#135

BICA VPS 2023

A hybrid PSO-Jaya algorithm for optimization problems.

E.M. Kazakova[1]. [1]Institute of Applied Mathematics and Automation KBSC RAS (IAMA KBSC RAS), Nalchik, Russia,
shogenovae@inbox.ru

INSTITUTE OF APPLIED MATHEMATICS AND AUTOMATION OF KABARDIN-BALKAR SCIENTIFIC CENTER OF RAS

SUMMARY

In this paper introduces and investigates a hybrid PSO-Jaya optimization algorithm based on two heuristic algorithms PSO and Jaya. Two problems: function optimization (Ackley (F1), function "eggholder" (F2), Holder (F3), Matyas (F4)) and training ANN for the classification problems Iris and breast cancer Wisconsin, are employed to evaluate the efficiencies of this new hybrid algorithm.

INTRODUCTION

A heuristic algorithm is a problem-solving method that uses rules of thumb or approximate strategies to find solutions. Swarm algorithms are metaheuristic methods that model the collective behavior of swarm organisms such as bees, fish, or birds to solve optimization problems. An example of a swarm algorithm is the Particle Swarm Optimization (PSO) algorithm. This optimization method, which uses the social behavior of one group of animals, was created by J. Kennedy and R. Eberhart in 1995 [1]. The Jaya algorithm (Jaya is a Sanskrit word meaning "Victory") is an optimization algorithm proposed by Rao R. in 2016 for solving optimization problems without constraints. It is based on the idea of improving the current best solution by comparing and updating all solutions in the population [2].

1. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: IEE International Conference on Neural Networks, pp. 1942-1948. (1995).

2. Rao, R.: Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems. Int J Indus Eng Comput 7(1), 19-34 (2016).

APPROACH

The PSO-Jaya hybrid algorithm is a combination of two optimization algorithms: PSO and Jaya. This hybrid approach combines the benefits of both methods to improve performance and the ability to find optimal solutions to complex optimization problems. At the initial iterations, the PSO particles are widely distributed in the space solution, but, as a rule, they are far from the solution of the problem. At the final iterations, the particles are concentrated in the vicinity of the found extremum, but this extremum is not guaranteed to be global. Therefore, as a condition for switching between algorithms, it is advisable to take the beginning of stagnation.

METHODS

The general principle of operation of the PSO-Jaya hybrid algorithm:

1. Initialization: an initial population of particles for PSO and Jaya.

2. PSO - phase: PSO particles iteratively update their position and velocity as they explore the solution space of formulas

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1} \quad (1),$$

$$v_{i,j}^{t+1} = \omega v_{i,j}^{t} + c_1 r_{1_{i,j}}^{t+1}\left(p_{best_{i,j}}^{t} - x_{i,j}^{t}\right) +$$

$$+ c_2 r_{2_{i,j}}^{t+1}\left(g_{best,j}^{t} - x_{i,j}^{t}\right) \quad (2)$$

respectively. A cost function is calculated for each particle, and the best solutions found by each particle and in the entire population are updated.

3. Jaya-phase: after completion of the PSO-phase, the stagnation condition is checked

$f(g_{best_{i+1}}) >= f(g_{best_i})$, if there is stagnation, then the updated solutions from the PSO are transferred to Jaya to search for another optimum. Using the Jaya algorithm, solutions are iteratively updated according to formula

$$x_{k,j}^{i\prime} = x_{k,j}^{i} + r_{1,j}^{i}\left(x_{bestj}^{i} - |x_{k,j}^{i}|\right) - r_{2,j}^{i}\left(x_{worstj}^{i} - |x_{k,j}^{i}|\right), \quad (3)$$

to improve their quality. After Jaya completes, particle positions in PSO are updated.

4. Stop criteria check: The algorithm continues to execute PSO phases and Jaya phases until a given stopping criterion is reached.

RESULTS

Table 1. Statistical error analysis $|f_{min} - f_{best}^{*}|$ for test functions, over 50 independent starts of 200 iterations, swarm size= 10

№	Algorithm	Average	Median	Standard deviation	"best" err
F1	PSO	0.0011	2.6076e-05	0.0037	3.6079e-08
	Jaya	7.3295e-11	0	5.1235e-10	0
	PSO_Jaya	**0**	**0**	**0**	**0**
F2	PSO	109.99	101.0395	102.1635	3.7279e-05
	Jaya	615.4141	713.7134	318.7311	3.7279e-05
	PSO_Jaya	**27.3561**	**3.7279e-05**	**39.9027**	**3.7279e-05**

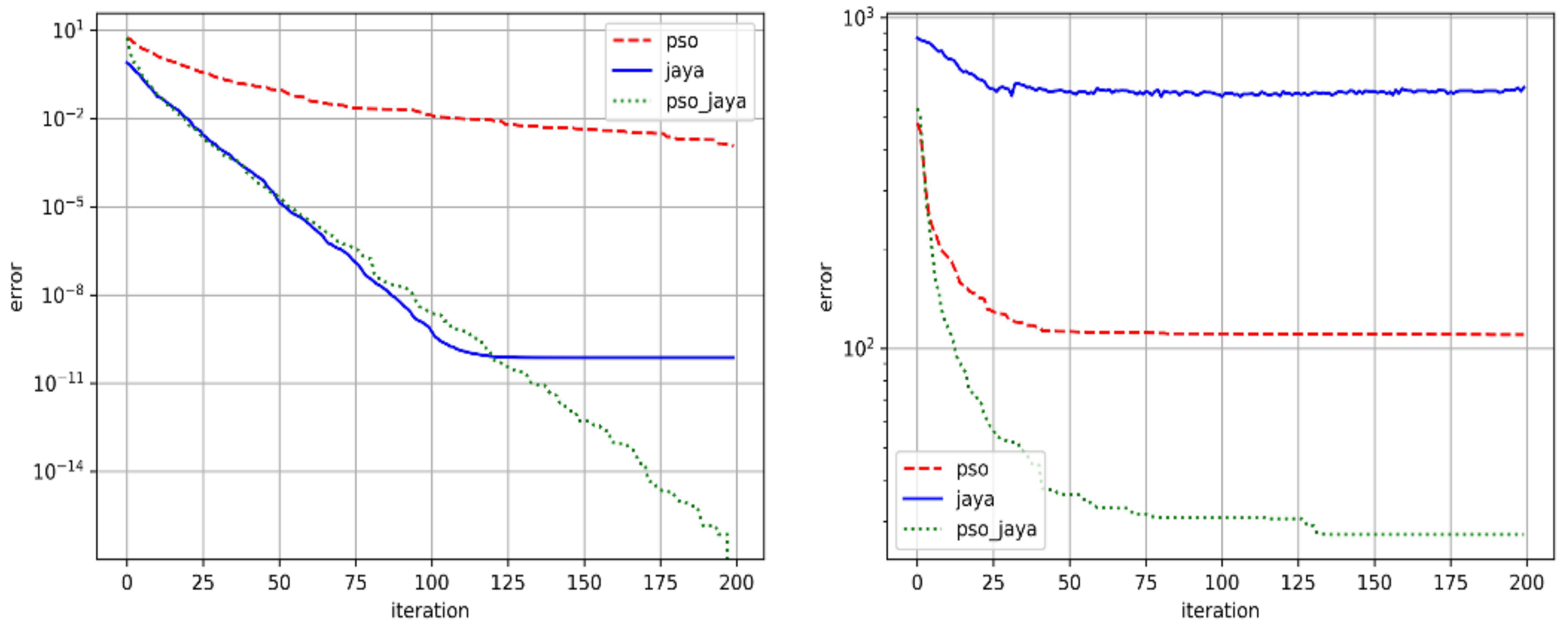

Fig. 1. Average error over 50 independent starts: a) Ackley's function

RESULTS

The neural network architecture is 4-8-3 (4 inputs, 8 neurons in the hidden layer, 3 output) for solving the Iris classification problem, and 30-15-2 for Breast cancer Wisconsin.

Table 2. Statistical error analysis MSE for datasets, over 30 independent starts of 500 iterations, size= 50

№	Algorithm	Average	Median	Stnd. deviation	"best" MSE
1	PSO	0.0117	0.0118	0.00075	0.0098
	Jaya	0.0259	0.0255	0.00073	0.0173
	PSO-Jaya	**0.0084**	**0.0086**	0.0015	**0.0046**
2	PSO	0.0604	0.0607	0.0044	0.0524
	Jaya	0.1172	0.1153	0.0111	0.0980
	PSO-Jaya	**0.0508**	**0.0511**	**0.0027**	**0.047**

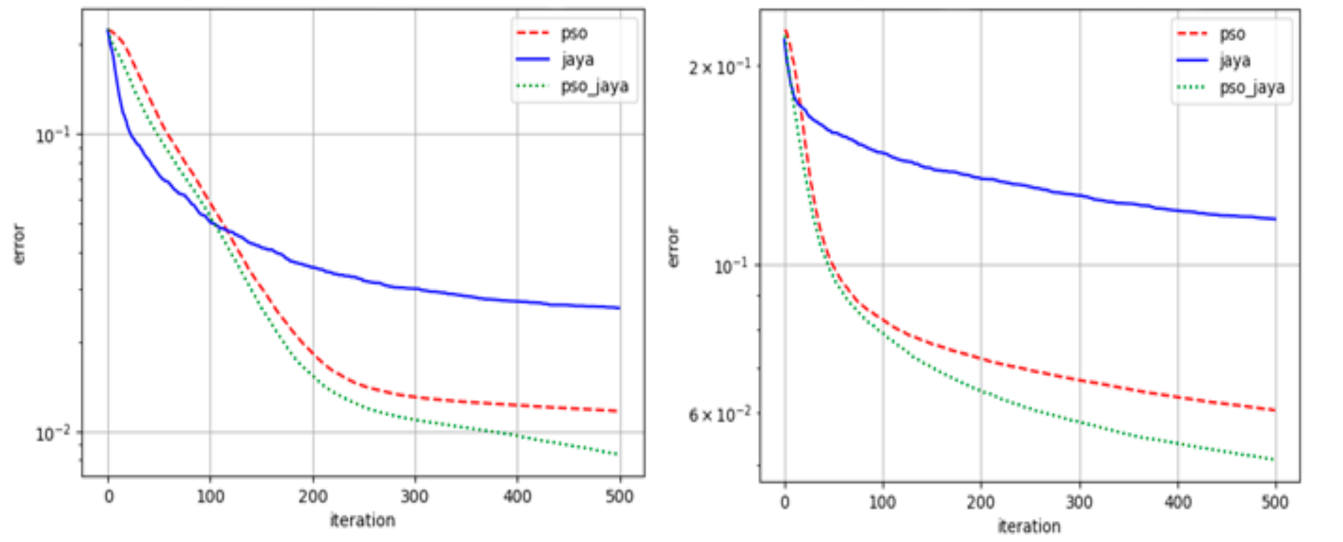

Fig. 2. Average of MSE: a) Iris; b) Breast cancer Wisconsin

CONCLUSIONS

The curves in Figure 2 show that the hybrid PSO-Jaya algorithm gives the best convergence rate for both classifiers.

Statistics show that the PSO-Jaya classifier has the best ability to avoid local minima. Also, for the minimum MSE, the hybrid algorithm gives better results, which indicates that the classifier is more accurate than the classifiers with the PSO and Jaya algorithms.