ISSN 0361-7688, Programming and Computer Software, 2019, Vol. 45, No. 5, pp. 221–227. © Pleiades Publishing, Ltd., 2019.
Russian Text © The Author(s), 2019, published in Programmirovanie, 2019, Vol. 45, No. 5.

Feature selection: Comparative Analysis of Binary Metaheuristics and Population Based Algorithm with Adaptive Memory

I. A. Hodashinsky[a],* and K. S. Sarin[a],**

[a] *Tomsk State University of Control Systems and Radio Electronics, Tomsk, 634050 Russia*

**e-mail: hodashn@rambler.ru*

***e-mail: sks@security.tomsk.ru*

Received May 15, 2018; revised October 31, 2018; accepted October 31, 2018

Abstract—The NP-hard feature selection problem is studied. For solving this problem, a population based algorithm that uses a combination of random and heuristic search is proposed. The solution is represented by a binary vector the dimension of which is determined by the number of features in the data set. New solution are generated randomly using the normal and uniform distribution. The heuristic underlying the proposed approach is formulated as follows: the chance of a feature to get into the next generation is proportional to the frequency with which this feature occurs in the best preceding solutions. The effectiveness of the proposed algorithm is checked on 18 known data sets. This algorithm is statistically compared with other similar algorithms.

DOI: 10.1134/S0361768819050037

1. INTRODUCTION

Feature selection is an important problem in intelligent data analysis and machine learning aimed at reducing the size of the input data and at improving the effectiveness of classification, clustering, regression, and prediction of time series algorithms [1]. The result of the feature selection is a synthetic representation, usually called a feature vector [2].

Structured data used for the analysis is usually represented by a set of objects or observations that are represented by rows and a number of features (variables, attributes, or columns), which actually describe objects of the real world. Primary data can include a lot of irrelevant or redundant features. A feature is relevant if it significantly affects the result of classification, regression, or prediction. A feature is redundant if it is strongly correlated with other features [3].

Usually three feature selection technologies are distinguished: filters, wrappers, and embedded methods [4, 5]. Filters evaluate the relevance of features on the basis of only internal data properties, and they are independent of the classification algorithm. In the wrapper technology, the subset of selected features is evaluated in the process of learning and (or) testing a specific classifier. In embedded methods, the optimal subset of features is sought in the process of constructing the classifier, and it can be considered as the search in the joint space of feature subsets and classifier parameters [5].

Feature selection algorithms can be categorized into batch methods and online algorithms. In the first case, the feature selection problem is solved offline when all instances of the data set are already available. In the second case, the instances of data and features are not known in advance but arrive in a sequential manner [6].

The feature selection problem was solved using various search methods, such as exhaustive search, greedy algorithms, and random search. However, the majority of existing feature selection methods can be trapped into local optima or are computationally costly [2]. Feature selection is an NP-hard problem [4], and the optimal solution can be guaranteed only by exhaustive search. The use of metaheuristic techniques makes it possible to obtain suboptimal solutions without examining the entire space of solutions.

The purpose of this paper is to describe a new population based algorithm with adaptive memory for solving the feature selection problem in batch mode and to compare the effectiveness of the proposed algorithm with other similar algorithms.

2. RELATED WORKS

2.1. Metaheuristics for Feature Selection

The complexity of the feature selection problem is caused not only by the large search space but also by interrelations between features. A feature that weakly affects the classification accuracy when considered separately can significantly improve this accuracy in combination with other features. For this reason, features should not be evaluated individually—an evaluation should be given to the entire subset of features [1].

Feature selection has two goals—maximization of the classification accuracy and minimization of the number of features. These goals are contradictory; therefore, feature

selection can be considered as a two-criteria optimization problem [1], which can be solved using metaheuristic methods, such as evolutionary computations, swarm intelligence techniques, and their hybrids.

In [7], three metaheuristic strategies for feature selection—GRASP, tabu search, and memetic algorithm—were studied. These three strategies were compared with some other feature selection techniques, including the family of greedy search algorithms. In [8], the harmonic search algorithm is used for feature selection, and the choice of the optimal classifier on the selected subsets of features is based on Akaike's information criterion. To select the optimal feature subset in wrapper mode, in [9] binary options of the whale algorithm are proposed; here the basic operators of the continuous whale optimization algorithm are replaced by binary operations and a number of evolutionary operators (selection, crossover, and mutation) are added. In [10], feature selection methods based on the combined use of two hybrid approaches of artificial bee colony with particle swarm optimization and artificial bee colony with genetic algorithm were studied. In reviews [1, 3], a deep and comprehensive analysis of the application of evolutionary computations and swarm intelligence method for feature selection was given. Paper [1] contains 45 references to works in which genetic algorithms in wrapper mode were used for feature selection; it also references 18 works in which genetic programming was used for feature selection; 29 and 16 references were made, respectively, to works on feature selection using the particle swarm and ant colony algorithm; the bee colony algorithm is mentioned six time, and the differential evolution algorithm is mentioned seven times. Memetic and bee colony algorithms, gravitational search, artificial immune system algorithms, and evolutionary strategy are slightly less popular.

Among drawbacks of metaheuristic methods are high computational complexity due to a large number of calculation of estimates and low stability, which manifests itself in that different subsets of features can be obtained after each execution of the algorithm; in turn, this can require the development of methods for further selection from the selected subsets of features [1].

2.2.Adaptive Memory

The use of memory in metaheuristics was first proposed by Glover in the tabu search technique, which actually is a local search based on the concepts of neighborhood and adaptive memory function that prevents the repeated search through earlier found solutions [11].

Adaptive memory underlies any kind of learning; moreover, such important procedures as intensification and diversification are most often implemented using adaptive memory [12]. The triad intensification–diversification–learning is investigated in [12] by combining relaxation adaptive memory programming algorithm, which integrates adaptive memory programming as embodied in tabu search, with Lagrangian-based heuristic.

In [13], a population based adaptive simplex method for stochastic optimization problems was proposed. It uses the reflection and contraction operators of the classical Nelder–Mead simplex method, the local search strategy, and mechanisms for detecting stagnation and eliminating duplicates. The adaptive probability threshold allows the authors of [13] to control the convergence process of the algorithm.

According to the no-free-lunch theorem, there are no metaheuristic algorithms that can successfully solve all optimization problems. If a certain metaheuristic algorithm outperforms other algorithms for a specific class of problems, one cannot be sure that it will be effective for another class of optimization problems. This fact makes researchers propose new metaheuristic algorithms and improve the existing ones [15].

3. STATEMENT OF THE PROBLEM

Introduce the following notation:

$X = \{x_1, x_2, ..., x_n\}$ is the set of input features;

$\mathbf{S} = (s_1, s_2, ..., s_n)^{\mathrm{T}}$ is the binary solution vector.

The variables in the problem are

$$s_i = \begin{cases} 1 & \text{if the } i\text{th feature is used in the classifier} \\ 0, & \text{otherwise} \end{cases}$$

The accuracy of solution *acc* obtained by the classifier on the observation table $\{(\mathbf{x}_i, c_i), i = 1, 2, ..., z\}$ using the features in the vector $\mathbf{S}$ is computed by

$$acc(\mathbf{S}) = \frac{\sum_{i=1}^{z} \begin{cases} 1 & \text{if } f(\mathbf{x}_i; \mathbf{S}) = c_i \\ 0, & \text{otherwise} \end{cases}}{z};$$

here $f(\mathbf{x}_i; \mathbf{S})$ is the output of the classifier for the instance of input data $\mathbf{x}_i$ obtained using the features in $\mathbf{S}$.

Each solution is evaluated according to an objective function that depends on the classification accuracy and the number of selected features:

$$F(\mathbf{S}) = \alpha(1 - acc(\mathbf{S})) + \beta(r/n),$$
$$\alpha + \beta = 1, \quad \alpha, \beta \in [0,1],$$
$$\min F(\mathbf{S}),$$

subject to the constraints

$$s_i \in \{0,1\}, \quad i = 1, \ldots, n,$$

where r is the number of features used in the classifier.

To solve this problem, we propose to use a novel population based algorithm with adaptive memory.

4. POPULATION BASED ALGORITHM WITH ADAPTIVE MEMORY

This algorithm is based on the following heuristic: the current value of a component of the solution vector depends on its values at the best solutions obtained at the preceding iterations. Since the vector is binary, it is sufficient to store for each ith component b_i iterations at which the ith component took the value 1; then, the adaptive parameter p of the algorithm computed as the relative frequency of occurrence of the value 1 is

$$p = b_i/t,$$

where t is the number of executed iterations.

The algorithm begins with forming a population—a set of randomly or otherwise generated vectors **S**. The number of vectors in the population is a preassigned integer called the population size. For each vector, the value of the objective function F is computed.

At each iteration, the vector with the minimum value of F is determined—this is the best solution at the current iteration. Another important element of the algorithm is the vector **B** in which each component b_i keeps the number of occurrences of feature i in the best solutions at the preceding iterations; the dimension of this vector coincides with the dimension of **S**. The vector **B**, which serves as adaptive memory, is used to implement the following heuristic: the chance of a feature to get into the next generation of the population is proportional to the frequency with which it occurs in the best preceding solutions. The new generation is formed on the basis of the heuristic mentioned above and random search. The normally distributed random variable $u \sim N(0, \sigma_g)$ determines the mechanism of eliminating or adding features and the number of eliminated or added features. If u is greater than zero, then new features are added by increasing the number of unities in the vector **S**; otherwise, the number of features is reduced. The number of potentially variable features is determined by the dimension of vector **S**, which equals n, and the number of unities (positions at which $s_i \neq 0$) in **S** (it is denoted by r). If u is less than zero, then l candidate features for elimination are randomly chosen among the r unit components of the vector **S** by the formula

$$l = \text{round}(r|\text{th}(u)|).$$

The number of added features is computed by the formula

$$l = \text{round}((n - r)|\text{th}(u)|).$$

However, the change of s_i is determined by the frequency p, and the new value of s_i is given by the formula

$$s_i = \begin{cases} 1 & \text{if } rand(0,1) \leq p \\ 0, & \text{otherwise} \end{cases}.$$

To avoid too early convergence of the algorithm, a constraint on the relative frequency must be added:

$$1 - p_g \leq p \leq p_g,$$

where p_g is a threshold.

The algorithm is executed iteratively; after the execution of a specified number of iterations, the best vector is decoded to obtain the solution.

Below, we give a pseudocode of the algorithm; here *popul* is the population size, T is the maximal number of iterations, p_g is the threshold value, $\mathbf{S}^j$ is the jth solution vector, **Sbest** is the best solution vector, and *Fbest* is the corresponding value of the objective function.

Initialization: $t = 1$, $\mathbf{S}^j = \text{rand}\{0, 1\}^n$ $j = 1, ..., popul$, $\mathbf{B} = \{0.5\}^n$;

Sbest = $\mathbf{S}^1$, *Fbest* = $F(\mathbf{S}^1)$;

Loop on iterations $t = 1, 2, ..., T$

Loop on population $j = 1, 2, ..., popul$

If $F(\mathbf{S}^j) < Fbest$, *then* $Fbest = F(\mathbf{S}^j)$, **Sbest** = $\mathbf{S}^j$;

Assign to r the number of unit components in $\mathbf{S}^j$.

$u \sim N(0, \sigma_g)$.

If $u \geq 0$, then randomly select among these r components round($r|\text{th}(u)|$) components, and assigned to each of these components the value of 1 if rand $< b_k/t$ and 0 otherwise. Here k is the index of the component in the vector $\mathbf{S}^j$.

If $u < 0$, then randomly select among the zero components of $\mathbf{S}^j$ round($(n - r)|\text{th}(u)|$) components. Assign to each of these component the value of 0 if rand $< b_k/t$ and 0 otherwise. Here k is the index of the component in the vector $\mathbf{S}^j$.

End of loop on population j;

B = **B** + **Sbest**;

If $b_k/(t + 1) > p_g$, *then* $b_k = p_g(t + 1)$, $k = 1,2, ..., n$;

If $b_k/(t + 1) < (1 - p_g)$, *then* $b_k = (1 - p_g)(t + 1)$, $k = 1, 2, ..., n$;

End of loop on iterations t;

Output **Sbest**, *Fbest*.

Table 1. Description of data sets

Data sets	Number of features	Number of instances
Breastcancer	9	699
BreastEW	30	569
CongressEW	16	435
Exactly	13	1000
Exactly2	13	1000
HeartEW	13	270
IonosphereEW	34	351
KrvskpEW	36	3196
Lymphography	18	148
M-of-n	13	1000
PenglungEW	325	73
SonarEW	60	208
SpectEW	22	267
Tic-tac-toe	9	958
Vote	16	300
WaveformEW	40	5000
WineEW	13	178
Zoo	16	101

Table 2. Values of the objective function

Набор данных	**ALO**	**GA**	**PSO**	**WOA**	**PAM**
Breastcancer	**0.021**	0.028	0.03	0.035	0.024
BreastEW	0.033	0.036	0.03	0.034	**0.021**
CongressEW	0.046	0.043	0.04	0.047	**0.023**
Exactly	0.289	0.281	0.28	**0.005**	0.036
Exactly2	0.24	0.25	0.25	0.259	**0.238**
HeartEW	**0.122**	0.138	0.15	0.193	0.142
IonosphereEW	0.108	0.125	0.14	**0.076**	0.082
KrvskpEW	0.05	0.068	0.05	**0.027**	0.034
Lymphography	0.136	0.171	0.19	0.148	**0.107**
M-of-n	0.107	0.075	0.11	**0.005**	0.009
PenglungEW	**0.139**	0.22	0.22	0.203	0.215
SonarEW	0.179	0.13	0.13	**0.079**	0.102
SpectEW	0.124	0.137	0.13	0.135	**0.123**
Tic-tac-toe	0.222	0.242	0.24	0.220	**0.08**
Vote	**0.037**	0.054	0.05	0.050	0.038
WaveformEW	0.206	0.203	0.22	0.250	**0.182**
WineEW	0.017	0.014	0.02	0.045	**0.013**
Zoo	0.073	0.082	0.1	0.023	**0.013**

5. EXPERIMENT AND DISCUSSION OF RESULTS

5.1. Description of the Experiment

The experiment and the data sets used in it corresponded to [9]. As the classifier for evaluating the validity of the algorithm, we used the k-nearest neighbors procedure with $k = 5$. A description of the data sets is given in Table 1. The experiment was conducted using the 10-fold cross validation procedure. According to the chosen approach, the number of instances in the training and validation samples was identical, the number of vectors in the population was 15, and the number of iterations was 100.

The results produce by the population based algorithm with memory were compared with the results obtained using other binary feature selection algorithms (they are described in [9]).

5.2. Fitting the Parameters of the Algorithm

The main phase of the experiment was preceded by the phase of the empirical evaluation of the algorithm parameters. The parameter σ_g was assigned the values of 1 and 2; and the parameter p_g was assigned the values of 0.75 and 0.85. Figure 1 shows the values of the objective function averaged over 20 runs of the algorithm as functions of the number of iterations. The algorithm with the parameters $\sigma_g = 2$ and $p_g = 0.85$ showed a small increase in the rate of reaching the optimum. These values of the parameters were used in the experiment.

5.3. Experimental Results

To estimate the effectiveness of the population based algorithm with adapted memory, we compared the results produced by it with the results obtained using the ant lion optimizer (ALO), particle swarm optimizer (PSO), genetic algorithm (GA), and the whale optimization algorithm (WOA) [9]. The coefficients α and β of the objective function were 0.99 and 0.01, respectively. Table 2 shows the mean values of the objective function F. The population based algorithm with adapted memory is denoted by the acronym PAM in this table.

To estimate the statistical significance of differences in the values of the objective function of the classifiers formed by the population based algorithm with adapted memory and by analogous classifiers, we

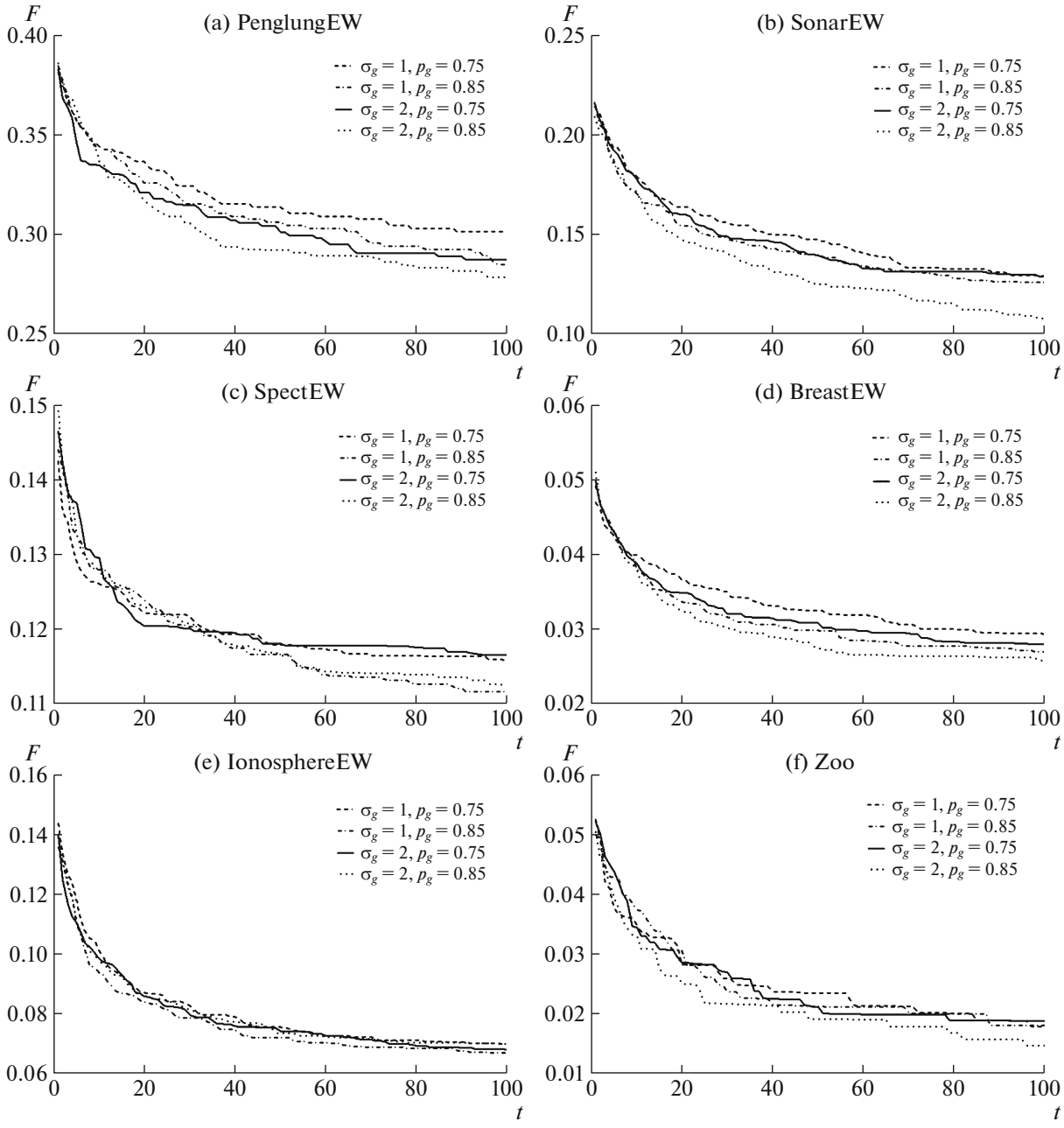

Fig. 1. The mean value of the objective function as a function of the number of iterations.

used the confidence intervals for the difference of means, the Friedman two-factor rank analysis of variance for interrelated samples, and Wilcoxon's signed rank test for related samples.

The hypothesis testing using confidence intervals is based on the following rule: If the 100(1–α)-percent confidence interval of the difference of means does not contain zero, then the differences are statistically significant ($P < \alpha$); otherwise, if this interval contains zero, then the differences are statistically insignificant ($P > \alpha$) [16].

The statistics of 95% confidence intervals for the difference of the values of the objective function are shown in Table 3.

The statistics of Wilcoxon's signed rank test for the medians of differences of the values of the objective function are shown in Table 4. The zero hypothesis H0 is formulated as follows: the median of differences

Table 3. Statistics of confidence intervals for pairwise differences of object function values

Pair	Mean	95% confidence interval for difference		Significance
		Lower	Upper	
PAM–WOA	–0.019556	–0.039032	–0.000079	0.049
PAM–ALO	–0.037056	–0.072797	–0.001314	0.043
PAM–GA	–0.045278	–0.076750	–0.013806	0.007
PAM–PSO	–0.049889	–0.082190	–0.017588	0.005

between pairs equals zero and the significance level is $\alpha = 0.05$.

The comparative analysis suggests the following conclusions:

(1) The 95% confidence interval for the difference of mean values of the objective function does not contain zero; therefore, differences of mean values of the objective function are statistically significant.

(2) The Friedman two-factor rank analysis of variance for interrelated samples testifies to the significant difference in the distributions of five compared values of the objective function (p-value < 0.001).

(3) Wilcoxon's signed rank test for related samples testifies to the significant difference of the values of the objective function (p-value < 0.036).

6. CONCLUSIONS

A novel population based algorithm with adaptive memory for binary optimization is described. This algorithm is applied for the practical problem of feature selection. The *k*-nearest neighbors algorithm is used as the classifier. Eighteen data sets from the collection UCI were used for evaluating the effectiveness of the proposed algorithm. The comparative statistical analysis of this algorithm with other analogous algorithms suggests the conclusion that the population based algorithm with adaptive memory outperforms the other algorithms in terms of the chosen objective function in the feature selection problem.

Table 4. Wilcoxon's test

Pair	Significance	Decision
PAM–WOA	0.035	H0 reject
PAM–ALO	0.012	H0 reject
PAM–GA	<0.001	H0 reject
PAM–PSO	<0.001	H0 reject

In future research, we are going to investigate the effectiveness of the population based algorithm with adaptive memory for feature selection on unbalanced data sets and classifiers of other types.

FUNDING

This work was supported by the Ministry for Science and Education of the Russian Federation, project no. 2.3583.2017/4.6.

REFERENCES

1. Xue, B., Zhang, M., Browne, W.N., and Yao, X., A survey on evolutionary computation approaches to feature selection, *IEEE Trans. Evolutionary Comput.,* 2016, vol. **20**, pp. 606–626.
2. Labati, R.D., Genovese, A., Munoz, E., Piuri, V., and Scotti, F., Applications of computational intelligence in industrial and environmental scenarios, *Studies Comput. Intell.,* 2018, vol. **756**, pp. 29–46.
3. de la Iglesia, B., Evolutionary computation for feature selection in classification problems, *WIREs Data Mining and Knowledge Discovery,* 2013, vol. **3**, pp. 381–407.
4. Kohavi, R. and John, G.H., Wrappers for feature subset selection, *Artif. Intell.,* 1997, vol. **97**, pp. 273–324.
5. Saeys, Y., Inza, I., and Larranaga, P., A review of feature selection techniques in bioinformatics, *Bioinformatics,* 2007, vol. **23**, pp. 2507–2517.
6. Armanfard, N., Reilly, J.P., and Komeili, M., Logistic localized modeling of the sample space for feature selection and classification, *IEEE Trans. Neural Networks Learning Syst.,* 2018, vol. **29**, pp. 1396–1413.
7. Yusta, S.C., Different metaheuristic strategies to solve the feature selection problem, Pattern Recognit. Lett., 2009, vol. **30**, pp. 525–534.
8. Hodashinsky, I.A and Mekh, M.A., Fuzzy Classifier Design Using Harmonic Search Methods, *Program. Comput. Software,* 2017, vol. **43**, no. 1, pp. 37–46.
9. Mafarja, M. and Mirjalili, S., Whale optimization approaches for wrapper feature selection, *Applied Soft Comput.,* 2018, vol. **62**, pp. 441–453.
10. Djellali, H., Djebbar, A., Zine, N.G., and Azizi, N., Hybrid artificial bees colony and particle swarm on feature selection, *Computational Intelligence and Its Appli-*

cations. CIIA 2018, IFIP Advances in Information and Communication Technology, 2018, vol. **522**, pp. 93–105.

11. Glover, F. and Hanafi, S., Tabu search and finite convergence, *Discrete Appl. Math.,* 2002, vol. **119**, pp. 3–36.
12. Riley, R.C.L. and Rego, C., Intensification, diversification, and learning via relaxation adaptive memory programming: A case study on resource constrained project scheduling, J. Heuristics, 2018, pp. 1–15.
13. Omran, M.G.H. and Clerc, M., APS 9: An improved adaptive population-based simplex method for real-world engineering optimization problems, Appl. Intell., 2018, vol. **48**, pp. 1596–1608.
14. Nelder, J. and Mead, R., A simplex method for function minimization, *Comput. J.,* 1965, vol. **7**, pp. 308–313.
15. Saha, S. and Mukherjee, V., A novel chaos-integrated symbiotic organisms search algorithm for global optimization, *Soft Comput.,* 2018, vol. **22**, pp. 3797–3816.
16. Glantz, S.A., *Primer of Biostatistics,* New York: McGraw-Hill, 1994.

Translated by A. Klimontovich