Available online at www.HighTechJournal.org

HighTech and Innovation Journal

Vol. 4, No. 1, March, 2023

ISSN: 2723-9535

Development of an Algorithm for Multicriteria Optimization of Deep Learning Neural Networks

Islam A. Alexandrov [1], Andrey V. Kirichek [2], Vladimir Z. Kuklin [1*], Leonid M. Chervyakov [1]

[1] IDTI RAS Institute for Design - Technological Informatics of RAS, Moscow, Russian Federation.

[2] Brjanskij Gosudarstvennyj Tehniceskij Universitet, Russian Federation.

Received 15 December 2022; Revised 17 February 2023; Accepted 24 February 2023; Published 01 March 2023

Abstract

Nowadays, machine learning methods are actively used to process big data. A promising direction is neural networks, in which structure optimization occurs on the principles of self-configuration. Genetic algorithms are applied to solve this nontrivial problem. Most multicriteria evolutionary algorithms use a procedure known as non-dominant sorting to rank decisions. However, the efficiency of procedures for adding points and updating rank values in non-dominated sorting (incremental non-dominated sorting) remains low. In this regard, this research improves the performance of these algorithms, including the condition of an asynchronous calculation of the fitness of individuals. The relevance of the research is determined by the fact that although many scholars and specialists have studied the self-tuning of neural networks, they have not yet proposed a comprehensive solution to this problem. In particular, algorithms for efficient non-dominated sorting under conditions of incremental and asynchronous updates when using evolutionary methods of multicriteria optimization have not been fully developed to date. To achieve this goal, a hybrid co-evolutionary algorithm was developed that significantly outperforms all algorithms included in it, including error-back propagation and genetic algorithms that operate separately. The novelty of the obtained results lies in the fact that the developed algorithms have minimal asymptotic complexity. The practical value of the developed algorithms is associated with the fact that they make it possible to solve applied problems of increased complexity in a practically acceptable time.

Keywords: Neural Networks; Genetic Algorithms; Hybrid Co-Evolutionary Algorithm; Feature Selection; Multicriteria Optimization.

1. Introduction

Today, data, along with capital and labor, have become essential resources for ensuring the prosperity of society [1–3]. However, selecting important and helpful data is quite a difficult task, for which machine learning methods are now actively applied [4]. Machine learning is a set of algorithms in which knowledge extraction is improved each time, i.e., by increasing the learning level [5].

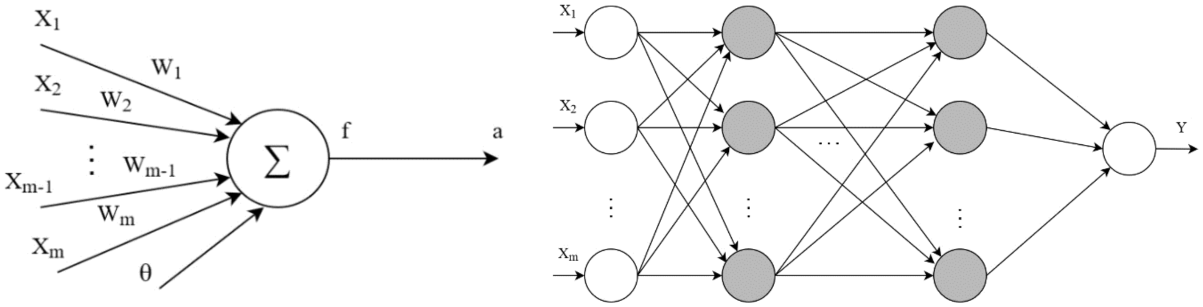
Machine learning techniques can be roughly classified as follows: learning with a teacher (supervised learning), learning without a teacher (unsupervised learning), partial learning (semi-supervised learning), learning with support (reinforcement learning), dynamic learning (online learning), and active learning [6, 7]. Supervised learning is the most popular machine learning method, where the type of "the object and its corresponding label" organizes the learning system structurally [8]. The problem is to train an algorithm that forms a nontrivial relationship between object labels

* Corresponding author: kuklin_vladimir_ran@mail.ru

http://dx.doi.org/10.28991/HIJ-2023-04-01-011

and their properties, or features. Here, initial data consist of two separate independent sets: a test set and a learning set. Unsupervised learning, on the other hand, uses information with unset object labels. Thus, any object is some set of features or a set of distance metrics in the multidimensional feature space relative to other objects considered in the sample [9–11]. In semi-supervised learning, only partially labeled data applies [12]. In online learning, learning objects feed sequentially one after another, so the algorithm needs to process each object sequentially, incrementally learning from the newly arrived knowledge about the new learning object [13]. In active learning, the learning objects feed the algorithm in a strictly selected sequence, which provides the algorithm with more effective learning [14, 15]. This approach to learning relates closely to methods of experiment planning. In this paper, the emphasis is on supervised learning algorithms and classification tasks. Classification algorithms operate with discrete data sets. They require a finite set of labels called classes. The task of the learning algorithm is to correctly and unambiguously assign an object to one of the given classes [15, 16].

Classification algorithms operate on a specific, discrete dataset. The presence of a final set of objective labels called classes is necessary. The task of the learning algorithm is to relate the object clearly to one of the presented classes [16]. Classification techniques inextricably link important information for the industry. A description type of the properties of the classification objects usually represents the data. Binkhonain & Zhao (2019) [17] presented a detailed description of the classification algorithms. One of the most promising classification algorithms is neural networks, based on the principle of the connection of many simple elements built into an optimal structure [18]. So-called artificial neurons act as simple elements. It is a noticeably simplified arithmetic model of the actual neuron structure. At the initial stage, the neuron receives output signals at its input, which then summarize and exceed the activation function of each of them, resulting in the output signal (Figure 1a). There is a multilayer frontier neural network of the most direct propagation [19] (Figure 1b), which is an oriented graph with one-way directed edges.



(a) (b)

Figure 1. a) The neuron model: b) The neural network structure in several layers with direct propagation, where gray color indicates neurons of hidden layers

A significant disadvantage of neural networks that one may consider is the solving tasks aimed at black box-type pattern recognition. In other words, they are almost understandable for logical thinking and somehow classify a specific object [20]. Problems may arise with the choice of a suitable neural network structure. A proven method to activate recognition often predetermines the structure. To configure the network parameters, a system of automatic correction of some parameters based on optimization algorithms is applied [21]. Often, the task of neural network self-configuration is a multicriteria optimization problem [22], for which solving evolutionary algorithms, a class of stochastic algorithms that simulate the process of natural evolution, is applied [23]. Genetic algorithms (GAs) are among the most demanded evolutionary search techniques [24]. Due to their intrinsic parallelism, these algorithms make it possible to find a set of Pareto-optimal solutions in a single algorithm run [25].

Many scientific papers are devoted to the self-configuration of neural network parameters by applying evolutionary search algorithms. For example, Lessmann et al. [23] configured the system parameters of the Support Vector Machine (SVM) using a genetic algorithm. The genetic algorithm provided more stable system results than the lattice search. Gavrilescu et al. [24] used evolutionary algorithms to configure the properties of neural networks in their study. Akhmedova & Semenkin [25] developed a unique collective algorithm combining various bionic techniques. Tan et al. [26] created a co-evolutionary algorithm for extended criterion optimization. Evolutionary algorithms under development are widely applied to configure the parameters of machine learning techniques [27], fuzzy logic [28], and genetic programming [29]. These algorithms are relevant for solving various practical problems, such as the problem of human-machine interaction [30]. Therefore, special attention is focused on developing software systems that promote quality interaction between human intelligence and personal computers [31–33]. Additional tasks of such interaction have become automated user data monitoring, detection of handwritten text features, oral speech, and generation of appropriate system responses in a comfortable form for perception [34–36].

Although many scientists and specialists have investigated the self-configuration of neural networks, they have not yet proposed a comprehensive solution to this problem. Moreover, solving the practically essential problem of human-machine interaction based on neural networks requires developing methods of self-configuration for machine learning algorithms in general and neural networks in particular [37–41].

Advanced multicriteria evolutionary algorithms can be divided into three classes: algorithms directly based on the Pareto dominance relation (examples: NSGA-II [42], SPEA2 [43], PESA-II [44]); algorithms that optimize the so-called indicators—functions that characterize the quality of the population as a whole with a single number (examples: IBEA [45], HypE [46]); algorithms that reduce a multicriteria problem to several scalar problems (example: MOEA/D [47]; hybrid algorithms such as NSGA-III [48] are also known.

A significant part of the algorithms belonging to the first of these classes (and hybrid algorithms) use a procedure known as non-dominant sorting to rank solutions. The disadvantage is that this non-dominated sorting procedure has a relatively high computational complexity: most algorithms known for it have $\Theta(n^2k)$ complexity, where n is the number of solutions and k is the number of objective functions.

Since this study does not address questions about objective functions and, consequently, about the compliance of the vectors of objective variable values (the "genotype" of an individual) and the vectors of objective function values (fitness of an individual), we will further identify the solution (individual) and its fitness (and consider only fitness), which allows us to consider solutions as points in the k-dimensional space of values of objective functions.

The non-dominated sorting procedure was proposed for use in multicriteria evolutionary algorithms as part of the NSGA algorithm [41, 49]. The same publication proposed an algorithm with $\Theta(n^3k)$ complexity in time and $O(\mathrm{n})$ in memory, where n is the number of points. In the next version of this algorithm, NSGA-II [42], the algorithm for non-dominated sorting was improved, its complexity was $\Theta(n^2k)$ in time and $O(\mathrm{n}^2)$ in memory, which partly contributed to the popularity of the NSGA-II algorithm. However, the complexity remains high. Thus, in almost all algorithms using non-dominated sorting, it remains either the bottleneck of the algorithm in terms of asymptotics or one of these points.

In this regard, several subsequent publications have proposed alternative, more efficient algorithms for non-dominated sorting. Most of them have $\Theta(n^2k)$ complexity in time in the worst case and $O(nk)$ in memory [50]. However, the algorithm proposed by Jensen [51] and subsequently improved in Fortin et al. [52] has a time complexity of $O(n(\log \mathrm{n})^{k\text{-}1})$ and greater efficiency than other algorithms for a large number of points.

The question of the efficiency of non-dominated sorting as part of an incremental multicriteria evolutionary algorithm was raised by Nebro & Durillo [53], where it was shown that the incremental version of the NSGA-II algorithm has advantages in the quality of the solutions obtained compared to the classical one (with the same restriction on the number of individuals' fitness calculations), but it significantly loses in time. A team of scholars headed by Deb proposed the first specialized non-dominated sorting algorithm designed to efficiently recalculate ranks when inserting new points, known as Efficient Non-dominated Level Update (ENLU) [54], and although it still requires $\Theta(n^2k)$ time in the worst case per insert operation, in practice the insert is often made faster. However, despite the fact that this result was improved in subsequent research [55], the efficiency of the procedure for adding points and updating rank values in non-dominated sorting—incremental non-dominated sorting—remains low. In addition, these studies did not consider the issue of operating in the context of asynchronous fitness computation when multiple threads insert points in parallel and independently of each other.

The presented research work aimed to improve the quality of information selection while solving neural network problems in human-machine interaction based on creating a self-constructed co-evolutionary optimization algorithm. The co-evolutionary genetic algorithm for multicriteria optimization is developed based on the standard genetic algorithms VEGA (Vector Evaluated Genetic Algorithm), SPEA (Strength Pareto Evolutionary Algorithm), and NSGA-2 (Non-Dominated Sorting Genetic Algorithm). The developed hybrid co-evolutionary multicriteria optimization algorithm combines the advantages and eliminates the disadvantages of its constituent algorithms, thus significantly increasing the efficiency of its work.

2. Materials and Methods

The authors employed efficient analytical techniques, statistics, probability theory, evolutionary systems, hardware learning, and monitoring of hidden information patterns. Figure 2 shows the algorithm of the research methodology.

To perform research to select informative features, the authors developed software in the C# language in Microsoft Visual Studio. It integrates evolutionary algorithms for multicriteria optimization, including VEGA, NSGA, SPEA, and the hybrid co-evolutionary algorithm.

VEGA is an algorithmic tool invented by Schaffer at the end of the 1980s. It actively uses sampling according to specific criteria; therefore, the percentage of individuals sampled according to given parameters is identical. An algorithm called "NSGA" created by Srinivas and Deb [41], uses a Pareto dominance approach to calculate fitness, dividing fitness to maintain the population distribution ranges in a stable state. In the SPEA methodology used by Zitzler et al. [43], not quite combinable solutions found on the actual iteration remain mostly external. The fitness of individuals is calculated from the position of Pareto dominance. Clustering reduces the number of external solutions stored in the set. The listed optimization methods have their pros and cons. VEGA has greater convergence but lacks a mechanism for uniform distribution of the upper layer of the Pareto front surface. NSGA provides excellent coverage of the Pareto front, like some other techniques, at an increased computational cost. This scientific paper presents a co-evolutionary genetic tool for multicriteria optimization (see Section 3.1), where the investigated algorithms combine into a group.

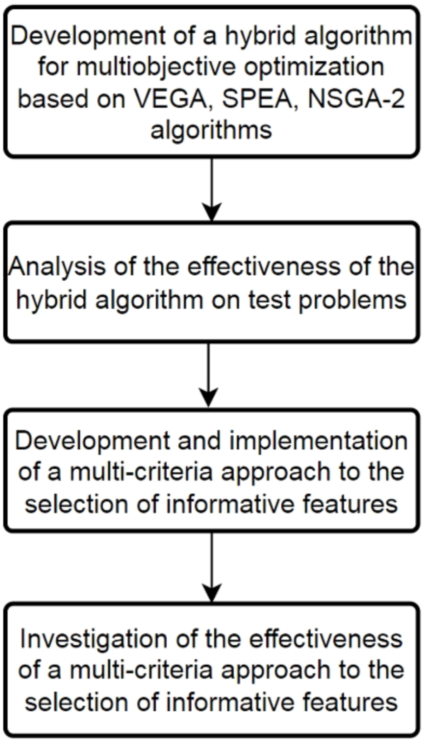

Development of a hybrid algorithm for multiobjective optimization based on VEGA, SPEA, NSGA-2 algorithms
Analysis of the effectiveness of the hybrid algorithm on test problems
Development and implementation of a multi-criteria approach to the selection of informative features
Investigation of the effectiveness of a multi-criteria approach to the selection of informative features

Figure 2. The research methodology

The study of the rationality of evolutionary algorithms for multicriteria optimization relied on the recognition of handwritten numeric symbols. The primary analytical information came from the Modified National Institute of Standards and Technology (MNIST) dataset, which included sixty thousand training samples and ten thousand handwritten digit samples [43]. The MNIST dataset contains five different handwritten digit recognition tasks:

- MNIST represents numeric characters from zero to nine written by hand on a black background;
- MNIST rotated represents images inverted at a random angle from the corresponding base;
- MNIST random background represents numbers arranged over randomly generated background noise;
- MNIST background images represent MNISTs applied to parts of a limited set of images;
- MNIST bare background images represent optional angle-spaced MNIST pictures applied to pieces of the images.

3. Results and Discussion

3.1. Creating a Multicriteria Optimization Algorithm SelfCOMOGA

Based on the VEGA, SPEA, and NSGA-2 methods, the authors developed a co-evolutionary genetic algorithm for multicriteria optimization called the Self-configuring Co-evolutionary Multi-objective Genetic Algorithm, which sounds SelfCOMOGA [28], where the previously listed algorithms belong to a single group. This algorithm is a progression of the technique of selective, extended learning systems with distance learning proposed in his time by the researcher E.A. Sopov [29]. The authors chose the VEGA, SPEA, and NSGA-2 systems because of various approaches to selecting population individuals that provide an opportunity to avoid stagnation. The main feature of the hybrid algorithm assumed to be optimal for a particular activity is monitoring the performance of the evolving algorithms.

Figure 3 shows the composition of the SelfCOMOGA algorithm. At the initial stage of algorithm testing, instances of each method (VEGA, NSGAA-2, and SREA) are artificially generated. The initialization of algorithmic properties randomly selected from the existing number of predetermined variants occurs by developing a single instance of every previously mentioned algorithm.

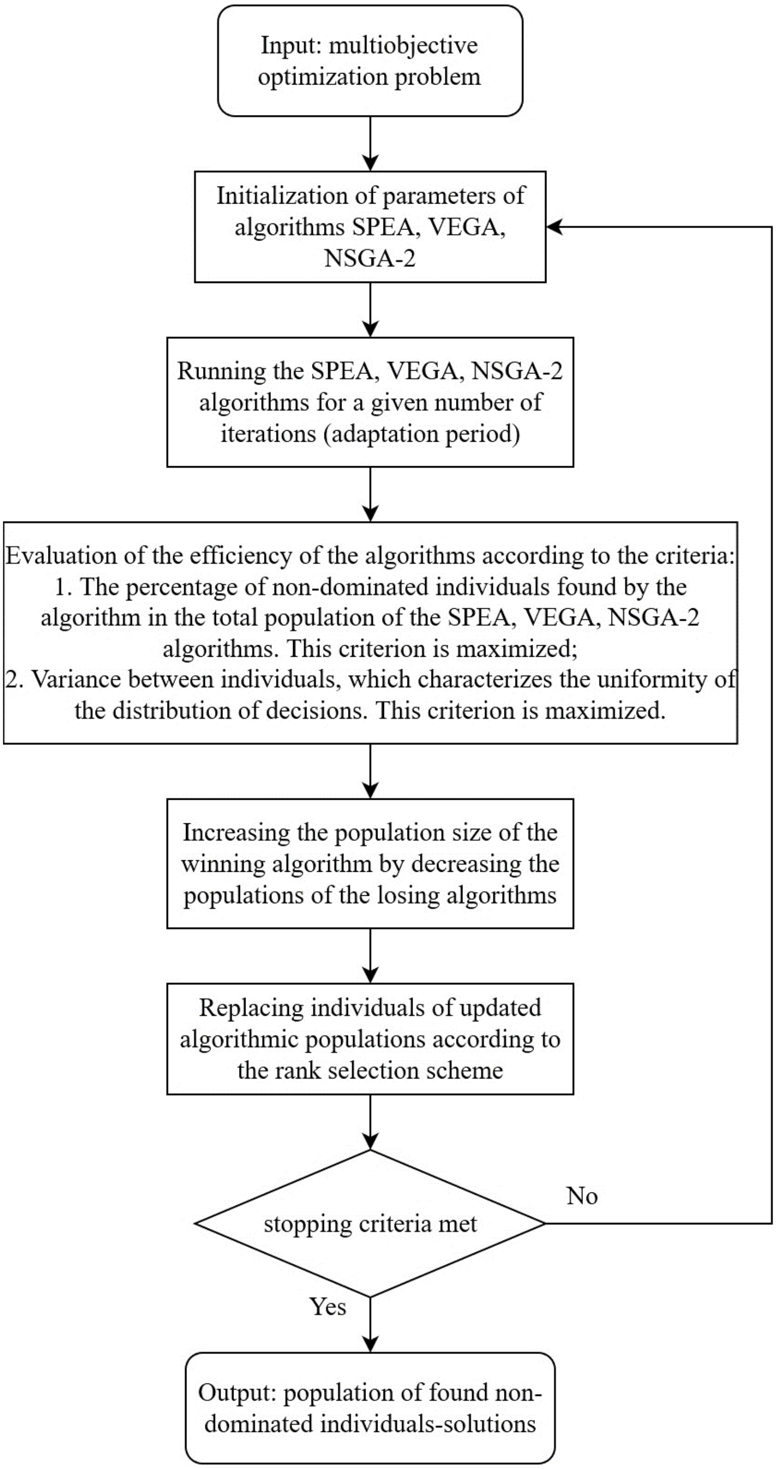

Input: multiobjective optimization problem
Initialization of parameters of algorithms SPEA, VEGA, NSGA-2
Running the SPEA, VEGA, NSGA-2 algorithms for a given number of iterations (adaptation period)
Evaluation of the efficiency of the algorithms according to the criteria:
1. The percentage of non-dominated individuals found by the algorithm in the total population of the SPEA, VEGA, NSGA-2 algorithms. This criterion is maximized;
2. Variance between individuals, which characterizes the uniformity of the distribution of decisions. This criterion is maximized.
Increasing the population size of the winning algorithm by decreasing the populations of the losing algorithms
Replacing individuals of updated algorithmic populations according to the rank selection scheme
stopping criteria met
No
Yes
Output: population of found non-dominated individuals-solutions

Figure 3. Flowchart of the hybrid co-evolutionary calculation method "SelfCOMOGA"

In the second stage, the testing of the initialized methods SPEA, VEGA, and NSGA-2 occurs with a given number of iterations. This stage is referred to as the adaptation period and is one of the co-evolutionary algorithms, SelfCOOMGA.

In the third stage, the co-evolutionary algorithm's efficiency is evaluated against the selected quality standards. Any algorithmic iteration has several ideal solutions without the possibility of comparing methods, so many of them cannot be applied qualitatively. The authors used the analyzed sum of the following criteria to monitor individual algorithms in SelfCOMOGA:

- K1 represents the percentage of not enough coordinated solutions. A general solution cycle of co-evolutionary systems undergoing non-dominated filtering is developed with further extraction of a large number of the best solutions from the sorted array for further calculating their proportion relative to the co-evolutionary methods included in this set;

- K2 is the uniformity of the distribution of not quite dominant solutions of the co-evolutionary method computed as the variance of distances in the displayed criteria space.

The fourth stage is the distribution of resources. The size of the winning algorithm group, found based on the analyzed sum of the criteria described above, increases due to a decrease in the population parameters of other algorithms. The population size of the losing method decreases by a few percent from its stable size. In the fifth stage, already completely new method populations are filled with problems from the general pool selected using rank selection. The stopping criterion of the hybrid co-evolutionary algorithm is a predetermined integration algorithmic number.

3.2. Analysis of Algorithm Efficiency

The efficiency evaluation of the developed co-evolutionary algorithm involves ten test tasks of multicriteria optimization for analyzing the practical work of the algorithm.

Test task number 1:

$$\begin{aligned} f_1 &= x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}^{\Sigma}\left[x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2 \to min, \\ f_2 &= 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_1}^{\Sigma}\left[x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right)\right]^2 \to min, \end{aligned} \tag{1}$$

where $n = 30, J_1 = \{j|j\ -\ \text{odd}, 2 \le j \le n\}, J_2 = \{j|j\ -\ \text{even}, 2 \le j \le n\}$; domain: $\bar{x} \in [0,1] \cdot [-1,1]^{n-1}$.

Pareto set: $0 \le x_1 \le 1, x_j = sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = \overline{2,n}$. Pareto front: $f_2 = 1 - \sqrt{f_1}, 0 \le f_1 \le 1$.

Test task number 2:

$$\begin{cases} f_1 = x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}^{\Sigma} y_j^2 \to min, \\ f_2 = 1 - \sqrt{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}^{\Sigma} y_j^2 \to min, \{ \end{cases} \tag{2}$$

where $n = 30, J_1 = \{j|j - \text{odd}, 2 \le j \le n\}, J_2 = \{j|j - \text{even}, 2 \le j \le n\}$.

$$y_j = \begin{cases} x_j - \left(0.3x_1^2 \cdot cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right) \cdot cos\left(6\pi x_1 + \frac{j\pi}{n}\right),.\ j \in J_1, \\ x_j - \left(0.3x_1^2 \cdot cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right) \cdot cos\left(6\pi x_1 + \frac{j\pi}{n}\right),.\ j \in J_2, \end{cases} \tag{3}$$

Domain: $\bar{x} \in [0,1] \cdot [-1,1]^{n-1}$; Pareto set: $0 \le x_1 \le 1$

$$x_j = \begin{cases} \left(0.3x_1^2 \cdot cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right) \cdot cos\left(6\pi x_1 + \frac{j\pi}{n}\right),.\ j \in J_1, \\ \left(0.3x_1^2 \cdot cos\left(24\pi x_1 + \frac{4j\pi}{n}\right) + 0.6x_1\right) \cdot cos\left(6\pi x_1 + \frac{j\pi}{n}\right),.\ j \in J_2, \end{cases} \tag{4}$$

Pareto front: $f_2 = 1 - \sqrt{f_1}, 0 \le f_1 \le 1$.

Test task number 3:

$$\begin{aligned} f_1 &= x_1 + \frac{2}{|J_1|}(4 \cdot \textstyle\sum_{j\in J_1} y_j^2 - 2 \cdot \prod_{j\in J_1} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2) \to min, \\ f_2 &= 1 - \sqrt{x_1} + \frac{2}{|J_2|}(4 \cdot \textstyle\sum_{j\in J_2} y_j^2 - 2 \cdot \prod_{j\in J_2} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2) \to min, \end{aligned} \tag{5}$$

where $n = 30, J_1 = \{j|j - \text{odd}, 2 \le j \le n\}, J_2 = \{j|j - \text{even}, 2 \le j \le n\}, j = \overline{2,n}$; $y_j = x_j - x_1^{0.5\left[1+\frac{3(j-2)}{n-2}\right]}$

Domain: $\bar{x} \in [0,1]^n$.

Pareto set: $0 \le x_1 \le 1$, $x_j = x_1^{0.5(1+\frac{3(j-2)}{n-2})}, j = \overline{2,n}.$

Pareto front: $f_2 = 1 - \sqrt{f_1}, 0 \le f_1 \le 1$

Test task number 4:

$$\begin{aligned} f_1 &= x_1 + \frac{2}{|J_1|}\sum_{j\in J_1}^{\Sigma} h(y_j) \to min, \\ f_2 &= 1 - x_1^2 + \frac{2}{|J_2|}\sum_{j\in J_2}^{\Sigma} h(y_j) \to min, \end{aligned} \tag{6}$$

where $n = 30, J_1 = \{j|j - \text{odd}, 2 \le j \le n\}, J_2 = \{j|j - \text{even}, 2 \le j \le n\}, j = \overline{2,n}$;

$y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), h(t) = \frac{|t|}{1+e^{2|t|}}$

Domain: $\bar{x} \in [0,1] \cdot [-2,2]^{n-1}$.

Pareto set: $0 \le x_1 \le 1\ x_j = sin(6\pi x_1 + \frac{j\pi}{n}), j = \overline{2,n}.$

Pareto front: $f_2 = 1 - f_1^2, 0 \le f_1 \le 1.$

Test task number 5:

$$f_1 = x_1 + (\frac{1}{2N} + \varepsilon) \cdot |\,sin(2N\pi x_1)| + \frac{2}{|J_1|}\sum_{j\in J_1}^{\Sigma} h(y_j) \to min,$$
$$f_2 = 1 - x_1 + (\frac{1}{2N} + \varepsilon) \cdot |\,sin(2N\pi x_1)| + \frac{2}{|J_2|}\sum_{j\in J_2}^{\Sigma} h(y_j) \to min, \qquad (7)$$

where; $n = 30, J_1 = \{j|j - \text{odd}, 2 \le j \le n\}, J_2 = \{j|j - \text{even}, 2 \le j \le n\}, j = \overline{2,n}$.

$y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = \overline{2,n}, h(t) = 2t^2 - cos(4\pi t) + 1, \varepsilon = 0.1, N = 10$.

Domain: $\bar{x} \in [0,1] \cdot [-1,1]^{n-1}$. Pareto front: $\left(\frac{i}{2N}, 1 - \frac{i}{2N}\right), i = \overline{0,2N}.$

Test task number 6:

$$f_1 = x_1 + max\{0{,}2\left(\frac{1}{2N} + \varepsilon\right) sin(2N\pi x_1)\} +$$
$$\frac{2}{|J_1|}(4 \cdot \sum_{j\in J_1} y_j^2 - 2 \cdot \prod_{j\in J_1} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2) \to min,$$
$$f_2 = 1 - x_1 + max\{0{,}2\left(\frac{1}{2N} + \varepsilon\right) sin(2N\pi x_1)\} + \qquad (8)$$
$$\frac{2}{|J_2|}(4 \cdot \sum_{j\in J_2} y_j^2 - 2 \cdot \prod_{j\in J_2} cos\left(\frac{20y_j\pi}{\sqrt{j}}\right) + 2) \to min,$$

where; $n = 30, J_1 = \{j|j - \text{odd}, 2 \le j \le n\}, J_2 = \{j|j - \text{even}, 2 \le j \le n\}, j = \overline{2,n}$;

$y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = \overline{2,n}, \varepsilon = 0.1, N = 10$.

Domain: $\bar{x} \in [0,1] \cdot [-1,1]^{n-1}$.

Pareto front: (0.1) – isolated point and *N* discontinuous parts: $f_2 = 1 - f_1, f_1 \in \cup_{i=1}^{N}\left(\frac{2i-1}{2N}, \frac{2i}{2N}\right)$.

Test task number 7:

$$\begin{cases} f_1 = \sqrt[5]{x_1} + \frac{2}{|J_1|}\sum_{j\in J_1}^{\Sigma} y_j^2 \to min, \\ f_2 = 1 - \sqrt[5]{x_1} + \frac{2}{|J_2|}\sum_{j\in J_2}^{\Sigma} y_j^2 \to min, \end{cases} \qquad (9)$$

where; $n = 30, J_1 = \{j|j - \text{odd}, 2 \le j \le n\}, J_2 = \{j|j - \text{even}, 2 \le j \le n\}, j = \overline{2,n}$;

$y_j = x_j - sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = \overline{2,n}, \varepsilon = 0.1, N = 10.$

Pareto set: $0 \le x_1 \le 1$, $0 \le x_1 \le 1, x_j = sin\left(6\pi x_1 + \frac{j\pi}{n}\right), j = \overline{2,n}.$

Domain: $\bar{x} \in [0,1] \cdot [-1,1]^{n-1}$. Pareto front: $f_2 = 1 - \sqrt{f_1}, 0 \le f_1 \le 1.$

Test task number 8:

$$f_1 = cos(0.5x_1\pi)\, cos(0.5x_2\pi) + \frac{2}{|J_1|}\sum_{j\in J_1}^{\Sigma}(x_j - 2x_2 \cdot sin(2\pi x_1 + \frac{j\pi}{n}))^2 \to min,$$
$$f_2 = cos(0.5x_1\pi)\, cos(0.5x_2\pi) + \frac{2}{|J_2|}\sum_{j\in J_2}^{\Sigma}(x_j - 2x_2 \cdot sin(2\pi x_1 + \frac{j\pi}{n}))^2 \to min, \qquad (10)$$
$$f_3 = sin(0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j\in J_3}^{\Sigma}(x_j - 2x_2\, sin(2\pi x_1 + \frac{j\pi}{n}))^2 \to min,$$

where; $J_1 = \{j|(j-1) - \text{multiple } 3, 3 \le j \le n\}, J_2 = \{j|(j-2) - \text{multiple } 3, 2 \le j \le n\}, j = \overline{2,n}$;

$J_3 = \{j|j - \text{multiple } 3, 3 \le j \le n\}, n = 30.$

Pareto set: $0 \le x_1, x_2 \le 1, x_j = 2x_2\, sin\left(2\pi x_1 + \frac{j\pi}{n}\right), j = \overline{3,n}.$

Domain: $\bar{x} \in [0,1]^2 \cdot [-2,2]^{n-2}$. Pareto front: $f_1^2 + f_2^2 + f_3^2 = 1, 0 \le f_1, f_2, f_3 \le 1.$

Test task number 9:

$$\begin{cases} f_1 = 0.5(max\{0, (1+\varepsilon)(1 - 4(2x_1 - 1)^2)\} + 2x_1)x_2 + \frac{2}{|J_1|}\sum_{j\in J_1}^{\Sigma}(x_j - 2x_2\, sin(2\pi x_1 + \frac{j\pi}{n}))^2 \to min, \\ f_2 = 0.5(max\{0, (1+\varepsilon)(1 - 4(2x_1 - 1)^2)\} + 2x_1)x_2 + \frac{2}{|J_2|}\sum_{j\in J_2}^{\Sigma}(x_j - 2x_2\, sin(2\pi x_1 + \frac{j\pi}{n}))^2 \to min, \\ f_3 = 1 - x_2 + \frac{2}{|J_3|}\sum_{j\in J_3}^{\Sigma}(x_j - 2x_2\, sin(2\pi x_1 + \frac{j\pi}{n}))^2 \to min, \end{cases} \qquad (11)$$

where; $J_1 = \{j|(j-1) - \text{multiple } 3,\, 3 \le j \le n\}, J_2 = \{j|(j-2) - \text{multiple } 3,\, 3 \le j \le n\}, j = \overline{2,n}$;

$J_3 = \{j|j - \text{multiple } 3,\, 3 \le j \le n\}, n = 30.$

Domain: $\bar{x} \in [0,1]^2 \cdot [-2,2]^{n-2}$.

Pareto set: $x_1 \in [0,0.25] \cup [0.75,1],\, 0 \le x_2 \le 1, x_j = 2x_2\, sin\left(2\pi x_1 + \frac{j\pi}{n}\right), j = \overline{3,n}$.

Pareto front: $0 \le f_3 \le 1,\, 0 \le f_1 \le \frac{1}{4}(1-f_3), f_2 = 1 - f_1 - f_3$ и $0 \le f_3 \le 1$,

$\frac{3}{4}(1-f_3) \le f_1 \le 1, f_2 = 1 - f_1 - f_3$.

Test task number 10:

$$\begin{cases} f_1 = cos(\,0.5x_1\pi)\,cos(\,0.5x_2\pi) + \frac{2}{|J_1|}\sum_{j\in J_1}^{\Sigma}(4y_j^2 - cos(\,8\pi y_j) + 1) \rightarrow min, \\ f_2 = cos(\,0.5x_1\pi)\,cos(\,0.5x_2\pi) + \frac{2}{|J_2|}\sum_{j\in J_2}^{\Sigma}(4y_j^2 - cos(\,8\pi y_j) + 1) \rightarrow min, \\ f_3 = sin(\,0.5x_1\pi) + \frac{2}{|J_3|}\sum_{j\in J_3}^{\Sigma}(4y_j^2 - cos(\,8\pi y_j) + 1) \rightarrow min, \end{cases} \tag{12}$$

where; $J_1 = \{j|(j-1) - \text{multiple } 3,\, 3 \le j \le n\}, J_2 = \{j|(j-2) - \text{multiple } 3,\, 3 \le j \le n\}, j = \overline{2,n}$;

$J_3 = \{j|j - \text{multiple } 3,\, 3 \le j \le n\}, n = 30, y_j = x_j - 2x_2\, sin(\,2\pi x_1 + \frac{j\pi}{n}), j = \overline{3,n}.$

Pareto set: $0 \le x_1, x_2 \le 1, x_j = 2x_2\, sin\left(2\pi x_1 + \frac{j\pi}{n}\right), j = \overline{3,n}.$

Domain: $\bar{x} \in [0,1]^2 \cdot [-2,2]^{n-2}$. Pareto front: $f_1^2 + f_2^2 + f_3^2 = 1, 0 \le f_1, f_2, f_3 \le 1$.

Comparison of the efficiency of the algorithms under consideration used the IGD (the indicator of generational distinction) metric:

$$IGD(A, P^*) = \frac{\sum d(v,D)}{|P^*|}, \tag{13}$$

where $P*$ is an actual Pareto front, A represents a Pareto front approximation by the optimization algorithm, v is a point of the actual Pareto front, and $d(v, D)$ represents a minimal distance between points by Euclidean metric. The smaller the IGD metric is, the better the optimization problem solution is.

The SPEA, VEGA, NSGA-2, and SelfCOMOGA algorithms presented in this paper can be random; therefore, twenty activations of each algorithm were performed in each of the ten tests. At the end of the analysis, the IGD metric designation and the working time in seconds for the software algorithm were recorded. Based on the selected statistics, the importance of distinctive features of the IGD metric between the studied methods was analyzed using the Wilcoxon test. Table 1 demonstrates the parameter values of the genetic algorithms studied in this research work and the values used in experiments.

Table 1. Values of parameters of evolutionary algorithms of multicriteria functional optimization: application in testing on specific tasks

Algorithm	Parameter	Value
SelfCOMOGA, SPEA, VEGA, NSGA-2	N_p is the population size; for the coevolutionary algorithm – the total size of all subpopulations	600
	Number of generations	500
	Type of crossover	Single-point
	Probability of crossover	1
	Probability of mutation	$\left(k \cdot N_p\right)^{-1}, k = 3$
	Accuracy of solution representation	0.0001
SelfCOMOGA	Size of the adaptation period of coevolutionary algorithms (in generations)	10
	Penalty size (% of the current population size of the coevolutionary algorithm)	10
	Minimum guaranteed population size (% of the initial population size of the coevolutionary algorithm)	10
SPEA	The maximum size of the external set	N_p

Figures 4 to 7 present the results of testing methods on test items, including the qualitative mean of the sample, comparison of the standard deviation, minimum and maximum values of the IGD metric, and the time interval of the tasks.

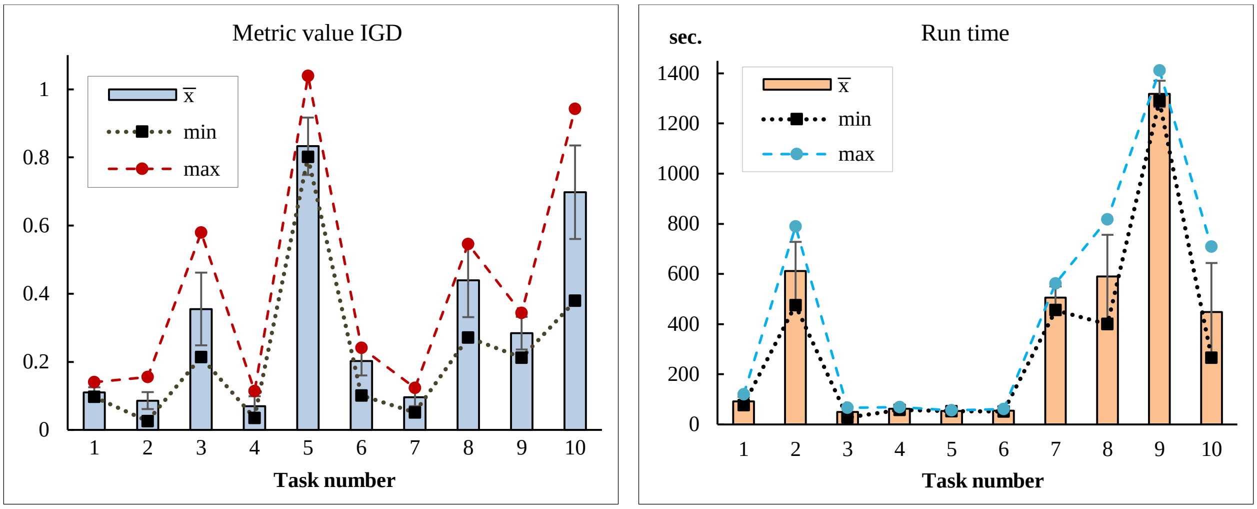


Figure 4. Results of testing the SPEA algorithm on test problems

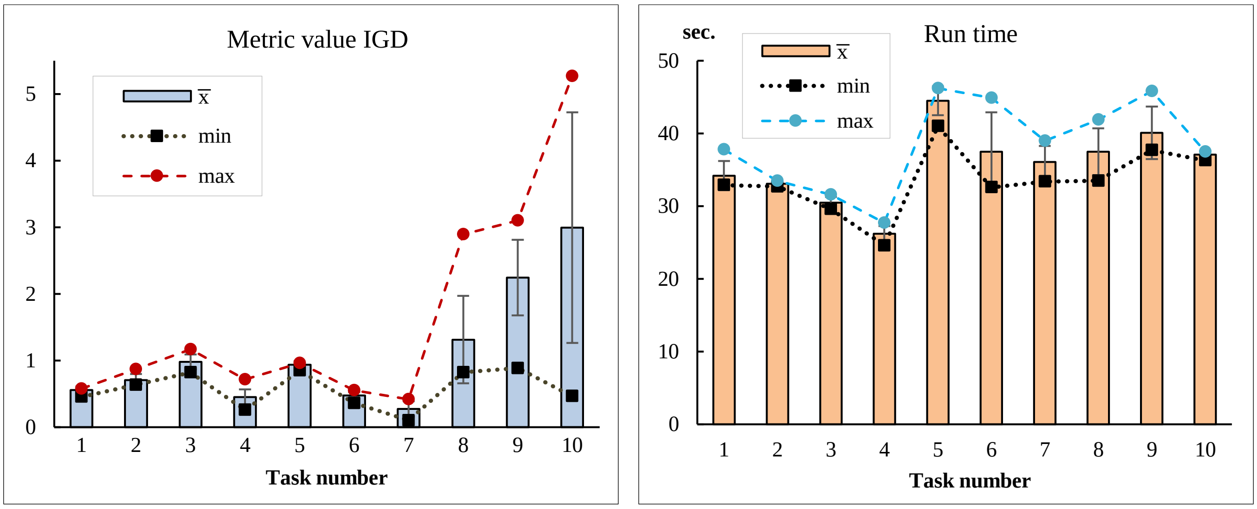


Figure 5. The results of testing the VEGA algorithm on test tasks

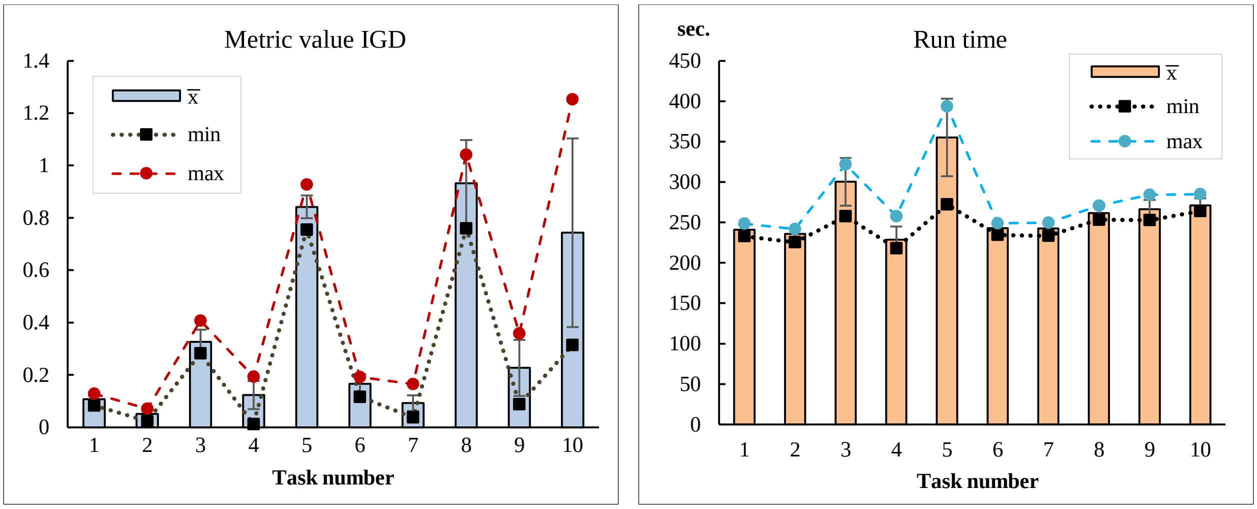


Figure 6. Results of testing the NSGA-2 algorithm on test tasks

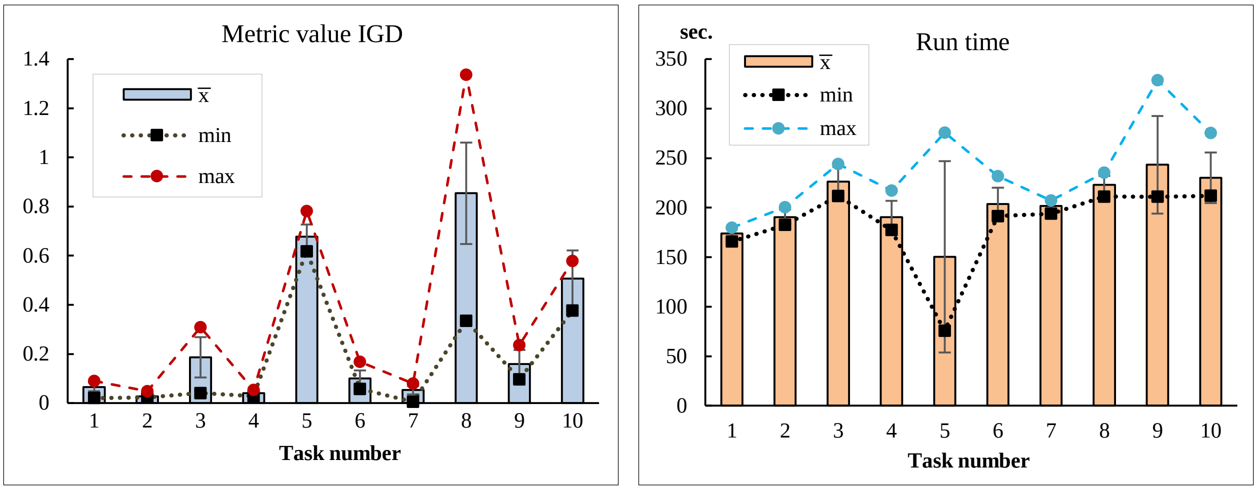


Figure 7. The results of testing the SelfCOMOGA algorithm on the test tasks

Table 2 shows the results of applying the W-criterion required to predetermine the significance of differences between the IGD-labeled metric of the SelfCOMOGA algorithm and the SPEA, VEGA, and NSGA-2 methods.

Table 2. The result of applying the Wilcoxon test for calculating the significance of differences between the IGD metric system of the SelfCOMOGA algorithm and the SPEA, VEGA, and NSGA-2 methods

Task No.	SelfCOMOGA vs. SPEA	SelfCOMOGA vs. VEGA	SelfCOMOGA vs. NSGA-2
1	Surpasses	Surpasses	Surpasses
2			
3			
4			
5			
6			
7			
8	Inferior		≈
9	Surpasses		
10			Surpasses

The results show that in nine of ten tests, the SelfCOMOGA algorithm is statistically significant ($p = 0.05$) and noticeably outperforms the available SPEA, VEGA, and NSGA-2 algorithms in the IGD parameter, giving way to the more powerful SPEA algorithm in unambiguity. The SelfCOMOGA algorithm outperforms SPEA and NSGA-2 in the implementation rate, being significantly inferior only to the VEGA algorithm and noticeably exceeding the competitor in the IGD parameter, which is the key one in this context. The obtained results prove the rationality of the developed co-evolutionary algorithm and the justification of its coupling into the co-evolution of the most commonly used evolutionary methods of multicriteria optimization with unequal decision selection technologies.

3.3. Creating and Integrating the Multicriteria Approach to Selecting Informative Features in the Process of Work

Selecting important informative features is not an easy stage in modern machine learning. The efficiency of the overall machine learning system directly depends on the quality of this stage. This research conducted and studied the method used in the multicriteria approach for automatic feature selection (Figure 8).

The proposed feature selection method based on multicriteria optimization belongs to the so-called "wrapper methods" [34, 56-57]. The presented scientific development involved a comparison with the principal component analysis (PCA). The optimization-driven feature selection tools are designed as written below. The input variables are binary vectors of length t. Here, t represents the initial number of sample features. An individual bit of the mentioned vector gains a value of one or zero, where one denotes selecting a specific feature for future integration into the model, and zero means the absence of selection. Optimization occurs according to a pair of criteria: the classification accuracy represents the maximum possible criterion and the number of features is the minimum criterion.

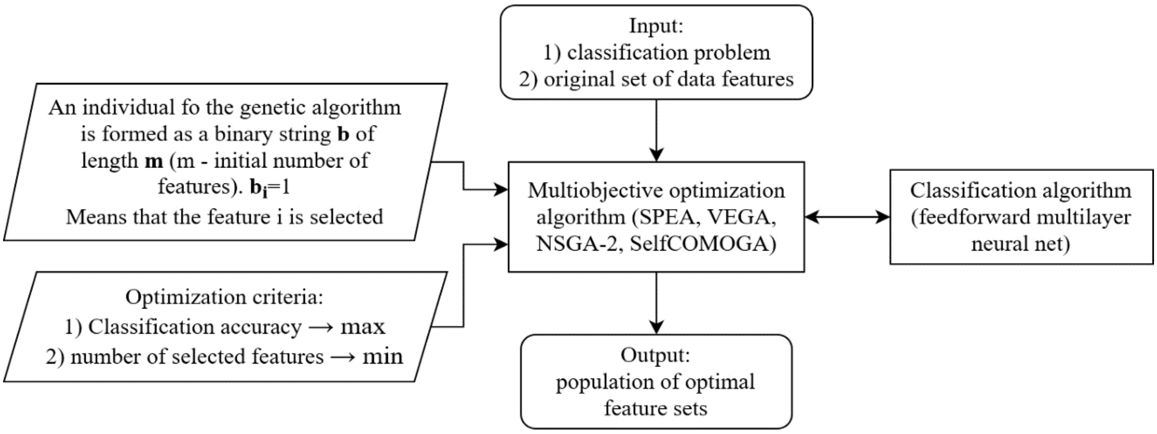

Input:
1) classification problem
2) original set of data features
An individual fo the genetic algorithm is formed as a binary string **b** of length **m** (m - initial number of features). $\mathbf{b_i}$=1
Means that the feature i is selected
Multiobjective optimization algorithm (SPEA, VEGA, NSGA-2, SelfCOMOGA)
Classification algorithm (feedforward multilayer neural net)
Optimization criteria:
1) Classification accuracy → max
2) number of selected features → min
Output:
population of optimal feature sets

Figure 8. Flowchart of the multicriteria approach selecting informative features

To streamline the process, the authors selected algorithms SPEA, VEGA, NSGA-2, and SelfCOMOGA with low probability of crossover or mutation. Their probability calculation can use the formula:

$$p = 1/(k \times |P|) \tag{14}$$

where k = 3 in completed experiments but can take other non-negative values, $|P|$ represents population size parameters, and p is a probability notation.

For conducting experiments on the target problem solving of selecting informative properties, the authors developed software in C# language in Microsoft Visual Studio 2012 with implementing the evolutionary methods of SREA, VEGA, NSGA-2, and SelfCOMOGA multicriteria optimization. The section below presents the results of the conducted digital experiment to monitor the solution of the multicriteria algorithmic problem of selecting the informative properties of the modern microsystem of the available classifiers.

3.4. Studying the Efficiency of a Multicriteria Approach to Select Special Informative Features

Studying the rationality of the hybrid learning algorithm occurred on the handwritten digit detection task, with primary data taken from MNIST [43]. The experiment involved a convolutional neural network (CNN) with a pair of convolutional layers, two subsampling layers, and a fully connected output layer. Table 3 presents the application of CNN parameter assignment in the experiments. From the start of the hybrid algorithm (HA), one can safely employ the co-evolutionary HA with the characteristics shown in Table 4.

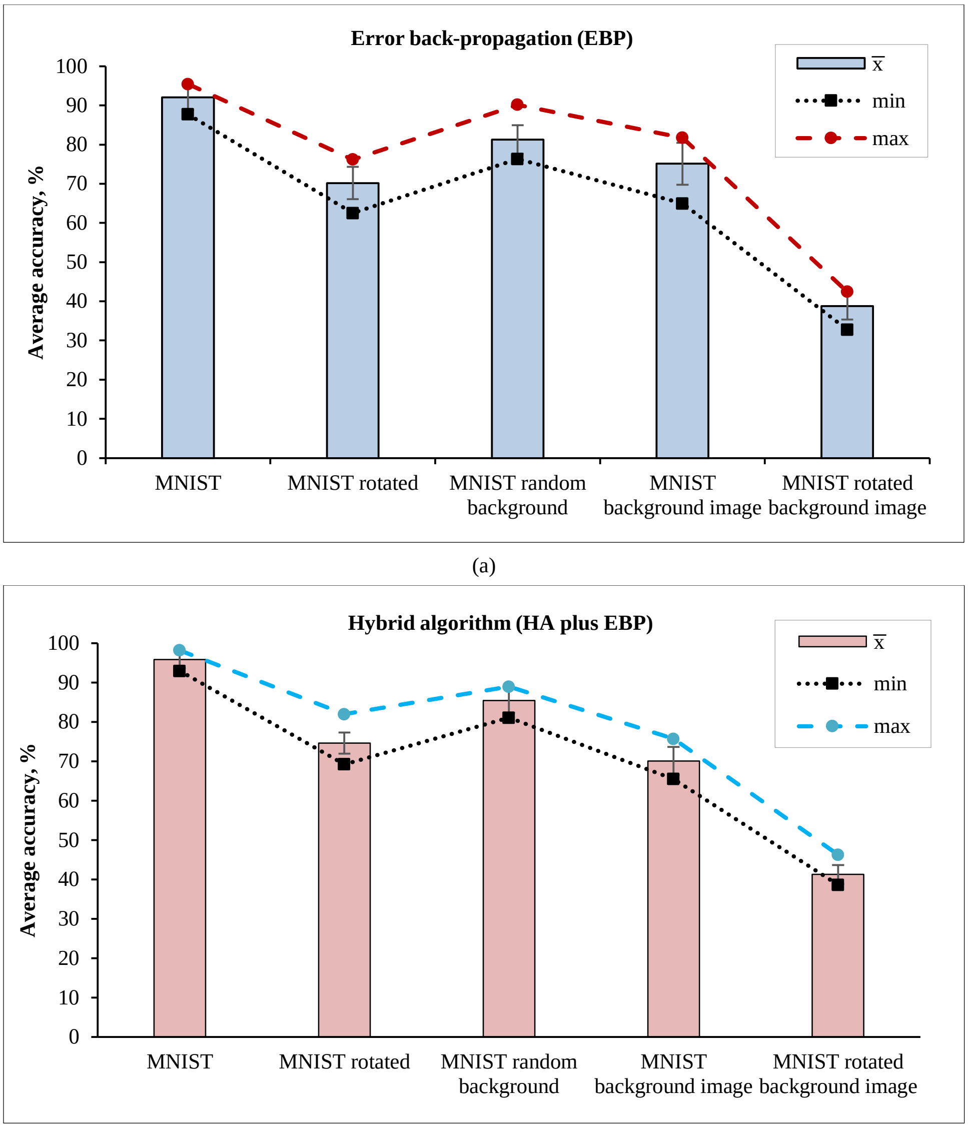
Table 3. CNN parameter notations for qualitative handwritten digit recognition tasks

Parameter	Value
Number of feature maps on the 1st convolutional layer	6
Number of feature maps on the 2nd convolutional layer	12
The dimensionality of the convolution kernel of the 1st and 2nd convolutional layers	5 × 5
The scale of the subsampling layers	2
The number of neurons of a fully connected layer	192
Number of learning epochs	50

Table 4. Parameters of co-evolutionary HA

Parameter	Value
Population size	50
Number of iterations	50
Search interval for each variable	[-3; 2]
Accuracy of search interval partitioning	0.0001
Percentage of elite solutions	4%

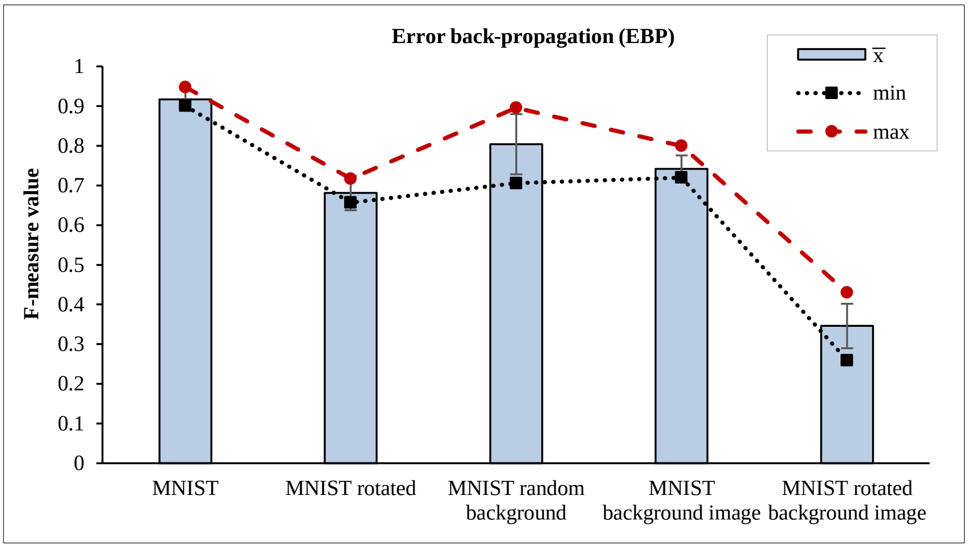
The authors compared the efficiency of CNNs studied by the error backpropagation (EBP) method and by a hybrid algorithm (HA plus EBP). However, the CNN investigated using the genetic algorithm did not show increased or noticeable efficiency due to the instantaneous spatial search dimension, so the authors omitted the results here. The analyzed criteria are classification accuracy and F-score. Totals of the available efficiency of approaches for the five previously described tasks of the MNIST group by the parameter of classification accuracy are shown in Figure 9, by the criterion of F-score – in Figure 10. The results of the algorithm operation on various test problems are shown in Figures 4 to 7.



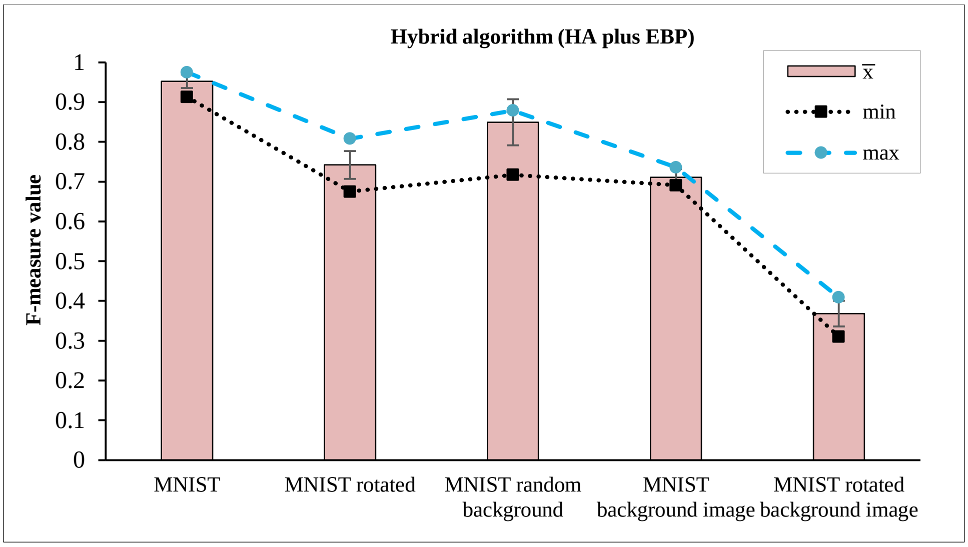
(a)

(b)

Figure 9. Study of average accuracy in percent of handwritten digits recognition by CNN learning criteria: a) Error back-propagation (EBP); b) Hybrid algorithm (HA plus EBP)



(a)



(b)

Figure 10. Comparative analysis of the average F-score of notations of handwritten numerical parameters by iterations of the learning process: a) EBP; b) HA plus EBP

According to the experimental results, applying a specialized hybrid algorithm to study CNN is much better than the error backpropagation method for four of the five tasks. The presented results demonstrate that the effectiveness of the generalized method as a whole strongly depends on the initial data used and on the chosen scheme for merging into a team at the final stage. Regarding the effectiveness of various multicriteria optimization algorithms as applied to feature selection and optimal team design, the algorithm that combines the advantages of the SPEA, NSGA-2, and VEGA algorithms turned out to be the most effective, and, as has been shown, it outperforms them noticeably. Thus, a self-configuring co-evolutionary algorithm for multicriteria optimization was developed, which outperforms the optimization algorithms included in it as components in solving test problems of multicriteria optimization and practical problems.

4. Conclusions

This research developed and studied in practice a hybrid learning algorithm for the convolutional neural network called SelfCOMOGA based on a specialized progressive optimization algorithm and the method of backpropagation of the existing error in the system. The hybrid co-evolutionary algorithm SelfCOMOGA simultaneously involves many populations undergoing independent evolution. In obtaining solutions, the listed algorithms exchange the necessary information and permissions. SelfCOMOGA is a unique algorithm that uses three well-known methods simultaneously with proven effectiveness (SPEA, VEGA, and NSGA-2). It is essential to understand that the reasons for using these algorithms are that they allow the most rationalization techniques for selecting individuals and maintaining the diversity of the population as a whole. The authors organized a comparative analysis of the primary algorithms and SelfCOMOGA on ten tests to monitor the performance value by applying two to three properties that need optimization. The experiment predetermined the superiority of the SelfCOMOGA algorithm over the individual algorithms comprising it. If you do not achieve a noticeable increase, the degree of efficiency remains at the same level.

Based on the proposed method, the authors developed software in the C# language in Microsoft Visual Studio, which was used as a basis for conducting experiments to optimize the developed techniques. The system combines VEGA, SPEA, NSGA-2, and SelfCOMOGA and makes it possible to test these algorithms on an extended set of multicriteria optimization tasks, considering the necessity of feature selection.

A hybrid training method for convolutional neural networks (CNN) based on the systematic use of genetic algorithms (HA) and error backpropagation (EBP) was created. The proposed algorithm is noticeably superior to the HA and EBP algorithms that function autonomously on the handwritten digit detection task. A comparative analysis of the available algorithmic efficiency occurred on the parameters of classification determination clarity and the F-score option.

Thus, a hybrid co-evolutionary algorithm was developed, which significantly outperforms all the algorithms included in it, including error backpropagation algorithms and genetic algorithms that operate separately. The scientific novelty of the obtained results lies in the fact that the developed non-dominant incremental sorting algorithms have minimal asymptotic complexity. The practical value of the developed algorithms is related to the fact that they make it possible to solve applied problems of increased complexity in a practically acceptable time.

Among the possible directions for further research, the following can be distinguished: the use of a multicriteria approach for designing ensembles of other classification algorithms; the application of other optimization algorithms for feature selection, classifier ensemble designing, and convolutional neural network pretraining; building ensembles of convolutional neural networks; and other deep learning algorithms.

4.1. Recommendations for the Future

Among the available ways of the present research, it is possible to highlight the application of the multicriteria approach aimed at designing ensembles of other classification algorithms, application of more optimization algorithms of feature selection, creating ensembles of classifiers and pre-training of CNN, creating combinations of convolutional neural networks and algorithms of the deep learning process.

5. Declarations

5.1. Author Contributions

Conceptualization, I.A.A. and V.Z.K.; methodology, A.V.K. and V.Z.K.; software, L.M.C.; validation, V.Z.K. and I.A.A.; formal analysis, L.M.C.; investigation, I.A.A.; resources, L.M.C.; data curation, V.Z.K.; writing—original draft preparation, I.A.A. and A.V.K.; writing—review and editing, I.A.A. and V.Z.K.; visualization, L.M.C.; supervision, V.Z.K.; project administration, V.Z.K.; funding acquisition, V.Z.K. All authors have read and agreed to the published version of the manuscript.

5.2. Data Availability Statement

All data generated or analyzed during this study are included in this published article.

5.3. Funding

Selected findings of this work were obtained under the Grant Agreement in the form of subsidies from the federal budget of the Russian Federation for state support for the establishment and development of world-class scientific centers performing R&D on scientific and technological development priorities dated April 20, 2022, No. 075-15-2022-307.

5.4. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

6. References

[1] Zhang, J. Z., Srivastava, P. R., Sharma, D., & Eachempati, P. (2021). Big data analytics and machine learning: A retrospective overview and bibliometric analysis. Expert Systems with Applications, 184, 115561. doi:10.1016/j.eswa.2021.115561.

[2] Kuklin, V., Alexandrov, I., Polezhaev, D., & Tatarkanov, A. (2023). Prospects for developing digital telecommunication complexes for storing and analyzing media data. Bulletin of Electrical Engineering and Informatics, 12(3), 1536–1549. doi:10.11591/eei.v12i3.4840.

[3] Kuklin, V. Z., Alexandrov, I. A., Umyskov, A. A., & Lampezhev, A. K. (2022). Analysis of the Prospects for Developing Storage and Processing Complexes for Multiformat Media Data. Journal of Computer Science, 18(12), 1159–1169. doi:10.3844/jcssp.2022.1159.1169

[4] Xu, Y., Wang, Z., Gao, H., Jiang, Z., Yin, Y., & Li, R. (2023). Towards machine-learning-driven effective mashup recommendations from big data in mobile networks and the Internet-of-Things. Digital Communications and Networks, 9(1), 138–145. doi:10.1016/j.dcan.2022.12.009.

[5] Antonakoudis, A., Barbosa, R., Kotidis, P., & Kontoravdi, C. (2020). The era of big data: Genome-scale modelling meets machine learning. Computational and Structural Biotechnology Journal, 18, 3287–3300. doi:10.1016/j.csbj.2020.10.011.

[6] Párizs, R. D., Török, D., Ageyeva, T., & Kovács, J. G. (2022). Machine Learning in Injection Molding: An Industry 4.0 Method of Quality Prediction. Sensors, 22(7), 2704. doi:10.3390/s22072704.

[7] Bachute, M. R., & Subhedar, J. M. (2021). Autonomous Driving Architectures: Insights of Machine Learning and Deep Learning Algorithms. Machine Learning with Applications, 6, 100164. doi:10.1016/j.mlwa.2021.100164.

[8] Yakimovich, A., Beaugnon, A., Huang, Y., & Ozkirimli, E. (2021). Labels in a haystack: Approaches beyond supervised learning in biomedical applications. Patterns, 2(12), 100383. doi:10.1016/j.patter.2021.100383.

[9] Alexandrov, I. A., Kuklin, V. Z., Muranov, A. N., & Tatarkanov, A. A. (2022). Theoretical Foundations of an Algorithm of Visualization of a Set of Points of a Multidimensional Space for Use in Anthropotechnical Decision Support Systems. Proceedings of the Institute for System Programming of the RAS, 34(4), 201–210. doi:10.15514/ispras-2022-34(4)-14.

[10] Ige, A. O., & Mohd Noor, M. H. (2022). A survey on unsupervised learning for wearable sensor-based activity recognition. Applied Soft Computing, 127, 109363. doi:10.1016/j.asoc.2022.109363.

[11] Tatarkanov, A., Alexandrov, I., Muranov, A., & Lampezhev, A. (2022). Development of a Technique for the Spectral Description of Curves of Complex Shape for Problems of Object Classification. Emerging Science Journal, 6(6), 1455–1475. doi:10.28991/esj-2022-06-06-015.

[12] Sawant, S. S., & Prabukumar, M. (2020). A review on graph-based semi-supervised learning methods for hyperspectral image classification. The Egyptian Journal of Remote Sensing and Space Science, 23(2), 243–248. doi:10.1016/j.ejrs.2018.11.001.

[13] Rožanec, J. M., Trajkova, E., Dam, P., Fortuna, B., & Mladenic, D. (2022). Streaming Machine Learning and Online Active Learning for Automated Visual Inspection. IFAC-PapersOnLine, 55(2), 277–282. doi:10.1016/j.ifacol.2022.04.206.

[14] Zhao, T., Zheng, Y., & Wu, Z. (2022). Improving computational efficiency of machine learning modeling of nonlinear processes using sensitivity analysis and active learning. Digital Chemical Engineering, 3, 100027. doi:10.1016/j.dche.2022.100027.

[15] Khayrutdinov, M. M., Golik, V. I., Aleksakhin, A. V., Trushina, E. V., Lazareva, N. V., & Aleksakhina, Y. V. (2022). Proposal of an algorithm for choice of a development system for operational and environmental safety in mining. Resources, 11(10), 88. doi:10.3390/resources11100088.

[16] Zheng, X., Li, P., & Wu, X. (2022). Data stream classification based on extreme learning machine: a review. Big Data Research, 30, 100356. doi:10.1016/j.bdr.2022.100356.

[17] Binkhonain, M., & Zhao, L. (2019). A review of machine learning algorithms for identification and classification of non-functional requirements. Expert Systems with Applications: X, 1, 100001. doi:10.1016/j.eswax.2019.100001.

[18] Singh, M. P., & Saraswat, V. K. (2017). Multilayer feed forward neural networks for non-linear continuous bidirectional associative memory. Applied Soft Computing, 61, 700–713. doi:10.1016/j.asoc.2017.08.026.

[19] Laddach, K., Łangowski, R., Rutkowski, T. A., & Puchalski, B. (2022). An automatic selection of optimal recurrent neural network architecture for processes dynamics modelling purposes. Applied Soft Computing, 116, 108375. doi:10.1016/j.asoc.2021.108375.

[20] Stewart, R. H., Palmer, T. S., & DuPont, B. (2021). A survey of multi-objective optimization methods and their applications for nuclear scientists and engineers. Progress in Nuclear Energy, 138, 103830. doi:10.1016/j.pnucene.2021.103830.

[21] Maier, H. R., Razavi, S., Kapelan, Z., Matott, L. S., Kasprzyk, J., & Tolson, B. A. (2019). Introductory overview: Optimization using evolutionary algorithms and other metaheuristics. Environmental Modelling & Software, 114, 195–213. doi:10.1016/j.envsoft.2018.11.018.

[22] Grygar, D., & Fabricius, R. (2019). An efficient adjustment of genetic algorithm for Pareto front determination. Transportation Research Procedia, 40, 1335–1342. doi:10.1016/j.trpro.2019.07.185.

[23] Lessmann, S., Stahlbock, R., & Crone, S. F. (2006). Genetic Algorithms for Support Vector Machine Model Selection. The 2006 IEEE International Joint Conference on Neural Network Proceedings. doi:10.1109/ijcnn.2006.1716515.

[24] Gavrilescu, M., Floria, S. A., Leon, F., & Curteanu, S. (2022). A Hybrid Competitive Evolutionary Neural Network Optimization Algorithm for a Regression Problem in Chemical Engineering. Mathematics, 10(19), 3581. doi:10.3390/math10193581.

[25] Akhmedova, S. A., & Semenkin, E. S. (2016). Collective bionic algorithm with biogeography based migration operator for binary optimization. Journal of Siberian Federal University - Mathematics & Physics, 9(1), 3–10. doi:10.17516/1997-1397-2016-9-1-3-10.

[26] Tan, K. C., Chew, Y. H., Lee, T. H., & Yang, Y. J. (n.d.). A cooperative coevolutionary algorithm for multiobjective optimization. SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme - System Security and Assurance (Cat. No.03CH37483). doi:10.1109/icsmc.2003.1243847.

[27] Fan, Z., Zi-xuan, X., & Ming-hu, W. (2023). State of health estimation for Li-ion battery using characteristic voltage intervals and genetic algorithm optimized back propagation neural network. Journal of Energy Storage, 57, 106277. doi:10.1016/j.est.2022.106277.

[28] Zangeneh, M., Aghajari, E., & Forouzanfar, M. (2020). A Review on Optimization of Fuzzy Controller Parameters in Robotic Applications. IETE Journal of Research, 68(6), 4150–4159. doi:10.1080/03772063.2020.1787878.

[29] Sopov, E. (2015). A Self-configuring Metaheuristic for Control of Multi-Strategy Evolutionary Search. Advances in Swarm and Computational Intelligence. ICSI 2015. Lecture Notes in Computer Science, 9142. Springer, Cham, Switzerland. doi:10.1007/978-3-319-20469-7_4.

[30] Semeraro, F., Griffiths, A., & Cangelosi, A. (2023). Human–robot collaboration and machine learning: A systematic review of recent research. Robotics and Computer-Integrated Manufacturing, 79, 102432. doi:10.1016/j.rcim.2022.102432.

[31] Hamdi, Y., Boubaker, H., Rabhi, B., Qahtani, A. M., Alharithi, F. S., Almutiry, O., Dhahri, H., & Alimi, A. M. (2022). Deep learned BLSTM for online handwriting modeling simulating the Beta-Elliptic approach. Engineering Science and Technology, an International Journal, 35, 101215. doi:10.1016/j.jestch.2022.101215.

[32] Xu, X., Li, D., Zhou, Y., & Wang, Z. (2022). Multi-type features separating fusion learning for Speech Emotion Recognition. Applied Soft Computing, 130, 109648. doi:10.1016/j.asoc.2022.109648.

[33] Peer, D., Stabinger, S., & Rodriguez-Sanchez, A. (2021). Auto-tuning of Deep Neural Networks by Conflicting Layer Removal. Neural Networks. doi:10.48550/arXiv.2103.04331.

[34] Alkatheiri, M. S. (2022). Artificial intelligence assisted improved human-computer interactions for computer systems. Computers and Electrical Engineering, 101, 107950. doi:10.1016/j.compeleceng.2022.107950.

[35] Ualiyeva, R. (2023). Functional role of vitelline glands and mehlis gland in the process of resistant egg shell formation in trematodes. OnLine Journal of Biological Sciences, 23(2), 124–132. doi:10.3844/ojbsci.2023.124.132.

[36] Fan, X., & Zhong, X. (2022). Artificial intelligence-based creative thinking skill analysis model using human–computer interaction in art design teaching. Computers and Electrical Engineering, 100, 107957. doi:10.1016/j.compeleceng.2022.107957.

[37] Abiodun, O. I., Jantan, A., Omolara, A. E., Dada, K. V., Mohamed, N. A. E., & Arshad, H. (2018). State-of-the-art in artificial neural network applications: A survey. Heliyon, 4(11), 938. doi:10.1016/j.heliyon.2018.e00938.

[38] Coulibaly, S., Kamsu-Foguem, B., Kamissoko, D., & Traore, D. (2022). Deep Convolution Neural Network sharing for the multi-label images classification. Machine Learning with Applications, 10, 100422. doi:10.1016/j.mlwa.2022.100422.

[39] Costa-Carrapiço, I., Raslan, R., & González, J. N. (2020). A systematic review of genetic algorithm-based multi-objective optimisation for building retrofitting strategies towards energy efficiency. Energy and Buildings, 210, 109690. doi:10.1016/j.enbuild.2019.109690.

[40] Schaffer, J. D. (1985). Multiple objective optimization with vector evaluated genetic algorithms. Proceedings of an international conference on genetic algorithms and their applications, 24-26 July, 1985, Carnegie-Mellon University, Pittsburgh, United States.

[41] Srinivas, N., & Deb, K. (1994). Muiltiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evolutionary Computation, 2(3), 221–248. doi:10.1162/evco.1994.2.3.221.

[42] Deb, K., Pratap, A., Agarwal, S., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation, 6(2), 182–197. doi:10.1109/4235.996017.

[43] Zitzler, E., Laumanns, M., & Thiele, L. (2001). SPEA2: Improving the strength 418 Pareto evolutionary algorithm. Tech. Rep., Eidgenössische Technische 419 Hochschule Zürich (ETH), Institut für Technische Informatik und Kom-420 munikationsnetze (TIK), 1-22. doi:10.3929/ethz-a-004284029.

[44] Corne, D. W., Jerram, N. R., Knowles, J. D., & Oates, M. J. (2001). PESA-II: Region-based selection in evolutionary multiobjective optimization. Proceedings of the 3rd annual conference on genetic and evolutionary computation, 7-11 July, 2001, San Francisco, United States.

[45] Zitzler, E., & Künzli, S. (2004). Indicator-Based Selection in Multiobjective Search. Parallel Problem Solving from Nature - PPSN VIII, PPSN 2004, Lecture Notes in Computer Science, 3242, Springer, Berlin, Germany. doi:10.1007/978-3-540-30217-9_84.

[46] Bader, J., & Zitzler, E. (2011). HypE: An algorithm for fast hypervolume-based many-objective optimization. Evolutionary Computation, 19(1), 45–76. doi:10.1162/EVCO_a_00009.

[47] Zhang, Q., & Li, H. (2007). MOEA/D: A multiobjective evolutionary algorithm based on decomposition. IEEE Transactions on Evolutionary Computation, 11(6), 712–731. doi:10.1109/TEVC.2007.892759.

[48] Deb, K., & Jain, H. (2014). An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, Part I: Solving problems with box constraints. IEEE Transactions on Evolutionary Computation, 18(4), 577–601. doi:10.1109/TEVC.2013.2281535.

[49] Zheng, W., Liu, Y., & Doerr, B. (2022). A first mathematical runtime analysis of the Non-Dominated Sorting Genetic Algorithm II (NSGA-II). In Proceedings of the AAAI Conference on Artificial Intelligence, 36(9), 10408-10416. doi:10.1609/aaai.v36i9.21283.

[50] Roy, P. C., Islam, Md. M., & Deb, K. (2016). Best Order Sort. Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion. doi:10.1145/2908961.2931684.

[51] Jensen, M. T. (2003). Reducing the Run-Time Complexity of Multiobjective EAs: The NSGA-II and Other Algorithms. IEEE Transactions on Evolutionary Computation, 7(5), 503–515. doi:10.1109/TEVC.2003.817234.

[52] Fortin, F.-A., Grenier, S., & Parizeau, M. (2013). Generalizing the improved run-time complexity algorithm for non-dominated sorting. Proceedings of the 15th Annual Conference on Genetic and Evolutionary Computation. doi:10.1145/2463372.2463454.

[53] Nebro, A. J., & Durillo, J. J. (2009). On the effect of applying a steady-state selection scheme in the multi-objective genetic algorithm NSGA-II. Studies in Computational Intelligence, 193, 435–456. doi:10.1007/978-3-642-00267-0_16.

[54] Li, K., Deb, K., Zhang, Q., & Zhang, Q. (2017). Efficient Nondomination Level Update Method for Steady-State Evolutionary Multiobjective Optimization. IEEE Transactions on Cybernetics, 47(9), 2838–2849. doi:10.1109/TCYB.2016.2621008.

[55] Mishra, S., Mondal, S., & Saha, S. (2017). Improved solution to the non-domination level update problem. Applied Soft Computing, 60, 336–362. doi:10.1016/j.asoc.2017.06.038.

[56] Canayaz, M. (2022). Classification of diabetic retinopathy with feature selection over deep features using nature-inspired wrapper methods. Applied Soft Computing, 128, 109462. doi:10.1016/j.asoc.2022.109462.

[57] Greenberg, C. S., Mason, L. P., Sadjadi, S. O., & Reynolds, D. A. (2020). Two decades of speaker recognition evaluation at the national institute of standards and technology. Computer Speech & Language, 60, 101032. doi:10.1016/j.csl.2019.101032.