Population Initialization Methods for the Swallow Swarm Algorithm in Solving the Problem of Fuzzy Classifier Parameter Optimization

Artem Slezkin and Ilya Hodashinsky

Faculty of Security, Tomsk State University of Control Systems and Radioelectronics, 40 Lenin Ave. 634050, Tomsk, Russia

Abstract

This paper considers and compares population initialization methods for the swallow swarm algorithm in solving the problem of the fuzzy classifier parameter optimization. Population initialization is important in swarm and evolutionary optimization algorithms, in which the lack of diversity in the population can lead to early convergence and to hitting the local optimum. Methods based on quasi-random sequences, chaotic maps and random value distributions were considered. The hybrid initialization method based on the normal and uniform distribution revealed the lowest classification error. The fastest convergence was shown by the method based on the beta distribution.

Keywords

Population initialization, optimization, swallow swarm algorithm, fuzzy rule-based classifier

1. Introduction

Stochastic methods, called metaheuristics, make it possible to find satisfactory solutions for problems of large dimensions in reasonable time. Metaheuristics, which provide a new solution based on a single previous solution, belong to trajectory methods. Population metaheuristics give a solution based on the previous experience and information about the best solutions in the population. Among the many population metaheuristics, one can distinguish two most common classes: evolutionary and swarm algorithms [1].

The Swallow Swarm algorithm is one of the population swarm algorithms which has unique features that are absent in other swarm algorithms, such as the division of the population into several subpopulations with their local leaders and the use of three types of particles, with each of them performing certain functions in the population. Due to these features, the algorithm has a good rate of convergence and copes with the problem of leaving local optima [2, 3]. The algorithm is used for solving the problems of node control in sensor networks [4], for optimizing digital filter parameters [5], and for solving the problem of selecting tests for fault diagnostics [6].

Population initialization is important in the swarm and evolutionary optimization algorithms, in which the lack of diversity in the population can lead to early convergence and to hitting the local optimum. The efficiency of the optimization methods can be improved by developing methods for generating perspective initial solutions. However, to date there are no systematic studies of the initialization process and impact of the initial solution distributions on the efficiency of the swarm and evolutionary algorithms when solving optimization problems [1,7].

Many real-world problems can be reduced to the problem of classification. Fuzzy classifiers use fuzzy rules to describe the relationship between the values of the features of an object and the class to which the object belongs. The main advantages of the fuzzy classifiers are their interpretability and

SibDATA 2021: The 2nd Siberian Scientific Workshop on Data Analysis Technologies with Applications 2021, June 25, 2021, Krasnoyarsk, Russia

EMAIL: hodashn@gmail.com (I. Hodashinsky)
ORCID: 0000-0002-8644-9698 (A. Slezkin); 0000-0002-9355-7638 (I. Hodashinsky)

tolerance to inaccurate and missing data [8]. The simple and clear architecture of the fuzzy classifiers, as well as a large number of training schemes, including the gradient, swarm, and evolutionary methods, have allowed the fuzzy classification to play one of the central roles in data mining [9, 10, 11].

The purpose of this study is to compare the population initialization methods for the swallow swarm algorithm when solving the problem of optimizing the fuzzy classifier parameters.

2. Literature review

The initialization procedure includes specifying the initial population size and actually creating the initial population. The population size affects the reliability and computational cost of the algorithm. A small population size can lead to fast convergence to the local optimum, while a large population size will increase the computational costs and may lead to slow convergence [12]. The population size can be related to the dimensionality of the search space. However, most often the population size is chosen experimentally. Another approach to solve this problem is to introduce an adaptive method of forming the population size, which makes the algorithm less sensitive to the initial choice of the population size [1].

It is possible to distinguish several different approaches to creating the initial population: stochastic methods, quasi-random sequences, oppositional learning method, multistep method, chaotic mappings (maps) [1,13]. The stochastic methods can be divided into two groups: pseudorandom number generators and chaotic number generators. Levy distribution, beta distribution, normal distribution, lognormal distribution, etc. are used to create the initial population along with a uniform distribution. [7]. The main advantage of the method based on the generation of pseudorandom numbers is the simplicity of its realization. Chaotic number generators are based on Gaussian mapping, logistic mapping, sinusoidal mapping, tent-map mapping, etc. [13,14]. Using the chaotic initialization methods can improve the efficiency of the metaheuristic optimization algorithms in terms of population diversity and convergence rate [15].

Quasi-random sequences are neither random nor pseudorandom, their generation algorithms do not use random elements, and these sequences are fully deterministic. The advantage of the method based on quasi-random sequences is the homogeneity of the generated initial population, which allows increasing the research capability of the algorithm at the early stages of its operation [13]. In [16] an improvement of the gravitational search algorithm by generating the initial population using the Sobol quasi-random number generator is described. To initialize the population of the swarming particle algorithm, the quasi-random Van der Corput sequences were used in [17], while in [18] the Faure sequences were used.

In [15], to improve the convergence of the artificial bee colony algorithm, an initial population is created by combining chaotic systems with oppositional learning. The idea of initialization based on oppositional learning is to simultaneously generate both the current solutions and the solutions opposite to the current ones, using opposite numbers for this purpose [19]. When considering several test problems using the differential evolution algorithm, the authors [20] proved that it is possible to improve the efficiency of metaheuristics by using oppositional learning for the initialization and during the optimization process.

3. Fuzzy classifier

The fuzzy classifier rules have the following form [21]:

$$IF\ x_1 = A_{1j}\ AND\ x_2 = A_{2j}\ AND\ \ldots\ AND\ x_n = A_{nj}\ THEN\ class = c_j,$$

where $j = \overline{1...m}$, m is the number of rules, A_{ij} is the term of the j-th rule of the i-th variable, c_j is the identifier of the j-th level.

Then, the fuzzy classifier itself can be represented as:

$$C = f(\mathbf{x}, \boldsymbol{\theta}), \tag{1}$$

where **x** is the input vector, **θ** is the vector of the parameters of the antecedent membership functions.

Then, the optimization task of the membership function parameters will be to find such a vector of parameters **θ**, at which the input data classification quality will be maximum. However, the population metaheuristic optimization algorithms do not require a single initial solution, but a set of them.

4. Creation of initial populations

The most common approaches to initializing the initial population are considered in this study: quasi-random sequences, chaotic mappings and methods based on distributions of random variables. Of the above approaches, the methods having the best results in numerous experiments were chosen. The following notations will be introduced: *N* is the size of the initialized population, $\boldsymbol{\theta}_0$ is the vector obtained after the formation of the initial base of the fuzzy rules.

4.1. Methods based on quasi-random sequences

Halton's method. The Halton's formation of quasi-random vectors [22] is based on the Van der Corput transformation, a sequence of positive integers which transforms the natural number *n* for the given number *r* in the case when *n* has a representation in the *r*-number system according to the expression:

$$m = \sum_{k=1}^{m} a_k r^{k-1} \text{ , then } \varphi_r(n) = \sum_{k=1}^{m} a_k r^{-k}, \tag{2}$$

Faure's method. In the Faure sequence [23] the *n*-th element can be expressed using the following expression:

$$\left(\phi_b(\mathrm{P}^0 n), \phi_b(\mathrm{P}^1 n), \dots, \phi_b(\mathrm{P}^{s-1} n)\right), \tag{3}$$

where *b* is the prime number, *b* ≥ *s*, *P* is the Pascal matrix, whose element (*i, j*) is $\begin{pmatrix} i-1 \\ j-1 \end{pmatrix}$, where $i \geq 1,\ j \geq 1$.

4.2. Methods based on chaotic maps

To initialize the population based on chaotic procedure maps, the initial vector $\boldsymbol{\theta}_0$, is given as an input at the first iteration, and at each of the following N-2 iterations the result obtained at the previous iteration is given. The initial vector $\boldsymbol{\theta}_0$ is added to all the generated vectors.

The method based on the chaotic Gauss iterated map. This method generates the initial population according to the formula:

$$\boldsymbol{\theta}_{n+1} = \exp(-\alpha \boldsymbol{\theta}_n^2) + \beta, \tag{4}$$

where *α* and *β* are the real coefficients, $0 \leq n \leq N-2$.

The method based on the chaotic tent map. This method initializes the population according to the following formula:

$$\boldsymbol{\theta}_{n+1} = f_\mu(\boldsymbol{\theta}_n) = \begin{cases} \mu\boldsymbol{\theta}_n, & \boldsymbol{\theta}_n < 0.5 \\ \mu(1-\boldsymbol{\theta}_n), & 0.5 \leq \boldsymbol{\theta}_n \end{cases}, \tag{5}$$

where *μ* is the positive integer, $0 \leq n \leq N-2$.

4.3. Methods based on distributions of random variables

To initialize the population, N-1 vectors of random variables are generated, with each of them element-wise multiplied with the initial vector $\boldsymbol{\theta}_0$. The last vector added to the population was the initial vector $\boldsymbol{\theta}_0$**.** Here, normal, uniform, gamma, and beta distributions are used.

5. Experiment description

A computational experiment was performed on 34 datasets from the KEEL repository (http://keel.es/) using tenfold cross-validation. The fuzzy classifier was used as a classifier. The swarm algorithm was used to optimize the parameters of the membership functions [2, 24]. Ten initialization methods presented in Table 1 were analyzed. The parameters of the initialization methods are shown in Table 2. The following parameters of the swarm swallow algorithm were established: population size - 40, number of local leaders - 3, number of aimless particles - 6, number of iterations - 100. The choice of the parameter values is due to the experiments conducted earlier [24]. For each data set, the experiment was repeated 30 times, and then, the results were averaged. Two parameters were evaluated during the experiment: the classification error on the test data and the number of iterations performed until convergence.

Table 1

Population initialization methods

Method	Abbreviation
Faure's method	Faure
Halton's method	Halton
Chaotic Gauss Iterated Map	Gauss Map
Chaotic Tent Map	Tent Map
Method based on normal distribution	Normal
Method based on uniform distribution	Uniform
Method based on gamma distribution	Gamma
Method based on beta distribution	Beta
Hybrid method based on uniform and normal distributions	UN
Hybrid method based on uniform, normal, beta and gamma distributions	BGNU

Table 2

Parameters of the population initialization methods

Method abbreviation	Parameters
Gauss Map	α = rand(1;3), β = rand(-1;1)
Tent Map	$\mu = 2$
Uniform	$a = 0.5$; $b = 2$
Normal	$\mu = 1$; $\sigma = 1$
Gamma	$k = 1$; $\theta = 1$
Beta	$\alpha = 0.5$; $\beta = 0.5$

The description of the data sets and the results of the initialization methods are presented in [25].

The Friedman two-factor analysis of variance for related samples (significance level equal to 0.05) was used for the statistical evaluation of the initialization methods. Tables 3 and 4 show the ranks of each of the population initialization methods for the parameters evaluated. The p-value for the classification error is 2.74E-07, and the p-value for convergence is 0.021.

Table 3
Ranks of the population initialization methods with regard to the classification error

Faure	Halton	Normal	Uniform	UN	BGNU	Beta	Gamma	Gauss Map	Tent Map
6.76	7.29	4.71	5.00	3.96	4.34	4.44	5.43	6.25	6.82

Table 4
Ranks of the population initialization methods with regard to convergence

Faure	Halton	Normal	Uniform	UN	BGNU	Beta	Gamma	Gauss Map	Tent Map
6.37	6.16	4.62	6.44	5.1	5.22	4.5	4.65	5.9	6.04

All the *p*-values are lower than the significance level of 0.05; thus, the results are statistically significant and the null hypotheses of equality of classification errors and equality of the performed iterations to the point of convergence are rejected.

In Figure 1 the asterisks mark the elements that are part of the Pareto set. Regarding the classification error and convergence considered as separate criteria, the UN method has the minimum classification error, while the Beta method shows the best convergence.

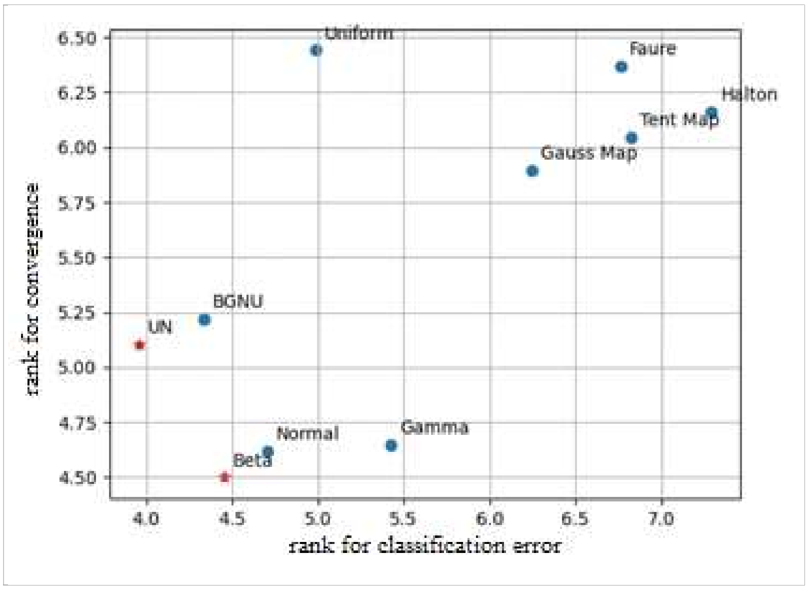

Figure 1: Graphical representation of the results of the initialization methods

6. Conclusion

In this research ten methods of initialization of the swarm algorithm population were studied to solve the problem of optimization of the fuzzy classifier parameters when classifying 34 datasets. The hybrid initialization method based on the normal and uniform distribution showed the lowest classification error. This population initialization method is recommended if minimizing the classification error is a priority. The fastest convergence was shown by the method based on the beta distribution. This method is recommended if the priority is the speed of the algorithm.

In future studies, it is planned to test the proposed population initialization methods on other classifiers and other metaheuristic algorithms.

7. Acknowledgements

This work was supported by the Ministry of Science and Higher Education of the Russian Federation (Project No. FEWM-2020-0042).

8. References

[1] I. A. Hodashinsky, Methods for Improving the Efficiency of Swarm Optimization Algorithms. A Survey, Automation and Remote Control 82(6) (2021) 935–967.

[2] M. Neshat, G. Sepidnam, M. Sargolzaei, Swallow swarm optimization algorithm: a new method to optimization, Neural Computing and Application 23(2) (2013) 429–454.

[3] M. Neshat, G. Sepidnam, A new hybrid optimization method inspired from swarm intelligence: Fuzzy adaptive swallow swarm optimization algorithm (FASSO), Egyptian Informatics Journal 16(3) (2015) 339–350.

[4] A. Nithya, A. Kavitha,Node Weight Swallow Swarm Optimization Convex Node Segmentation (Nws^{2}cns) Algorithm for Distributed 3-D Localization in Wireless Sensor Networks (Wsns), International Journal of Scientific & Technology Research 9(2) (2020) 5130–5137.

[5] S. K. Sarangi, R. Panda, A. Abraham, Design of optimal low-pass filter by a new Levy swallow swarm algorithm, Soft Computing 24 (2020) 18113–18128.

[6] Z. Yao, L. Zhu, T. Zhang, J.Wang, Optimal Selection of Tests for Fault Diagnosis in Multi-Path System with Time-delay, Journal of Electronic Testing 36 (2020) 75–86.

[7] Q. Li, S.-Y. Liu, X.-S.Yang, Influence of initialization on the performance of metaheuristic optimizers, Applied Soft Computing 91 (2020) 106193.

[8] P. Carmona, J.L. Castro, Fuzzy Feature Rank. Bringing order into fuzzy classifiers through fuzzy expressions, Fuzzy Sets and Systems 401 (2020) 78–90.

[9] X. Hu, W. Pedrycz, X.Wang, Fuzzy classifiers with information granules in feature space and logic-based computing, Pattern Recognition 80 (2018)156–167.

[10] I. A. Hodashinsky, I. V. Gorbunov, Algorithms of the tradeoff between accuracy and complexity in the design of fuzzy approximators, Optoelectronics, Instrumentation and Data Processing 49, (2013) 569–577.

[11] A. Lavygina, I. Hodashinsky, Hybrid algorithm for fuzzy model parameter estimation based on genetic algorithm and derivative based methods, in: ECTA 2011 FCTA 2011, Proceedings of the International Conference on Evolutionary Computation Theory and Applications and International Conference on Fuzzy Computation Theory and Applications, Portugal, 2011, pp. 513–515.

[12] D. B. Chen, C. X. Zhao, Particle Swarm Optimization with Adaptive Population Size and Its Application, Applied Soft Computing 9(1) (2009) 39–48.

[13] B. Kazimipour, X. Li, A. K. Qin, A review of Population Initialization Techniques for Evolutionary Algorithms, in: Proceedings of IEEE Congress on Evolutionary Computation, 2014, pp. 2585–2592.

[14] B. Alatas, Chaotic bee colony algorithms for global numerical optimization, Expert Systems with Applications 37 (2010) 5682–5687.

[15] W.-f. Gao, S.-y. Liu, A modified artificial bee colony algorithm, Computers & Operations Research 39(3) (2012) 687–697.

[16] O. T. Altinoz, A .E. Yilmaz, G. Weber, Improvement of the Gravitational Search Algorithm by Means of Low-Discrepancy Sobol Quasi Random-Number Sequence Based Initialization, Advances in Electrical and Computer Engineering 14(3) (2014) 55–62.

[17] M. Pant, R. Thangaraj, A. Abrana, Low Discrepancy Initialized Particle Swarm Optimization for Solving Constrained Optimization Problems, Fundamenta Informaticae 95 (2009) 511–531.

[18] R. Brits, A. P. Engelbrecht, F. Van den Bergh, A Niching Particle Swarm Optimizer, in: Proc. 4th Asia-Pacific Conference on Simulated Evolution and Learning, 2002, pp. 692–696.

[19] S. Mahdavi, S. Rahnamayan, K. Deb, Opposition Based Learning: A Literature Review, Swarm and Evolutionary Computation 39 (2018) 1–23.

[20] S. Rahnamayan, H. R. T izhoosh, M. M. A. Salama, Opposition-Based Differential Evolution, in: IEEE Transactions on Evolutionary Computation, volume 12(1), 2008, pp. 64–79.

[21] M. Bardamova, A. Konev, I. Hodashinsky, A. Shelupanov, A fuzzy classifier with feature selection based on the gravitational search algorithm, Symmetry 10(11) (2018) 609.

[22] S. M. Ermakov, On the Halton quasi-random sequences randomization, Vestnik of St Petersburg University. Mathematics. Mechanics. Astronomy 4(62) (2017) 570–576.

[23] H. Faure, Good permutations for extreme discrepancy, Journal of Number Theory 42 (1992) 47–56.

[24] I. Hodashinsky, K. Sarin, A. Shelupanov, A. Slezkin, Feature selection based on swallow swarm optimization for fuzzy classification, Symmetry 11(11) (2019) 1423.

[25] A. Slezkin, Population Initialization Methods for Swallow Swarm Algorithm when Solving Fuzzy Classifier Parameters Optimization Problem, Mendeley Data 2 (2021). doi: 10.17632/zwntkycn83.2.