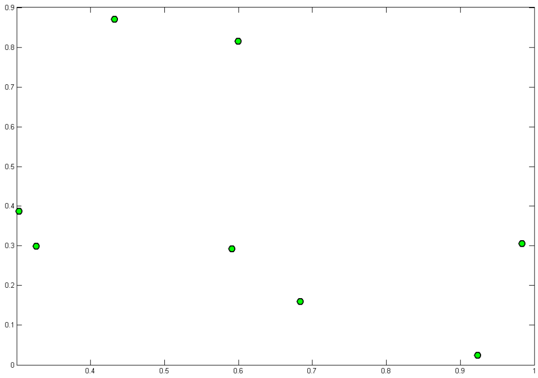

Fig.25 Random cities position

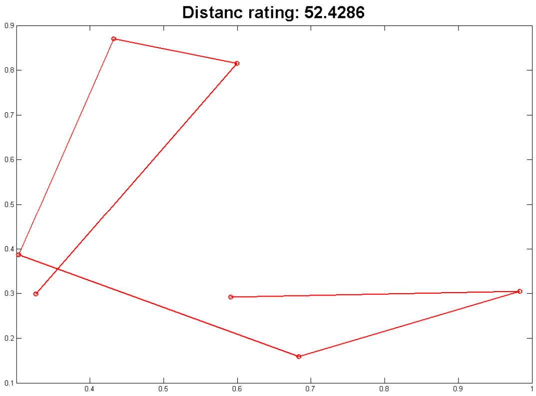

Fig.26 After 1000 steps

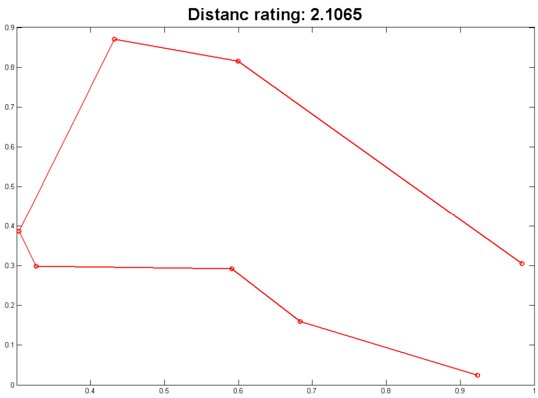

Fig.27 After 10 000 steps

3.2.2 Circle cities position

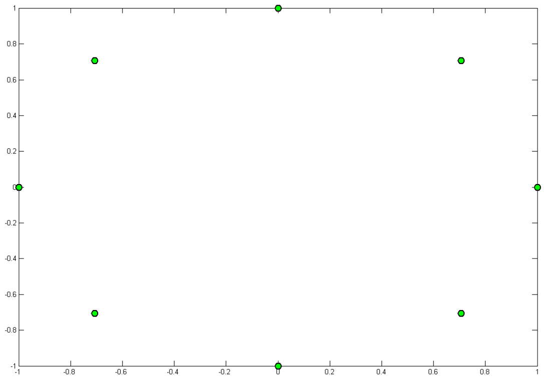

Fig.28 Circle cities position

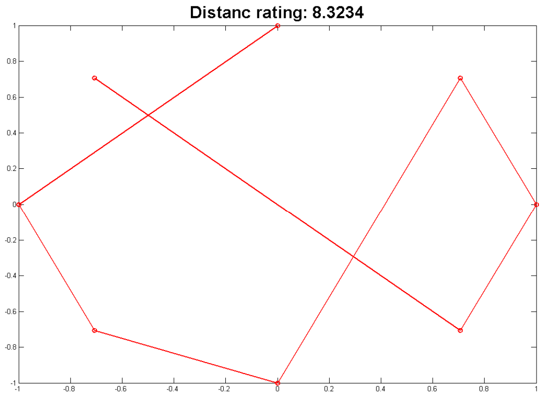

Fig.29 After 1 000 steps

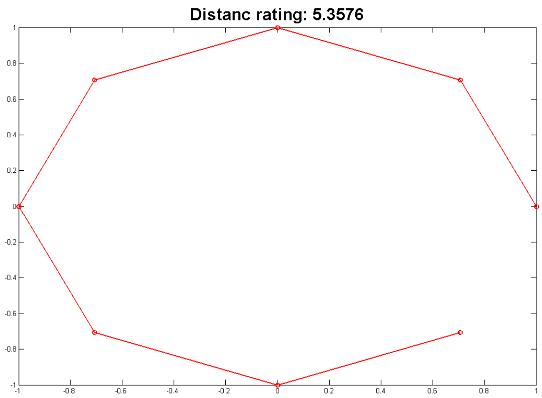

Fig.30 After 6 000 steps

4 Influence of the quantity of individuals upon the best achieved fitness function

At the same time with function minimization the influence of the individuals' quantity upon the best reached fitness function was tested. Every test had to fulfil the same rule: 100 000 fitness function calling. It is 100 000 steps (fitness function calling) for population of one individual and 2000 steps per individual for population of the size 50. The variables α, β, λ were the same as in examples 3.1.1-3.1.8. The colour lines on the examples 4.1.1-4.1.8 are average values from the ten experiments of evolution of the fitness function for every population (1, 10, 30, and 50). Red (dash-dot) line represents evolution of the fitness function for population of the 1 individual. Green (dashed) line represents evolution of the fitness function for population of the 10 individuals. Blue (dotted) line represents evolution of the fitness function for population of the 30 individuals and magenta (solid) line represents evolution of the fitness function for population of the 50 individuals.

4.1.1 Rastrigin's function

Tested with parameters α = 0.3, β= 0.4, λ= 0.1.

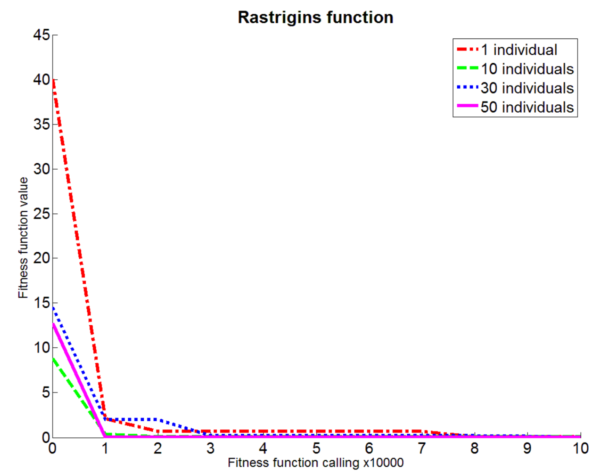

Fig.31 Rastrigin's function

4.1.2 Schwefel's function

Tested with parameters α = 0.3, β= 0.4, λ= 0.1.

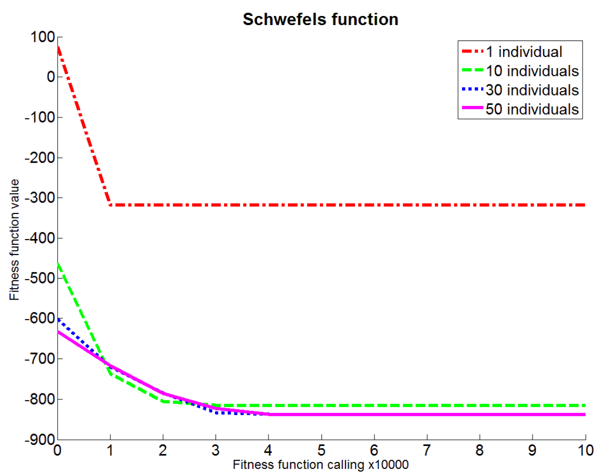

Fig.32 Schwefel function

4.1.3 Griewangk's function

Tested with parameters α = 0.3, β= 0.4, λ= 1.

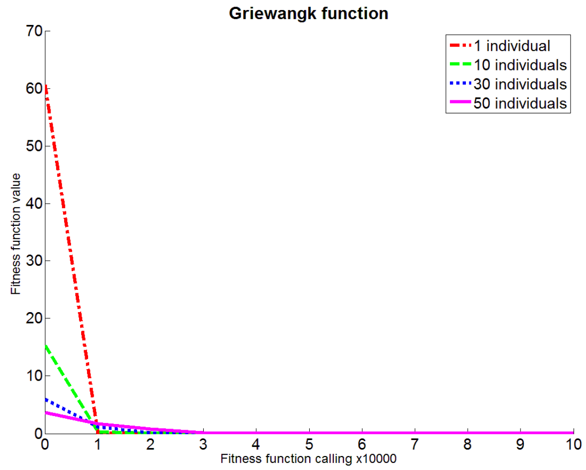

Fig.33 Griewangk's function

4.1.4 Ackley function

Tested with parameters α = 0.1, β= 0.9, λ= 1.

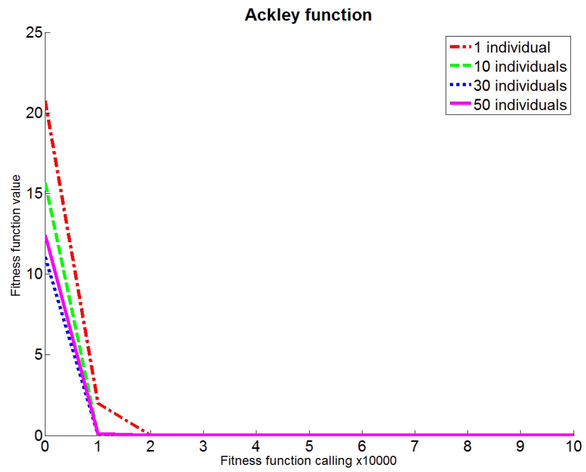

Fig.34 Ackley function

4.1.5 Sine envelope wave function

Tested with parameters α = 0.3, β= 0.4, λ= 1.

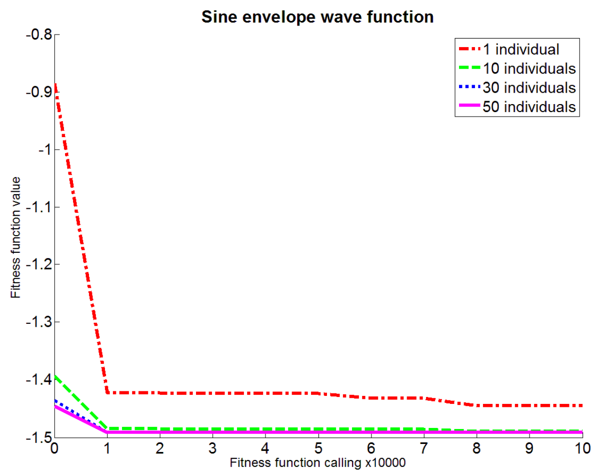

Fig.35 Sine envelope wave function

4.1.6 Test function Ackley

Tested with parameters α = 0.3, β= 0.4, λ= 0.1.

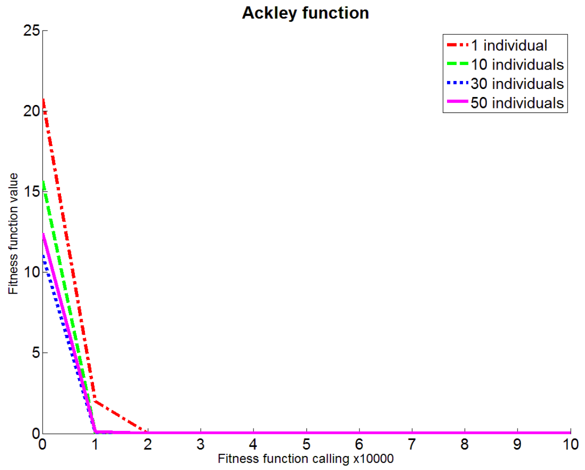

Fig.36 Test function Ackley

4.1.7 Michalewicz function

Tested with parameters α = 0.3, β= 0.4, λ= 0.01.

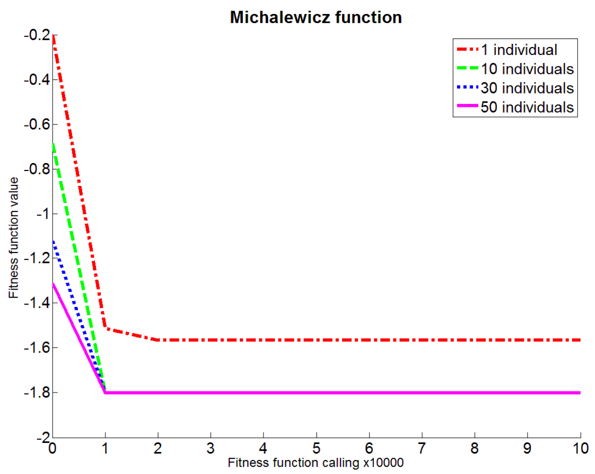

Fig.37 Michalewicz function

4.1.8 Master's cosine wave function

Tested with parameters α = 0.3, β= 0.4, λ= 0.1.

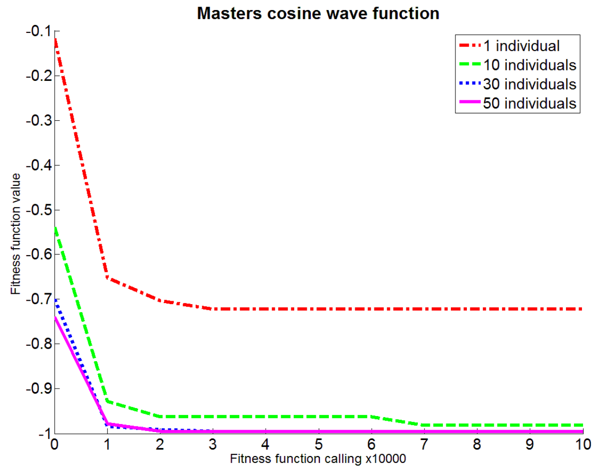

Fig.38 Master's cosine wave function

5 Conclusion

The newly developed algorithm PSV was introduced in this paper. The aim was to show its possible usage in two different fields of optimization (finding extreme and permutation problem).

Algorithm testing is still at the beginning phase. The main advantages of the PSV algorithm are its implementation simplicity and adaptability to different types of optimization problems. PSV can treat both the integers and the real numbers. The examples of solving function minimization problem (real number coding) and permutation problem (positive integer coding) were shown in chapters 3.1 and 3.2.

In chapter 3.1, 2D functions were chosen because it is possible to display them and it is possible to display the movement of individuals during optimization in 2D state space. The dot (green) represents the best fitness function reached by single individual during whole optimization process and the red dot represents the best fitness function reached in population.

In chapter 4 the influence of the population quantity upon the best reached fitness function was tested. As we expect in most of the examples the population with several individuals is more successful.

Future work will be focused on the extension of the current algorithm with distribution of the information among individuals in the population and on a modification of the algorithm with adaptive moving and adaptive momentum coefficients λ, α, β.

Acknowledgements

This work was supported by the Ministry of Education of the Czech Republic under Projects GA102/09/1897 and MSM0021630529.

References:

[1] Balicki. J.: *Tabu-based Evolutionary Algorithm with Negativ Selection for PAreto-optimization in Distributed Systems*, (AIKED'08), 7th WSEAS international conference on Artificial inteligence, knowledge engineering and data bases, Cambridge UK, pages 327-332, (2008), ISSN 1790-5109

[2] Jian Ye, Junfei Qiao, Ming-Ai Li ,Xiaogang Ruan: *A tabu based neural network learning algorithm*, ScienceDirect NEUROCOMPUTING 70 (2007) 875-882,

[3] Olteanu A. L.,: *Ant colony optimization meta-heuristic in project scheduling*, (AIKED'09), 8th WSEAS international conference on Artificial inteligence, knowledge engineering and data bases, Cambridge UK, pages 29-34, (2009), ISSN 1790-5109

[4] Onwubolu. C. G., Babu B.V..: *New Optimization Techniques in Engineering*, Springer-Heidelberg 2004, ISBN 3-540-20167-X

[5] Pohl, J., Polách, P.: *Stochastic Learning Algorithm, 15 th International on soft computing Mendel conference 2009*,ISSN 1803-3814, pp. 166-170.

[6] Pohl, J.: *Stochastický algoritmus učení pro umělé neuronové sítě s pravděpodobnostním směrovým vektorem,* Časopis pro elektrotechniku Elektrorevue, 09.11.2009, ISSN 1213-1539

[7] Schalkoff, J. R.: *Artificial neural networks*, McGraw-Hill international editions Computer Science Series (1997) , ISBN 0-07-115554-6

[8] Šíma, J., Neruda, R.: *Teoretické otázky neuronových sítí.* Praha: MATFYZPRESS, (1996), 390 s. ISBN 80-85863-18-9.

[9] Slušný S., Neruda R., Vinderová P.: *Behaviour Patterns Evolution on Individual and Groupe Level*, (CIMMACS'07), 6th WSEAS International Conference on computational intelligence, man-machine systems and cybernetics, pages 23-28, ISSN 1790-5117

[10] Ting-Yu Chen, Yi Liang Chen, *Global Optimization Using Hybrid Approach*, (SMO'07), 7th WSEAS International Conference on simulation, modelling and optimization, pages 309-313, ISSN 1790-5117

[11] Zelinka, I.: *Umělá intelligence v problémech globální optimalizace*, BEN Technická literatura, (2002), ISBN 80-7300-069-5

[12] Zelinka, I., Včelař, F., Čandlík, M.: *Fraktální geometrie principy a aplikace*, BEN Technická literature, (2006), ISBN 80-7300-191-8