sciendo

JAISCR, 2022, Vol. 12, No. 4, pp. 239 – 253
10.2478/jaiscr-2022-0016

MULTI-POPULATION-BASED ALGORITHM WITH AN EXCHANGE OF TRAINING PLANS BASED ON POPULATION EVALUATION

Krystian Łapa[1,*], Krzysztof Cpałka[1], Marek Kisiel-Dorohinicki[2], Józef Paszkowski[3], Maciej Dębski[4], Van-Hung Le[4]

[1] *Częstochowa University of Technology, Department of Intelligent Computer Systems 42-200 Częstochowa, Poland*

[2] *AGH University of Science and Technology, Institute of Computer Science 30-059 Kraków, Poland*

[3] *Institute of Information Technologies, University of Social Sciences, ul. Sienkiewicza 9, 90-113 Łódź*

[4] *University of Social Science, Management Department 90-113 Łódź, Poland*

[5] *Tan Trao University, Vietnam*

[*] *E-mail: krystian.lapa@pcz.pl*

Submitted: 31st May 2022; Accepted: 14th October 2022

Abstract

Population Based Algorithms (PBAs) are excellent search tools that allow searching space of parameters defined by problems under consideration. They are especially useful when it is difficult to define a differentiable evaluation criterion. This applies, for example, to problems that are a combination of continuous and discrete (combinatorial) problems. In such problems, it is often necessary to select a certain structure of the solution (e.g. a neural network or other systems with a structure usually selected by the trial and error method) and to determine the parameters of such structure. As PBAs have great application possibilities, the aim is to develop more and more effective search formulas used in them. An interesting approach is to use multiple populations and process them with separate PBAs (in a different way). In this paper, we propose a new multi-population-based algorithm with: (a) subpopulation evaluation and (b) replacement of the associated PBAs subpopulation formulas used for their processing. In the simulations, we used a set of typical CEC2013 benchmark functions. The obtained results confirm the validity of the proposed concept.

Keywords: population-based algorithm, multi-population algorithm, hybrid algorithm, island algorithm, subpopulation evaluation, training plan

1 Introduction

Searching the problem domain (search space) can be performed with a gradient or metaheuristic algorithms. Metaheuristics are an important tool for finding a problem's solution when the function used to evaluate the solutions is not differentiable. This is the case with the selection of the solution structure and the parameters of this structure. This applies, for example, to the selection of the structure and parameters of artificial neural networks, fuzzy systems, controllers, dynamic systems, filters, biometric systems, etc. Common metaheuristic algorithms are Population Based Algorithms (PBAs). The use of PBAs does not guarantee finding an optimal solution, but usually allows finding a satisfactory solution for the adopted evaluation function (fitness function). This and other advantages of PBAs make them readily used in practice [7, 10, 33, 35]. Each PBA usually uses a specific search formula. Such a formula takes into account the possibility of searching new areas of the search space (i.e. realizing an exploration) and/or the possibility of searching areas of space around solutions already found (i.e. realizing an exploitation). When searching, the aim is to ensure an appropriate compromise between exploration and exploitation, which is an interesting scientific problem [22, 24].

The popularity of PBAs and their easy application also result in the emergence of many new variations of these methods. However, a disadvantageous phenomenon is giving a new interpretation to previously known search formulas with only their symbolic modification. Undoubtedly, some interesting issues here include creating hybrid PBAs [6, 19, 20], creating multi-population PBAs [3, 8, 9, 14, 15, 17, 18, 22, 24, 28, 36, 37], creating multi-criteria PBAs [14, 15], and developing interesting applications [3, 8, 9, 17, 22, 25, 28, 34], etc.

In this paper, we propose a multi-population-based algorithm with an exchange of training plans based on population evaluation. By a training plan, we understand the formulas of exploration and exploitation of specific PBA and their parameters.

1.1 Related work

There are many interesting issues in the literature regarding multi-population-based algorithms (MPBAs). Some of the new methods from this field will be briefly summarized in this section.

In the paper [36] a multi-population biogeography-based optimization algorithm was proposed. In this algorithm, the whole sorted population is divided into 3 different subgroups. Each such subgroup is processed differently. Subgroups share information by combining individuals and sorting them. Thanks to this approach, the algorithm searched well the space of the considered problem of image segmentation.

In [37] a two-stage cooperative scatter search algorithm with a multi-population hierarchical learning mechanism was introduced. This algorithm is also based on the division of individuals into three groups. Each of these groups uses an interactive individual processing strategy to increase population diversity and avoid premature convergence. The proposed method was tested with the use of the known CEC2013 benchmark functions and selected engineering problems, and the obtained results were satisfactory.

In the paper [8] the authors provide a constrained cooperative adaptive multi-population differential evolutionary algorithm for economic load dispatch problems. This algorithm has several characteristics: (a) uses a hyperspace dynamic constraint handling region between the feasible region and infeasible region, (b) uses two subpopulation generation schemes ("one to one" and "one to many") to improve global solution search capability, and (c) is based on the elimination mechanism through the constraint handling technology (it replaces the selection operation of the differential evolution algorithm). The proposed algorithm was tested using the CEC2013 test functions and selected economic problems, and the thus obtained results were found to be satisfactory.

In the paper [3] an effective multi-population gray wolf optimizer based on reinforcement learning (RL) for a flow shop scheduling problem with multi-machine collaboration was proposed. In this algorithm, the whole population is divided into three subpopulations, and different search strategies are adopted in different subpopulations to enhance population diversity. The RL mechanism is applied to adaptively adjust the individual quantity of each subpopulation and strengthen the information exchange among different subpopulations. The considered algorithm was used to solve the problem of resource composition and task sequencing, and the obtained results were satisfactory.

The authors of the paper [24] introduced a multi-population improved whale optimization algorithm. In this algorithm, individuals are divided into better and worse groups. Better individuals are used to improve the results of search space exploitation, while worse individuals are used for exploration. In addition, several improvements have been made to the algorithm to ensure the effectiveness of the search and to achieve an appropriate compromise between exploration and exploitation. In the simulations, multidimensional problems (ranging from 100 to 2000 dimensions) were considered, and satisfactory results were obtained.

In the paper [17] a hybrid multi-population metaheuristic applied to load-sharing optimization of gas compressor stations was proposed. The approach under consideration combines the diversification capability of the crow search algorithm and the intensification capability of the symbiotic organisms search. Moreover, it uses two subpopulations, each of which is processed with a different intensity (as in the algorithms described earlier). The results obtained in the simulations performed were satisfactory.

In the paper [18] the authors propose a multi-population-based adaptive sine cosine algorithm with a modified mutualism strategy for global optimization. A feature characteristic of this algorithm is that the population is halved. The resulting subpopulations are processed with the sine or cosine strategy. The CEC2013 function was used in the simulations, which produced satisfactory results.

In the article [9] a multi-population-based particle swarm optimization for feature selection was presented. In this method, multi-population start with initial solutions generated by random and relief-based initialization and searches solution space simultaneously using both populations. In the simulations performed, several dozen known problems in the field of data selection (UCI and ASU) were used, and satisfactory results were obtained.

In the paper [22] a multi-population parallel co-evolutionary differential evolution for optimizing the parameters of a photovoltaic system was proposed. This algorithm uses a reverse learning mechanism to generate initial subpopulations and uses dedicated mechanisms for parallel management of subpopulations. These mechanisms are based on the use of different mutation strategies to ensure: (a) a trade-off between exploration and exploitation, and (b) the implementation of an appropriate migration strategy. The obtained results were found to be satisfactory.

In the article [14] a grid search-based multi-population particle swarm optimization algorithm for multimodal multi-objective optimization was proposed. This approach is based on the k-means clustering method to detect equivalent Pareto sets. Popular benchmark functions were used in the simulations and satisfactory results were obtained.

In [15] its authors proposed a tri-population-based co-evolutionary framework for constrained multi-objective optimization problems. This approach assumes that there are three subpopulations processed with different intensities and different congestion processing approaches. In the simulations, typical benchmark functions for the problem under consideration were used and satisfactory results were obtained.

The authors of [28] proposed a multi-population competitive-cooperative GWO for scheduling field service resources in cloud manufacturing. The use of this algorithm has proved to be a perfect solution for the dynamic and flexible allocation of geographically dispersed production resources to perform specific production tasks (such as assembly, measurement, and maintenance of large devices). The obtained results proved satisfactory.

The issues discussed in this paper fit perfectly with the subject of the latest papers on MPBAs. Moreover, they have not yet been considered in the literature.

1.2 Motivation

In our earlier work, we proposed single-population PBAs that use multiple effective search formulas. These include the following methods: OPn [11], OP11 [13], and OP1 [12]. The operators used in their search formulas were derived from PBAs known in the literature and changed dynamically throughout the search process. When these methods were tested, it turned out that:

– using many different operators to search the problem domain gave better results than using a static operator or operators, which is typical of most PBAs. The use of multiple dynamically assigned operators is advantageous because it reduces the problem of selecting a single algorithm to find a solution;

– the relationship between the course of the search and the types of search operators used was not found and it was not repeatable. The set of operators used changed dynamically during the algorithm's operation. This demonstrates a flexible use of exploration and exploitation mechanisms.

In this paper, we also consider the use of multiple search operators in the context of MPBAs. However, we assumed that not single individuals, but entire subpopulations of the algorithm are associated with different search formulas. We believe that such a solution will favorably reduce the dynamics of operator exchange that occurred in the single-population algorithms proposed earlier [11, 12, 13]. Thanks to this, the capabilities of the search operators can be used better, which was difficult when their exchange was extensive/intensive.

An important issue considered in this paper is also the possibility of evaluating subpopulations and their processing formulas. We believe that the use of this option will provide interesting opportunities, therefore we propose original solutions in this field.

1.3 Contribution of the paper

In this paper, we propose a multi-population-based algorithm with the exchange of training plans based on population evaluation. Its characteristics can be summarized as follows:

– it uses many subpopulations, each of which is processed with different search operators having their own parameter values. The algorithm and its parameters are called a training plan. In the island algorithms considered in the literature, subpopulations are usually processed by a single PBA (e.g. for a genetic algorithm, the processing formula is a crossover and mutation type). In our earlier work [21] we considered the possibility of processing subpopulations with separate PBAs, which along with parameters correspond to training plans. However, in that algorithm, the training plans did not change and were not evaluated;

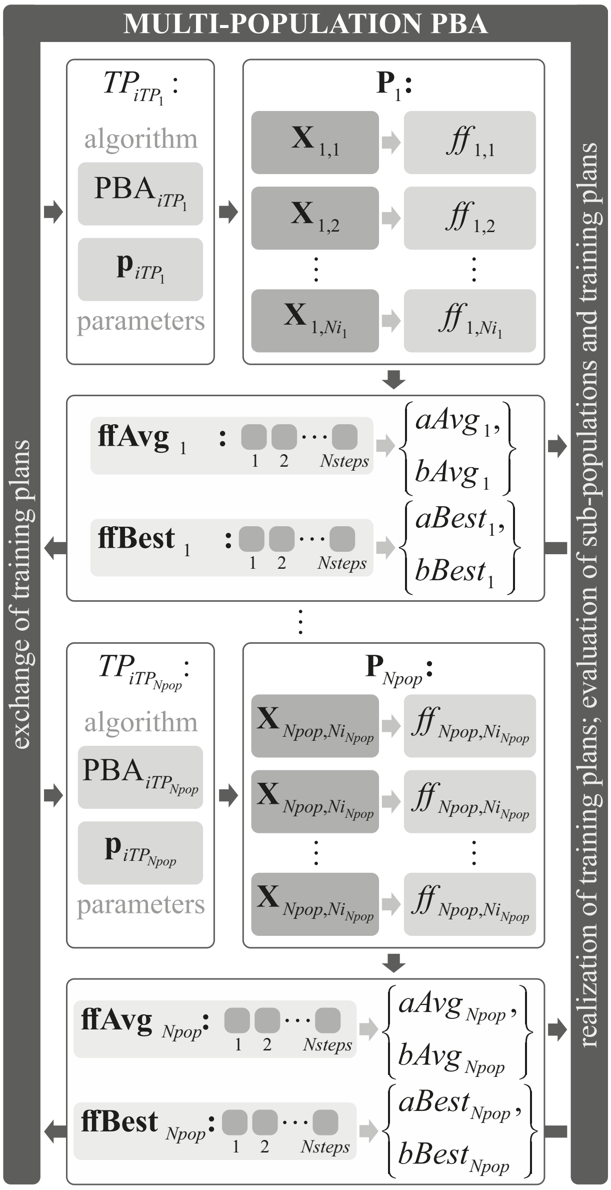

Figure 1. The idea of the multi-population-based algorithm with the exchange of training plans based on population evaluation proposed in this paper.

– it implements the exchange of training plans. This exchange depends on the evaluation of the training plan and the evaluation of the subpopulation. The scoring takes into account the evaluation function value of the best individual of the subpopulation and the mean evaluation function values of all individuals in the subpopulation. In addition, these evaluations will take into account the values from the current step and several previous steps. On their basis, the coefficients of linear trends are determined which are included in the exchange of plans. The use of linear trends is to eliminate the influence of random fluctuations of the evaluation function values on the course of the exchange.

1.4 Structure of the paper

In Section 2 we have described the multi-population-based algorithm with the exchange of training plans based on population evaluation proposed in this paper. In Section 3 we have included the obtained results and conclusions from the simulations. Finally, in Section 4 a summary and future research plans are provided.

2 Description of the proposed algorithm

This Section describes the multi-population-based algorithm with the exchange of training plans based on the population evaluation proposed in this paper. Its idea is presented in Listing 1 and shown in Figure 1. The steps of this algorithm can be summarized as follows:

- In Step 1, the following parameters are set: the number of individuals ($Nind$), the number of subpopulations ($Npop$), the number of steps followed by a change of training plans ($Nsteps$), and the number of subpopulations for which training plans are exchanged ($Nrep$).
- In Step 2, $Nind$ individuals are randomly allocated to the $Npop$ subpopulations. It was assumed that in each subpopulation $\mathbf{P}_p$ there may be a different number of individuals.
- In Step 3, training plans TP_p, $p = 1.2,\ldots,Npop$ are initialized. A single training plan consists of a PBA with individual parameters. For convenience, the number of training plans is assumed to be equal to the number of subpopulations, and each training plan is used to process one subpopulation (this assumption can be changed). This assumption no longer applies when exchanging training plans.
- In Step 4, a training plan is assigned to each subpopulation $\mathbf{P}_p$ from a pool of initialized training plans. This allocation is random and without repetition. The selected training plan for the p subpopulation is indicated by index $iTP_p \in \{1,2,\ldots,Npop\}$.
- In Steps 5-9, the evaluation of individuals in subpopulations is carried out. All individuals of the subpopulation are subject to evaluation (Step 6, see Section 2.1). Then, the best individual $\mathbf{X}_p^*$ is selected for each subpopulation and its evaluation function value ff_p^* is stored (Step 7).
- In Step 9, the auxiliary variable $step$ is initialized, which is the count of the algorithm steps performed. Its value is used in Step 11 and updated in Step 32.
- In Steps 10-33, the essential part of the algorithm is executed. If the stop condition in Step 10 is satisfied, the algorithm proceeds to the execution of Step 34. In step 34 the best individual **XBest** is presented and the algorithm ends its operation.
- In Step 11, the r index is determined. It is used in indexing auxiliary arrays $\mathbf{ffAvg}_p$ and $\mathbf{ffBest}_p$. These arrays hold the mean values of the evaluation function ($\mathbf{ffAvg}_p$) and the best values of the evaluation function ($\mathbf{ffBest}_p$) of the $\mathbf{P}_p$ subpopulation. These tables are used in determining the parameters of the linear trends (Steps 19 and 20) for each subpopulation. The formula for determining the coefficients of the line describing the linear trend is well described in the literature and hence is not discussed in this paper (see e.g. [27]).
- Each subpopulation is processed in Steps 12-24. In Step 13, the training plan assigned to the subpopulation is executed at the index $iTP_p \in \{1,2,\ldots,Npop\}$. This is due to the use of appropriate search operators with their parameters specified in the training plan. The search operators are typical of the PBA indicated in the training plan.

Algorithm 1 The idea of the multi-population-based algorithm with the exchange of training plans based on population evaluation proposed in this paper.

1: initialize $Nind$, $Npop$, $Nsteps$, $Nrep$ parameters
2: initialize randomly $Nind$ individuals and assign them to $Npop$ subpopulations $\mathbf{P}_p$ (each $\mathbf{P}_p$ can have Ni_p individuals)
3: initialize $Npop$ training plans TP_p ($p = 1, 2, \ldots, Npop$), each of which is composed of the PBA and its corresponding parameters
4: assign randomly with no repetition a training plan to each $\mathbf{P}_p$; the selected plan is indicated by index $iTP_p \in \{1, 2, \ldots, Npop\}$
5: **for** $p := 1$ **to** $Npop$ **do**
6: evaluate individuals $\mathbf{X}_{p,i}$, $i = 1, 2, \ldots, Ni_p$, in $\mathbf{P}_p$
7: select best individual $\mathbf{X}_p^*$ with fitness value ff_p^* from $\mathbf{P}_p$
8: **end for** p
9: set $step := 1$
10: **while** !(stop condition) **do**
11: calculate index r: $r := 1 + (step - 1)\,\%Nsteps$
12: **for** $p := 1$ **to** $Npop$ **do**
13: follow the training plan indicated by iTP_p
14: evaluate updated individuals $\mathbf{X}_{p,i}$ in $\mathbf{P}_p$
15: select best individual $\mathbf{X}_p^*$ with fitness value ff_p^* from $\mathbf{P}_p$
16: determine the mean fitness function in $\mathbf{P}_p$: $\overline{ff_p}$
17: calculate: $ffAvg_p[r] := \overline{ff_p}$ and $ffBest_p[r] := ff_p^*$
18: **if** $(r == Nsteps)$ **then**
19: using $\mathbf{ffAvg}_p$ determine $\{aAvg_p, bAvg_p\}$
20: using $\mathbf{ffBest}_p$ determine $\{aBest_p, bBest_p\}$
21: using $\{aAvg_p, bAvg_p\}$ and $\{aBest_p, bBest_p\}$ evaluate $\mathbf{P}_p$
22: using $\{aAvg_p, bAvg_p\}$ and $\{aBest_p, bBest_p\}$ evaluate the training plan indicated by iTP_p; if plan iTP_p was evaluated before then average its values
23: **end if**
24: **end for** p
25: select best individual from $\left\{\mathbf{X}_1^*, \mathbf{X}_2^*, \ldots, \mathbf{X}_{Npop}^*\right\}$; if $step == 1$ or the fitness function value of the selected individual is better than fitness function value $ffBest$ of **XBest**, then update **XBest** and $ffBest$
26: exchange information between subpopulations according to the adopted migration plan
27: **if** $(r == Nsteps)$ **then**
28: choose randomly using the roulette wheel method $Nrep$ subpopulations to change their training plan
29: choose randomly using the roulette wheel $Nrep$ training plans to assign them to subpopulations (the pool of plans is not changed)
30: associate randomly selected training plans with selected subpopulations
31: **end if**
32: update $step := step + 1$
33: **end while**
34: return **XBest**

– Steps 14-15 are similar to Steps 6-7. Additionally, Step 16 is performed where an average evaluation function value is determined for each $\overline{ff_p}$ subpopulation. In Step 17 the best individual fitness function value of the $fBest_p[r]$ subpopulation and the mean fitness function value of subpopulation $ffAvg_p[r]$ are stored.

– In Steps 18-23, the activities related to the determination of linear trends in the subpopulation (Steps 19-20) and the activities related to the assessment of: (a) the subpopulation (Step 21, see Section 2.2) and (b) the subpopulation training plan (Step 22, see Section 2.2).

– In Step 25, the best individual **XBest** is selected from the set of the best individuals of the subpopulations $\left\{\mathbf{X}_1^*, \mathbf{X}_2^*, \ldots, \mathbf{X}_{Npop}^*\right\}$. The selection criterion is the value of the evaluation function. If: (a) this is the first step of the algorithm ($step == 1$) or (b) the evaluation function of the selected individual is better than the $ffBest$ evaluation function of the individual **XBest**, then update **XBest** and $ffBest$.

– In Step 26, individuals are exchanged between the subpopulations according to the adopted migration plan. In the simulations, the migration strategy summarized in Table 1 was adopted.

– In Steps 27-31, the selection of subpopulations (Step 28, for details see Section 2.2) to exchange their training plan and selection of training plans (Step 29, for details see Section 2.2) are conducted. After performing Steps 28 and 29, the selected training plans are associated with the selected subpopulations (Step 30).

The remainder of this Section covers the subject of encoding and evaluating individuals, and evaluating both the subpopulations and training plans.

2.1 Encoding and evaluation of individuals

The algorithm considered in this paper is presented in a basic version and tested with the use of simulation problems that do not require a specific approach to encoding solutions and their evaluation. Therefore, these issues will not be considered in detail. However, the algorithm can be easily adapted, for example, to solving hybrid-type problems (consisting in finding the solution structure and its parameters), for which a complex evaluation function should additionally be designed. Then, the method of encoding solutions and their evaluation may be analogous to that presented in the papers [11, 21].

2.2 Subpopulations and training plans evaluations

The proposed algorithm introduces the possibility of evaluating subpopulations and training plans. The subpopulation evaluation facilitates the marking of the subpopulations made up of individuals with the worst fitness function values, e.g. to change their training plan. On the other hand, the evaluation of training plans facilitates the selection of those training plans that allowed to efficiently process subpopulations in the previous steps of the algorithm.

In Step 21, the $\mathbf{P}_p$ subpopulation is evaluated according to the weighted average formula:

$$ffP_p = \frac{\left(\overline{ff_p} \cdot wAvg + ff_p^* \cdot wBest\right)}{wAvg + wBest}, \quad (1)$$

where $\overline{ff_p} \in [0,1]$ is a current normalized average value of the fitness function for subpopulation $\mathbf{P}_p$, $ff_p^* \in [0,1]$ is a current normalized fitness function value of the best individual within subpopulation $\mathbf{P}_p$, $wAvg, wBest \in [0,1]$ are weights of function ffP_p (wherein $wAvg = 1 - wBest$). Therefore, the function defined by equation (1) took the simplest possible form; however, it may also take into account the course of linear trends associated with the subpopulation $\mathbf{P}_p$ - however, this is not considered in this paper.

In Step 22, an evaluation of the training plan assigned to the $\mathbf{P}_p$ subpopulation is performed. It is implemented as follows:

$$ffTP_p = \frac{\left(\begin{array}{c}\left(\frac{\arctan(aAvg_p)}{\pi} + 0.5\right) \cdot wAvg + \\ + \left(\frac{\arctan(aBest_p)}{\pi} + 0.5\right) \cdot wBest\end{array}\right)}{wAvg + wBest}, \quad (2)$$

where $aAvg_p$ is the slope of the linear trend associated with the mean evaluation function of subpopulation $\mathbf{P}_p$ and designated based on $Nsteps$ steps, $aBest_p$ is the slope of the linear trend associated with the best subpopulation $\mathbf{P}_p$ evaluation function and also designated based on $Nsteps$ steps.

The functions in the forms of (1) and (2) return values in the range $[0,1]$. This facilitates the use of the roulette wheel method in Steps 28 and 29. Expressed in degrees and associated with the subpopulation $\mathbf{P}_p$, circle clippings $\{csTP_p, csP_p\}$ can be determined as follows:

$$csTP_p = \frac{ffTP_p}{\sum_{q=1}^{Npop} ffTP_p} \cdot 360^\circ \quad (3)$$

and

$$csP_p = \frac{ffP_p}{\sum_{q=1}^{Npop} ffP_p} \cdot 360^\circ. \quad (4)$$

Table 1. The parameters used in the simulations of the proposed multi-population-based algorithm.

no.	parameter name	parameter interpretation
1.	$Npop = 8$	number of subpopulations
2.	$Nrep = 4$	number of subpopulations, for which training plans are exchanged
3.	$Nind = 32$	number of individuals in each population
4.	$Nsteps = 10$	number of steps after which training plans are exchanged
5.	$Nmig = \{1,4,10,20\}$	number of steps after which individuals are migrated
6.	migration topology: {STAR1D, STARWR, RSTARB, LAD2LP, RANDM2}	it determines the way subpopulations cooperate, the details can be found in [5], the best topologies were selected
7.	migration strategy: {BEST, RWHEEL, TRNT, RAND}	it determines the way of individual selection for subpopulations migration, the details can be found in [5]
8.	$wAvg = 0.5$	weight of the component $\mathbf{ffAvg}_p$
9.	$wBest = 0.5$	weight of the component $\mathbf{ffBest}_p$
10.	exchange of plans: {YES, NO}	if the value is equal to NO, the used plans are static

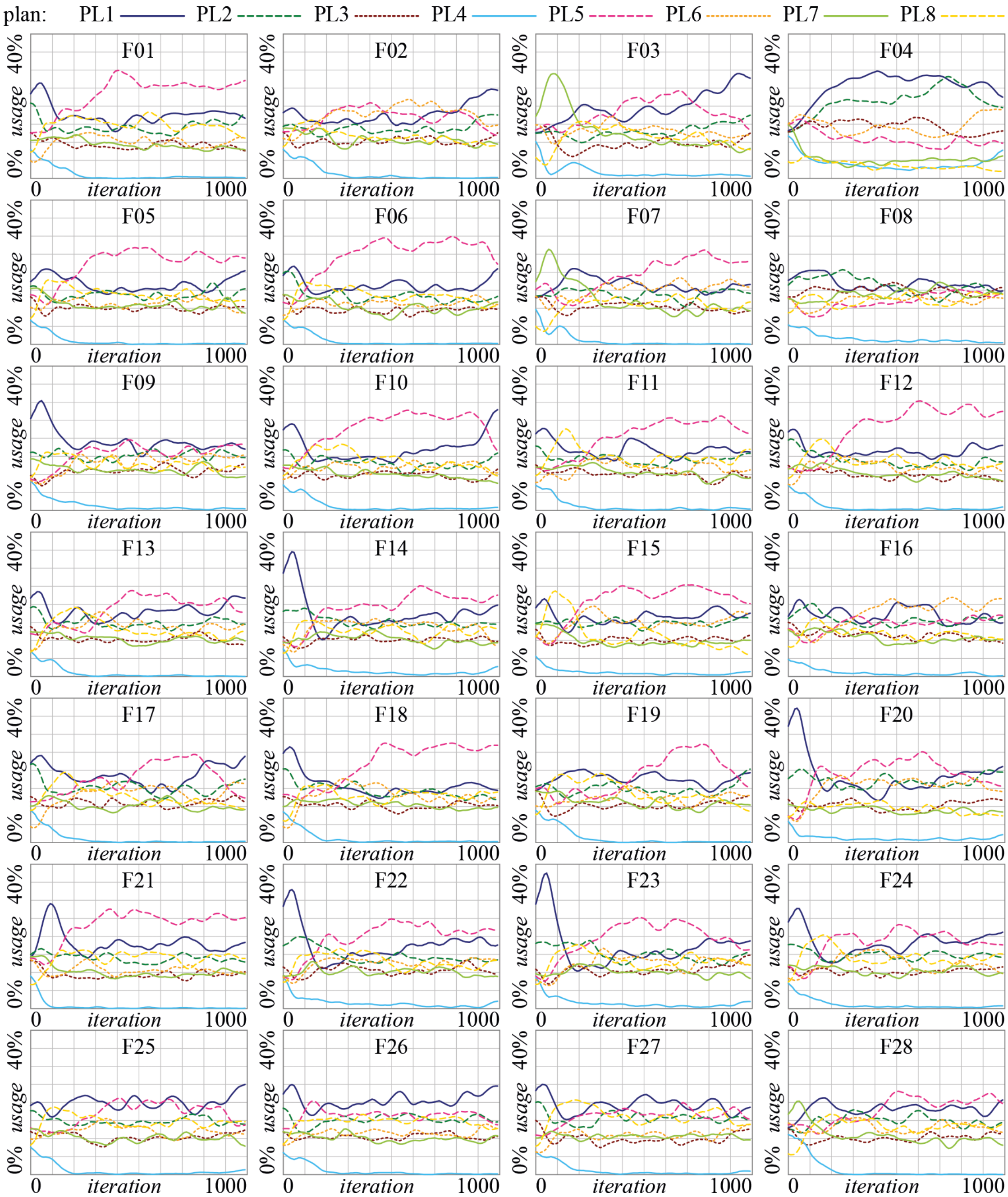

Figure 2. The course of using training plans in the individual steps of the proposed algorithm, averaged for all benchmark functions.

Table 2. Training plans used in the simulations.

no.	plan name	PBA	PBA parameters	plan purpose
1.	GA1	GA	$p_c = 0.9$, $m_c = 0.5$, $m_r = 0.1$	exploitation
2.	GA2	GA	$p_c = 0.7$, $m_c = 0.7$, $m_r = 0.2$	universal
3.	GA3	GA	$p_c = 0.5$, $m_c = 0.9$, $m_r = 0.3$	exploration
4.	GW1	GWO	-	universal
5.	DE1	DE	$F = 0.50$, $CR = 0.95$	universal
6.	DE2	DE	$F = 0.75$, $CR = 0.85$	exploration
7.	CS1	CS	$PA = 0.1$	universal
8.	CS2	CS	$PA = 0.4$	diversification

3 Simulations

The multi-population-based algorithm with the exchange of training plans based on population evaluation proposed in this paper was tested using the CEC2013 benchmark functions (improved CEC2005 benchmark set as used in [26]), hereinafter referred to as F01-F28 [23]. The other comments regarding the simulations can be summarized as follows:

– Each simulation was repeated 50 times, and the results given in the paper are average results. The settings and parameters of the proposed multi-population-based algorithm used in the simulations are presented in Table 1.

– The training plans used in the simulations are shown in Table 2. The course of using the training plans in the individual steps of the search for a solution for the best simulation case is shown in Figure 2.

– The tested simulation variants are combinations of settings presented in Table 1 and they can take the following name in full variant description: $A\!-\!B\!-\!C\!-\!D$, where A stands for $Nsteps$, B stands for migration topology, C stands for migration strategy and D stands for dynamic exchange of plans. Thus, the total number of simulation variants is equal to $4 \cdot 5 \cdot 4 \cdot 2 = 160$. The obtained results considered for individual benchmark functions are shown in Tables 3 and Table 4, and the obtained results averaged for individual benchmark functions are shown in Table 5. The comparison of the results obtained using the multi-population algorithm proposed in this article with the evaluation and exchange of the training plans with the results obtained using other methods is shown in Table 6. For other population-based algorithms, migration strategies and typologies were also used. The table presents the results only for the best variants of these cases (analogous to the best variant of the proposed method).

The simulation conclusions can be summarized as follows:

– The multi-population-based algorithm proposed in this paper works as expected (Tables 3 and 5). The possibility of evaluating and exchanging training plans increased the effectiveness of searching the problem domain and allowed us to obtain very good results in comparison with the results for other methods (Table 6).

– The variation in the use of training plans is high during the operation of the algorithm (Figure 2). It follows that the algorithm alternates between exploration and exploitation mechanisms. This means that in practice it would be difficult to indicate a clear principle of switching between exploitation and exploration.

– The best results were obtained with migrations performed every 10 and 20 iterations (see $Nmig = \{10, 20\}$ in Table 5). With such variants, the use of dynamic plan exchange improved the results in 82.5% of cases and allowed to obtain the best results (see 20-STARWR-RWHEEL-

Table 3. The obtained normalized results for individual benchmark functions averaged for migration strategies and migration topologies. $Nmig = 0$ stands for a case with no migration.

$Nmig \rightarrow$	0		1		4		10		20	
exchange of plans $\rightarrow$	YES	NO	YES	NO	YES	NO	YES	NO	YES	NO
F01	0.1245	0.2510	0.0239	0.0141	0.0212	0.0592	0.0265	**0.0081**	0.0152	0.0137
F02	0.8294	0.7883	0.2774	0.4395	**0.2683**	0.4917	0.3025	0.2694	0.3409	0.3147
F03	0.2374	0.3862	0.2063	0.1677	0.1874	**0.1553**	0.1756	0.1838	0.1920	0.1715
F04	0.5754	0.6228	**0.1372**	0.2935	0.1541	0.3458	0.1978	0.1879	0.2325	0.2185
F05	0.1828	0.2842	0.0206	0.0051	0.0102	0.0543	0.0113	0.0080	0.0195	**0.0035**
F06	**0.0572**	0.0605	0.3954	0.1939	0.3227	0.2018	0.2550	0.3653	0.2305	0.2428
F07	0.6774	0.6633	0.6979	0.3763	0.6254	**0.3409**	0.6174	0.5835	0.5404	0.4608
F08	0.3163	0.3225	0.6064	0.5920	0.6341	0.5596	0.5420	0.5644	0.6441	0.5808
F09	**0.1917**	0.2023	0.6067	0.3232	0.5121	0.3192	0.3886	0.4637	0.3271	0.3480
F10	0.7560	0.7432	0.2746	0.2633	0.2378	0.3463	0.2455	**0.1754**	0.2680	0.1788
F11	0.5653	0.5506	0.5362	0.2770	0.4522	**0.2240**	0.3291	0.4762	0.2584	0.3260
F12	0.5349	0.5608	0.4609	0.3359	0.3604	**0.3346**	0.3419	0.4883	0.3434	0.4239
F13	0.5534	0.5659	0.5412	0.2742	0.4628	**0.2285**	0.3444	0.5012	0.2931	0.3380
F14	0.7285	0.6528	0.6378	0.6672	0.5906	0.6368	0.5017	0.6555	**0.4130**	0.6342
F15	**0.3211**	0.3312	0.6173	0.4624	0.5192	0.4030	0.4890	0.5468	0.4516	0.4198
F16	**0.0866**	0.1297	0.5072	0.4475	0.3998	0.4795	0.2613	0.5833	0.2311	0.4059
F17	0.4916	0.5256	0.5678	0.4387	0.5203	0.3890	0.4729	0.6123	**0.3659**	0.4889
F18	0.6539	0.6736	0.4043	0.3449	0.3022	0.4024	0.2411	0.4112	**0.2405**	0.3342
F19	0.2753	0.2412	0.5966	0.1591	0.4537	**0.1183**	0.3437	0.4357	0.2848	0.2751
F20	0.8356	0.7921	0.4555	0.6696	0.4292	0.7570	0.4122	0.5495	**0.3783**	0.5618
F21	0.1218	0.2954	0.0317	0.0183	0.0173	0.0567	0.0221	**0.0093**	0.0465	0.0128
F22	0.5941	0.5730	0.6239	0.6180	0.5324	0.6799	0.4453	0.6322	**0.3398**	0.6696
F23	0.6553	0.6043	0.5600	0.6271	0.4801	0.6077	0.3928	0.5704	**0.3297**	0.6174
F24	0.9620	0.8934	0.6438	0.5285	0.5771	0.5126	0.4760	0.6258	**0.4071**	0.6161
F25	0.5653	0.5409	0.5067	0.2438	0.4416	**0.2172**	0.3548	0.4457	0.2797	0.3304
F26	0.4113	0.4603	0.4757	**0.1012**	0.3627	0.1346	0.2443	0.2993	0.1918	0.1611
F27	0.6877	0.6505	0.6059	**0.1501**	0.4086	0.1531	0.3583	0.4189	0.2591	0.2020
F28	0.5326	0.5657	0.5776	0.3295	0.5154	**0.2377**	0.3909	0.4817	0.3086	0.3447
AVG	0.4830	0.4975	0.4499	0.3343	0.3857	0.3374	0.3280	0.4126	**0.2940**	0.3462

YES and 20-LAD2LP-RWHEEL-YES in Table 5).

– Too frequent migration of individuals between populations gave the worst results (see $Nmig = \{1,4\}$ in Table 5). In this case, also the use of dynamic exchange of plans did not often bring improvement.

– For different benchmark functions, different approaches gave the best results, but on average, the best results were achieved when migrating 20 iterations and using a plan exchange (see Table 3). Even more effective use of a training plan exchange can be seen in the best case of the migration strategy (see Table 4).

– The strategy of exchanging plans can dynamically adapt to a given problem (although some dependencies are apparent for most problems) and the learning process itself (other plans are more often used in various stages of optimization), which can be seen in Figure 2.

Table 4. The obtained normalized results for individual benchmark functions for the best simulation case (migration topology=STARWR, migration strategy=RWHEEL).

Nmig →	0		1		4		10		20	
exchange of plans →	YES	NO	YES	NO	YES	NO	YES	NO	YES	NO
F01	0.1245	0.2510	**0.0000**	0.0495	**0.0000**	0.0186	**0.0000**	**0.0000**	**0.0000**	0.0004
F02	0.8294	0.7883	0.1413	0.8456	0.1672	0.7700	0.1538	**0.0509**	0.2901	0.1491
F03	0.2374	0.3862	0.2354	0.5592	0.1572	0.3557	0.1270	**0.0317**	0.0665	0.0979
F04	0.5754	0.6228	**0.0084**	0.5400	0.0537	0.5180	0.1209	0.0413	0.1885	0.1155
F05	0.1828	0.2842	**0.0001**	0.0094	**0.0001**	0.0086	**0.0001**	0.0002	0.0015	0.0013
F06	**0.0572**	0.0605	0.7494	0.0833	0.3437	0.0881	0.0704	0.4919	0.5234	0.2280
F07	0.6774	0.6633	0.7132	0.4507	0.5917	0.5805	0.5984	0.3959	0.5236	**0.2079**
F08	**0.3163**	0.3225	0.6317	0.5653	0.8242	0.4459	0.5985	0.4283	0.6023	0.5745
F09	0.1917	0.2023	0.8318	0.3094	0.5018	0.2257	0.4498	0.2479	0.3653	**0.1128**
F10	0.7560	0.7432	0.0723	0.5929	0.0611	0.6633	0.0265	**0.0246**	0.0480	0.1033
F11	0.5653	0.5506	0.6840	0.4151	0.5273	0.3204	0.2427	0.4782	**0.0000**	0.2517
F12	0.5349	0.5608	0.6309	0.5460	**0.3217**	0.4091	0.3655	0.5647	0.4730	0.4036
F13	0.5534	0.5659	0.6174	0.4629	0.6303	0.5666	0.2282	0.2348	**0.0403**	0.1920
F14	0.7285	0.6528	0.6727	0.6998	0.5510	0.8235	0.4920	0.5238	**0.3325**	0.7845
F15	0.3211	0.3312	1.0000	0.4247	0.3800	0.3058	0.4456	0.7086	0.3638	**0.1974**
F16	**0.0866**	0.1297	0.8928	0.8284	0.7609	0.6036	0.2368	0.6898	0.1792	0.6577
F17	0.4916	0.5256	0.5443	0.4419	0.4272	0.5375	0.5387	0.5625	**0.1533**	0.4042
F18	0.6539	0.6736	0.3788	0.7113	0.0977	0.7451	**0.0713**	0.3461	0.1022	0.4226
F19	0.2753	0.2412	0.5784	0.1232	0.4144	0.1383	0.2670	0.3069	**0.0846**	0.2403
F20	0.8356	0.7921	0.3470	0.9226	0.2857	0.9246	**0.2316**	0.8062	0.2988	0.9428
F21	0.1218	0.2954	**0.0000**	0.1346	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**	**0.0000**
F22	0.5941	0.5730	0.6560	0.7403	0.4189	0.8772	0.2968	0.4678	**0.2077**	0.6043
F23	0.6553	0.6043	0.6222	0.7709	0.4551	0.8233	0.2955	0.3693	**0.0540**	0.7071
F24	0.9620	0.8934	0.7899	0.6825	0.6361	0.6838	0.3048	0.5756	**0.2357**	0.5082
F25	0.5653	0.5409	0.6493	0.2936	0.4707	0.3504	0.2277	0.4833	**0.2056**	0.2435
F26	0.4113	0.4603	0.5061	0.1349	0.4043	0.2322	0.0817	0.3141	**0.0220**	0.0930
F27	0.6877	0.6505	0.6363	0.1422	0.3899	0.2132	0.2596	0.3940	0.1251	**0.0528**
F28	0.5326	0.5657	0.7769	0.4461	0.5369	0.5386	0.3912	0.4599	**0.0240**	0.2043
AVG	0.4830	0.4975	0.5131	0.4616	0.3717	0.4560	0.2544	0.3571	**0.1968**	0.3036

4 Conclusions

In this paper, a multi-population algorithm with the exchange of training plans based on population evaluation was proposed. It works as expected and the accuracy achieved is satisfactory. The advantage of this algorithm was evident in the considered problems. The advantage of the proposed algorithm is primarily due to the possibility of evaluating the subpopulation and related training plans. In this evaluation, linear trends of the components of the evaluation function were used. The necessity to table these components and the additional computational effort associated with their processing is a certain drawback of the algorithm. It seems, however, that this disadvantage is not important when compared to the advantages offered by such a solution. The proposed mechanism was initially used by us to exchange training plans for the subpopulation, but the possibilities of its use seem to be wider.

Our future plans include: controlling the size of the subpopulation based on its evaluation, controlling the number of subpopulations based on its evaluation, and adjusting parameters included in the training plan. In addition, we plan to apply a multi-population-based algorithm with the evaluation and exchange of training plans to solve problems related to the selection of a problem structure and its parameters. In particular, we plan to investigate the possibilities of application of the method proposed in this paper for the selection of the structure and parameters of artificial neural networks [2, 4, 16], fuzzy systems [1] and different types of biometric systems [29, 30, 31, 32].

Table 5. The obtained normalized results averaged for individual benchmark functions. The results below the value of 0.2 are marked in bold, and the best results are underlined.

Nmig	migration strategy→	BEST		RWHEEL		TRNT		RAND	
	migration topology ↓	exchange of plans ↓		exchange of plans ↓		exchange of plans ↓		exchange of plans ↓	
		YES	NO	YES	NO	YES	NO	YES	NO
0	NOMIGR	0.4830	0.4951	-	-	-	-	-	-
1	STAR1D	0.4426	0.5066	0.4802	0.4525	0.4517	0.4580	0.4549	0.4818
	STARWR	0.5131	0.4616	0.4034	0.3078	0.4199	0.3651	0.4324	0.3358
	RSTARB	0.6396	0.3712	0.5230	0.3502	0.5717	0.3462	0.5227	0.2888
	LAD2LP	0.5778	0.3182	0.4091	0.2391	0.4182	0.2512	0.3625	0.2141
	RANDM2	0.4151	0.2085	0.3383	0.2277	0.3381	0.2660	0.2830	0.2362
4	STAR1D	0.4649	0.7141	0.4793	0.4646	0.4684	0.4266	0.4120	0.4576
	STARWR	0.3717	0.4560	0.3538	0.3241	0.3510	0.3880	0.3108	0.3344
	RSTARB	0.5398	0.3806	0.4440	0.2756	0.4803	0.2739	0.4659	0.2881
	LAD2LP	0.4287	0.2861	0.3114	0.2117	0.3707	0.2401	0.3279	0.2294
	RANDM2	0.3270	0.2425	0.2540	0.2527	0.2913	0.2469	0.2604	0.2546
10	STAR1D	0.4414	0.4612	0.4663	0.4512	0.4496	0.4703	0.4167	0.4586
	STARWR	0.2544	0.3571	0.2178	0.3431	0.2547	0.3041	0.2188	0.3286
	RSTARB	0.4731	0.5993	0.4421	0.4373	0.4452	0.5188	0.4489	0.4422
	LAD2LP	0.3231	0.5575	0.2572	0.3731	0.2559	0.4073	0.2274	0.3156
	RANDM2	0.2526	0.4736	0.2560	0.3191	0.2562	0.3431	0.2027	0.2908
20	STAR1D	0.4495	0.4825	0.4918	0.4459	0.4688	0.4826	0.4559	0.4119
	STARWR	**0.1968**	0.3036	<u>**0.1661**</u>	0.3391	0.2040	0.3042	0.2044	0.3386
	RSTARB	0.4221	0.4300	0.3906	0.3556	0.3775	0.3929	0.3720	0.3320
	LAD2LP	0.2168	0.3555	**0.1701**	0.3124	0.2263	0.2649	**0.1880**	0.2504
	RANDM2	0.2521	0.3225	0.2066	0.2479	0.2154	0.2593	0.2052	0.2930

Acknowledgment

The project financed under the program of the Polish Minister of Science and Higher Education under the name "Regional Initiative of Excellence" in the years 2019-2023 project number 020/RID/2018/19 the amount of financing PLN 12,000,000.

References

[1] Ł. Bartczuk, A. Przybył, K. Cpałka, A new approach to nonlinear modelling of dynamic systems based on fuzzy rules, International Journal of Applied Mathematics and Computer Science (AMCS), 26(3), 603-621, 2016.

[2] J. Bilski, B. Kowalczyk, A. Marchlewska, J.M. Żurada, Local Levenberg-Marquardt Algorithm for Learning Feedforwad Neural Networks, Journal of Artificial Intelligence and Soft Computing Research, 10(4), 299-316, 2020, https://doi.org/10.2478/jaiscr-2020-0020.

[3] R. Chen, B. Yang, S. Li, S. Wang, Q. Cheng, An Effective Multi-population Grey Wolf Optimizer based on Reinforcement Learning for Flow Shop Scheduling Problem with Multi-machine Collaboration, Computers & Industrial Engineering, 162, 2021, https://doi.org/10.1016/j.cie.2021.107738.

[4] P. Duda, M. Jaworski, A. Cader, L. Wang, On Training Deep Neural Networks Using a Streaming Approach, Journal of Artificial Intelligence and Soft Computing Research, 10(1), 15-26, 2020, https://doi.org/10.2478/jaiscr-2020-0002.

[5] K. Cpałka, K. Łapa, L. Rutkowski, A multi-population-based algorithm with different ways of subpopulations cooperation, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, Springer, 2022 (in print).

[6] P. Dziwiński, Ł. Bartczuk, J. Paszkowski, A New Auto Adaptive Fuzzy Hybrid Particle Swarm Optimization and Genetic Algorithm, Journal of Artificial Intelligence and Soft Computing Research, 10(2), 95-111, 2020, https://doi.org/10.2478/jaiscr-2020-0007.

[7] P. Dziwiński, P. Przybył, P. Trippner, J. Paszkowski, Y. Hayashi, Hardware Implementation of a Takagi-Sugeno Neuro-Fuzzy

Table 6. Comparison of the best results obtained using the proposed algorithm with the results obtained using other methods (best cases of migration strategy and topology were selected for each algorithm).

function	proposed solution exchange of plans ↓		other population-based algorithms with the use of island models and migrations					
	YES	NO	CS	DE	FWA	GA	GWO	WOA
F01	**0.0000**	0.0004	0.2235	0.2358	0.1634	0.1952	0.2314	0.1788
F02	0.2901	**0.1491**	0.2592	0.3121	0.3347	0.2485	0.2626	0.2739
F03	**0.0665**	0.0979	0.2864	0.2498	0.2651	0.2300	0.2530	0.1967
F04	0.1885	**0.1155**	0.2077	0.2646	0.3233	0.2733	0.2432	0.2675
F05	0.0015	**0.0013**	0.2337	0.1956	0.2355	0.1936	0.1788	0.2088
F06	0.5234	0.2280	0.3280	0.2523	0.3759	0.2812	0.2840	**0.1618**
F07	0.5236	0.2079	0.2853	0.2541	0.4059	0.3551	0.3151	**0.2053**
F08	0.6023	0.5745	0.4163	0.4749	0.7975	0.4755	0.5069	**0.3920**
F09	0.3653	**0.1128**	0.2692	0.2108	0.3119	0.2598	0.2613	0.2076
F10	**0.0480**	0.1033	0.2274	0.2795	0.3353	0.2253	0.2444	0.2090
F11	**0.0000**	0.2517	0.3613	0.3466	0.4278	0.2695	0.3319	0.2230
F12	0.4730	0.4036	0.4089	0.3776	0.5840	0.4112	0.4215	**0.3010**
F13	**0.0403**	0.1920	0.3030	0.2379	0.3848	0.2844	0.3302	0.3422
F14	0.3325	0.7845	0.6621	0.6185	0.8644	0.7243	0.6172	**0.2694**
F15	0.3638	**0.1974**	0.2867	0.4249	0.4132	0.2850	0.2982	0.2764
F16	**0.1792**	0.6577	0.4893	0.4830	0.7106	0.4751	0.5515	0.2985
F17	**0.1533**	0.4042	0.3332	0.2809	0.6192	0.3841	0.4314	0.3827
F18	**0.1022**	0.4226	0.3768	0.3525	0.6325	0.3384	0.4371	0.2956
F19	**0.0846**	0.2403	0.2666	0.3129	0.4258	0.3426	0.3154	0.2452
F20	**0.2988**	0.9428	0.8730	0.7887	0.8277	0.8842	0.7282	0.4645
F21	**0.0000**	**0.0000**	0.2258	0.1921	0.2136	0.2350	0.1708	0.2104
F22	**0.2077**	0.6043	0.4513	0.5050	0.7768	0.4845	0.5166	0.3074
F23	**0.0540**	0.7071	0.5577	0.5073	0.8078	0.5666	0.5657	0.3397
F24	**0.2357**	0.5082	0.4783	0.4495	0.7294	0.4938	0.4891	0.3567
F25	**0.2056**	0.2435	0.2951	0.3974	0.4608	0.3203	0.3267	0.3005
F26	**0.0220**	0.0930	0.2745	0.2689	0.2851	0.2468	0.2341	0.1888
F27	0.1251	**0.0528**	0.2433	0.2569	0.2630	0.2193	0.2027	0.2180
F28	**0.0240**	0.2043	0.2900	0.2718	0.4005	0.3175	0.3142	0.2844
AVG	**0.1968**	0.3036	0.3541	0.3501	0.4777	0.3579	0.3594	0.2716

System Optimized by a Population Algorithm, Journal of Artificial Intelligence and Soft Computing Research, 11(3), 243-266, 2021, https://doi.org/10.2478/jaiscr-2021-0015.

[8] L. Fu, H. Ouyang, C. Zhang, S. Li, A.W. Mohamed, A constrained cooperative adaptive multi-population differential evolutionary algorithm for economic load dispatch problems, Applied Soft Computing, 121, 2022, https://doi.org/10.1016/j.asoc.2022.108719.

[9] F. Kılıç, Y. Kaya, S. Yildirim, A novel multi population based particle swarm optimization for feature selection, Knowledge-Based Systems, 219, 2021, https://doi.org/10.1016/j.knosys.2021.106894.

[10] M. Korytkowski, R. Senkerik, M.M. Scherer, R.A. Angryk, M. Kordos, A. Siwocha, Efficient Image Retrieval by Fuzzy Rules from Boosting and Metaheuristic, Journal of Artificial Intelligence and Soft Computing Research, 10(1), 57-69, 2020, https://doi.org/10.2478/jaiscr-2020-0005.

[11] K. Łapa, K. Cpałka, Flexible fuzzy PID controller (FFPIDC) and a nature-inspired method for its construction, IEEE Trans. on Industrial Informatics, 14(3), 1078-1088, 2018.

[12] K. Łapa, K. Cpałka, Ł. Laskowski, A. Cader, Z. Zeng, Evolutionary Algorithm with a Configurable Search Mechanism, Journal of Artificial Intelligence and Soft Computing Research, 10(3), 151-171, 2020.

[13] K. Łapa, K. Cpałka, M. Zalasiński, Algorithm Based on Population with a Flexible Search Mechanism, IEEE Access, 7, 132253-132270, 2019.

[14] G. Li, W. Wang, W. Zhang, Z. Wang, H. Tu, W. You, Grid search based multi-population

particle swarm optimization algorithm for multimodal multi-objective optimization, Swarm and Evolutionary Computation, 62, 2021, https://doi.org/10.1016/j.swevo.2021.100843.

[15] F. Ming, W. Gong, L. Wang, C. Lu, A tri-population based co-evolutionary framework for constrained multi-objective optimization problems, Swarm and Evolutionary Computation, 70, 2022, https://doi.org/10.1016/j.swevo.2022.101055.

[16] T. Niksa-Rynkiewicz, N. Szewczuk-Krypa, A. Witkowska, K. Cpałka, M. Zalasiński, A. Cader, Monitoring Regenerative Heat Exchanger in Steam Power Plant by Making Use of the Recurrent Neural Network, Journal of Artificial Intelligence and Soft Computing Research, 11(2), 143-155, 2021, https://doi.org/10.2478/jaiscr-2021-0009.

[17] L.R. Rodrigues, A hybrid multi-population metaheuristic applied to load-sharing optimization of gas compressor stations, Computers & Electrical Engineering, 97, 2022, https://doi.org/10.1016/j.compeleceng.2021.107632.

[18] A.K. Saha, Multi-population-based adaptive sine cosine algorithm with modified mutualism strategy for global optimization, Knowledge-Based Systems, 2022.

[19] A. Słowik, K. Cpałka, Guest Editorial: Hybrid Approaches to Nature-Inspired Population-Based Intelligent Optimization for Industrial Applications, IEEE Transactions on Industrial Informatics, 18(1), 542-545, 2022, DOI (identifier) 10.1109/TII.2021.3091137.

[20] A. Słowik, K. Cpałka, Hybrid Approaches to Nature-inspired Population-based Intelligent Optimization for Industrial Applications, IEEE Transactions on Industrial Informatics, 18(1), 546-558, 2022, DOI (identifier) 10.1109/TII.2021.3067719.

[21] A. Słowik, K. Cpałka, K. Łapa, Multi-Population Nature-Inspired Algorithm (MNIA) for the Designing of Interpretable Fuzzy Systems, IEEE Transactions on Fuzzy Systems, 28(6), 1125-1139, 2020, DOI (identifier) 10.1109/TFUZZ.2019.2959997.

[22] Y. Song, D. Wu, W. Deng, X.Z. Gao, T. Li, B. Zhang, Y. Li, MPPCEDE: Multi-population parallel co-evolutionary differential evolution for parameter optimization, Energy Conversion and Management, 228, 2021, https://doi.org/10.1016/j.enconman.2020.113661.

[23] P.N. Suganthan, N. Hansen, J.J. Liang, K. Deb, Y.P. Chen, A. Auger, S. Tiwari, Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization, KanGAL report, 2005.

[24] Y. Sun, Y. Chen, Multi-population improved whale optimization algorithm for high dimensional optimization, Applied Soft Computing, 112, 2021, https://doi.org/10.1016/j.asoc.2021.107854.

[25] J. Szczypta, A. Przybył, K. Cpałka, Some aspects of evolutionary designing optimal controllers, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, 7895, Springer, 91-100, 2013.

[26] R. Tanabe, A. Fukunaga, Evaluating the performance of SHADE on CEC 2013 benchmark problems. In 2013 IEEE Congress on evolutionary computation, pp. 1952-1959, IEEE, 2013.

[27] V. Thanasis, B.S. Efthimia, K. Dimitris, Estimation of linear trend onset in time series, Simulation Modelling Practice and Theory, 19(5), 1384-1398, 2011, https://doi.org/10.1016/j.simpat.2011.02.006.

[28] B. Yang, S. Wang, Q. Cheng, T. Jin, Scheduling of field service resources in cloud manufacturing based on multi-population competitive-cooperative GWO, Computers & Industrial Engineering, 154, 2021, https://doi.org/10.1016/j.cie.2021.107104.

[29] M. Zalasiński, K. Cpałka, A new method of on-line signature verification using a flexible fuzzy one-class classifier, Academic Publishing House EXIT, 38-53, 2011.

[30] M. Zalasiński, K. Cpałka, Novel algorithm for the on-line signature verification using selected discretization points groups, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, 7894, Springer, 493-502, 2013.

[31] M. Zalasiński, K. Cpałka, Y. Hayashi, New method for dynamic signature verification based on global features, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, 8467, Springer, 251-265, 2014.

[32] M. Zalasiński, K. Cpałka, Y. Hayashi, New fast algorithm for the dynamic signature verification using global features values, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, 9120, Springer, 175-188, 2015.

[33] M. Zalasiński, K. Cpałka, Ł. Laskowski, D.C. Wunsch, K. Przybyszewski, An Algorithm for the Evolutionary-Fuzzy Generation of on-Line Signature Hybrid Descriptors, Journal of Artificial Intelligence and Soft Computing Research, 10(3), 173-187, 2020, https://doi.org/10.2478/jaiscr-2020-0012.

[34] M. Zalasiński, K. Łapa, K. Cpałka, New algorithm for evolutionary selection of the dynamic signature global features, Artificial Intelligence and Soft Computing, Lecture Notes in Computer Science, 7895, Springer, 113-121, 2013.

[35] M. Zalasiński, K. Łapa, K. Cpałka, K. Przybyszewski, G.G. Yen, On-Line Signature Partitioning Using a Population Based Algorithm, Journal of Artificial Intelligence and Soft Computing Research, 10(1), 5-13, 2020, https://doi.org/10.2478/jaiscr-2020-0001.

[36] X. Zhang, S. Wen, D. Wang, Multi-population biogeography-based optimization algorithm and its application to image segmentation, Applied Soft Computing, 124, 2022, https://doi.org/10.1016/j.asoc.2022.109005.

[37] F. Zhao, G. Zhou, L. Wang, T. Xu, N. Zhu, Jonrinaldi, A two-stage cooperative scatter search algorithm with multi-population hierarchical learning mechanism, Expert Systems with Applications, 203, 2022, https://doi.org/10.1016/j.eswa.2022.117444.

Krystian Łapa received the M.Sc. and Ph.D. degrees from the Częstochowa University of Technology, Częstochowa, Poland, in 2010 and 2015, respectively. He received D.Sc. degree in Computer Science from Częstochowa University of Technology, Częstochowa, Poland, in 2019. He is currently an Associate Professor with the Department of Computer Engineering. Dr. Łapa has authored over 50 publications. His current research interests include computational intelligence, fuzzy systems, nature-inspired methods, and expert systems.
https://orcid.org/0000-0002-3926-5685

Krzysztof Cpałka received the M.Sc. and Ph.D. degrees from the Częstochowa University of Technology, Częstochowa, Poland, in 1997 and 2002, respectively, and the D.Sc. degree in Computer Science from the Systems Research Institute of the Polish Academy of Sciences in Warsaw, Warsaw, Poland, in 2010. Since 2010, he has been a professor with the Department of Computer Engineering, Częstochowa University of Technology. He is the author or co-author of two books and more than 100 papers, including several papers in various series of IEEE Transactions. He was a recipient of the IEEE Transactions on Neural Networks Outstanding Paper Award in 2005. His research interests include soft computing, computational intelligence, machine learning, and bio-inspired global optimization algorithms and their applications.
https://orcid.org/0000-0001-9761-118X

Marek Kisiel - Dorohinicki obtained Ph.D. in 2001 and D.Sc. in 2013 at the Department of Computer Science of the AGH University of Science and Technology in Krakow, Poland.
His main research interests are metaheuristics, agent-based systems and criminal analysis. He works as a Full Professor in the Institute of Computer Science at the AGH University of Science and Technology.
https://orcid.org/0000-0002-8459-1877

Józef Paszkowski graduated from the Military University of Technology in Warsaw, Faculty of Cybernetics (M.Sc. 1976) with specialization in organization and design of IT systems. He completed post-graduate studies in management systems at the University of Lodz and programming technology at the University of Warsaw/IMM PAN in Warsaw. In 1995 he received the Ph.D. in the fi eld of medical informatics. In the years 2000-2004 he worked as specialist in a IT company. Currently, he is an Assistant Professor at the University of Social Science in Łódź. He specializes in analyzing and modeling data processing system for distributed business and public utilities. His research and teaching interest include database systems, process management methods, information systems design technologies, computational intelligence, and knowledge engineering methods.
https://orcid.org/0000-0002-4604-651X

Maciej Dębski, Master in Management and Master in International Relations (SGH Warsaw School of Economics) Ph.D. in management (SGH Warsaw School of Economics). His research interests and practical aspects are focused on the use of marketing tools, especially in tourism and applications of artificial intelligence methods in marketing. Professionally working at University of Social Sciences as a Head of Tourism and Marketing Department.
https://orcid.org/0000-0002-8339-5073

Van-Hung Le received M.Sc. degree at Faculty Information Technology Hanoi National University of Education (2013). He received PhD degree at International Research Institute MICA HUSTCNRS/UMI - 2954 - INP Grenoble (2018). Currently, he is a lecture of Tan Trao University. His research interests include Computer vision, RANSAC and RANSAC variation and 3-D object detection, recognition; machine leaning, deep learning.
https://orcid.org/0000-0003-4302-0581