SURVEY ON MULTI-OBJECTIVE-BASED PARAMETER OPTIMIZATION FOR DEEP LEARNING

Abstract

Deep-learning models form some of the most powerful machine-learning models for the extraction of important features. Most of the designs of deep neural models (i.e., the initialization of parameters) are still manually tuned; hence, obtaining a model with high performance is exceedingly time-consuming and occasionally impossible. Optimizing the parameters of deep networks therefore requires improved optimization algorithms with high convergence rates. The single objective-based optimization methods that are generally used are mostly time-consuming and do not guarantee optimum performance in all cases. Mathematical optimization problems that contain multiple objective functions that must be optimized simultaneously fall under the category of multi-objective optimization (sometimes referred to as Pareto optimization). Multi-objective optimization problems form one of the alternative yet useful options for parameter optimization; however, this domain is a bit underexplored. In this survey, we focus on exploring the effectiveness of multi-objective optimization strategies for parameter optimization in conjunction with deep neural networks. The case studies that are used in this study focus on how the two methods are combined to provide valuable insights into the generation of predictions and analysis in multiple applications.

Keywords

deep learning, multi-objective optimization, parameter optimization, neural networks

Citation

Computer Science 24(3) 2023: 321–353

Copyright

© 2023 Author(s). This is an open access publication, which can be used, distributed and reproduced in any medium according to the Creative Commons CC-BY 4.0 License.
1. Introduction

Deep learning [46] is a particular kind of machine learning that uses artificial neural networks and is motivated by the idea of information processing in biological systems. Deep learning helps in classification, detection, segmentation, etc. Deep neural networks (DNNs) are made up of several layers of interconnected nodes, and they learn to perform tasks through the adjustments of a network’s parameters.

The hyperparameters of DNNs include the number of layers, the number of neurons in each layer, the activation functions, the learning rate, the optimizer, the loss function, and the batch size, among others. These hyperparameters need to be set up manually; therefore, developing a deep-learning model for various types of problems is a complex task that requires significant effort, as numerous parameters require fine-tuning. Typically, these models are developed using the knowledge of skilled experts; however, there has been a surge in research over the past few years for designing deep-learning architectures that use optimization techniques. Initially, optimization methods such as grid searches [3], random searches [5], and Bayesian optimization [72] were used for parameter optimization. Along with these techniques, single-objective optimization algorithms were also employed to achieve high levels of classification accuracy [50, 51, 54, 64, 74, 86, 87, 97]. Since the optimization of more than one objective is needed, multi-objective optimization methods were introduced. For instance, an objective could be to obtain the maximum accuracy with the least parameters or to do so in the least amount of time [38, 44, 52, 55].

The method of optimization involves determining the optimal or best solution, which can be done by searching for maximum or minimum values using single or multiple objectives. When a problem has more than one objective, it is known as multi-objective optimization (MOO). MOO has many applications in the real world, including mechanics, politics, finance, and economics. In the field of mechanics [18, 43], for example, MOO can be used to minimize the total cost of tube heat exchangers and shells (including annual energy expenditure and capital investment) while reducing the heat exchanger’s length using a genetic algorithm (GA) [59]. MOO algorithms are designed to find optimal values for variables such as baffle spacing, outer diameters, and outer tube diameters. In politics [34], MOO can be used to figure out important players who gain from political campaigns, while in finance [37, 76, 90, 102], it can be used to spot noteworthy technical analysis trends in time series of financial data. Advancements in the field of biotechnology have also been cited in [58]. Here, MOO has been applied to optimize the fisheries bio-economic model by minimizing waste, maintaining quota shares, and maximizing profits.

There are different settlement methods that are used for solving MOO problems. One of these methods is the global criterion method [63], which aims to transform multiple optimization problems into a single optimization problem by reducing the gap between several reference points and feasible solutions. Another method is the weighted-sum method [13, 20, 45, 61, 62, 69, 91], which combines all of the problems into a single problem using a weighted vector. However, choosing the weights for
problems with different magnitudes can be challenging and may lead to bias. In cases where a plural problem that is being optimized is not convex, the $\epsilon$-constraint method [35] is utilized. This approach optimizes one problem while transforming other problems into constraints or restrictions.

The lexicographic method [25] is used to optimize objectives by prioritizing their order of importance. Each objective is optimized individually, starting with the most vital goal. If only one solution is returned, it is considered to be the best solution. If not, the optimization continues on the next objective under new restrictions that are based on the solution from the first objective. The goal programming method [8–10, 39, 65, 85] involves determining an objective function's ambition level to be achieved. Multi-objective evolutionary algorithms (MOEAs) [7, 67] are stochastic optimization techniques that are used to find optimal Pareto solutions. Most of the time, MOEAs use dominance relationships in their actions; their optimization mechanism is similar to that of evolutionary algorithms. MOEAs can also apply conventional support techniques such as niching due to the existence of objective space. Various MOO settlement methods have been reviewed, which involve solving complex equations.

In this paper, survey work has been done on multi-objective-based parameter optimization for deep learning focusing on various applications like healthcare, language processing, machinery, and others.

2. Deep learning

Deep learning is a particular kind of machine learning that utilizes artificial neural networks (ANNs) [36, 41, 99] with multiple layers (also known as deep neural networks [DNNs] [47, 48, 78, 80]). These networks are designed to model and process non-linear relationships by taking inspiration from the anatomy and physiology of the human brain. They are capable of learning from vast amounts of data in unsupervised or semi-supervised manners. The layering of neurons makes up a DNN in its compacted form, where the neurons resemble those that are found in the brain and the connections among them. These neurons receive input, process it, and pass on signals to other neurons; this forms a sophisticated network that improves with experience. The given diagram in Figure 1 illustrates a deep neural network (DNN) that is made up of several layers (‘N’ layered) of artificial neurons. As can be seen in Figure 1, the first layer’s neurons process the input data, which is then passed on to the following layer, etc. until the final output is produced. One or more neurons may be present in each layer; these neurons compute an extremely tiny function called an activation function. If the incoming neurons’ result exceeds a certain cutoff, the output is forwarded to the next connected neuron. A weight is associated with the connection between two neurons in successive layers that shows the effect of the input on the output. These weights are iteratively changed during model training to discover the best way to predict the desired outcome. The logical building blocks of a neural network include a neuron, layer, weight, input, output, and activation function as
well as a learning mechanism (optimizer) that helps the neural network gradually change the weights for better predictions of outcomes. Deep learning employs several architectures, including feed forward neural networks (FNNs) [4], convolutional neural networks (CNNs) [70], and recurrent neural networks (RNNs) [60]. FNNs follow a simple linear information flow through a network and have been employed in natural language processing, spoken word identification, and picture classification. CNNs are specialized FNNs that have been designed specifically for image and video recognition that is capable of automatically learning image features and is useful for object-identification, picture-classification, and image-segmentation tasks [26]. RNNs are ideal for processing sequential data like time series and natural languages and can maintain an internal state that captures information about previous inputs, making them well-suited for tasks such as speech recognition, natural language processing, and language translation.

![Figure 1. Deep neural network with ‘N’ hidden layers](image-url)

### 3. Multi-objective optimization (MOO) methods

Single-objective (SO) optimization (as in [17, 29, 79]) refers to the method of searching for the optimal solution that maximizes or minimizes a single objective function. Although this kind of optimization is a helpful tool for giving decision-makers an understanding of an issue, it frequently does not offer alternate solutions that balance various goals. Instead, it lumps all objectives into one.

As suggested by Guantara et al. [30], multi-objective optimization (MOO) is a process of finding optimal solutions for more than one desired goal. MOO is a useful technique in optimization because it simplifies a problem by not requiring complicated
equations. In MOO, decision-making involves a balancing act between various opposing issues. The concept of MOO was created by Vilfredo Pareto in 1896 [95]; it involves the objective function vector where each vector is the solution vector function. There is not a single optimum solution in MOO for all objectives, but each needs to be evaluated for trade-offs.

The MOO problem mathematical equation can be stated as follows [22]:

\[
\begin{align*}
\text{min/} & \text{max } f_1(x), f_2(x), \ldots, f_n(x) \\
\text{subject to : } x & \in \cup
\end{align*}
\]  

(1)

where ‘x’, ‘n’, \( \cup \), \( f_n(x) \), and ‘min/max’ denote a solution, the number of objective functions, a feasible set, the \( n^{th} \) objective function, and the combined object operations, respectively.

A decision variable solution vector and a multi-dimensional objective function vector both have spaces in MOO. There is a corresponding point in the space of the objective function for each solution in the space of the decision variable. The relationship between the two spaces is depicted in Figure 2 (as in [16]).

![Figure 2. Mapping between spaces of solution and objective function [30]](image)

The convexity of the spaces of the solution and objective function are important when choosing which algorithm will be employed to solve a problem. If all of the objective functions and solution regions are convex, MOO problems are considered to be convex. If the objective function satisfies Equation 2, then it is convex [6]:

\[
f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)
\]  

(2)

with Value 1 and \( f \) in the \( x, y \) domain. For a better understanding of Equation 2, it can be deduced that the \( f \) graph lies above the line that joins \((x, f(x))\) and \((y, f(y))\) from \(x\) to \(y\) (as shown in Figure 3).
There are two different categories of MOO problem solutions; namely, the Pareto method, and the scalarization method [15]. The Pareto method is employed when the targeted outcomes and performance measures are distinct and get balanced (which is presented in Pareto optimal front [POF] form). On the other hand, the scalarization method involves the use of the performance measures to form a scalar function (which incorporates the fitness function) [31]. Both methods are explained in the following section.

### 3.1. Pareto method

The following is a Pareto-based mathematical equation for the MOO problem [23]:

\[
\begin{align*}
    f_{1,\text{opt}} &= \min f_1(x) \\
    f_{2,\text{opt}} &= \min f_2(x) \\
    &\quad \vdots \\
    f_{n,\text{opt}} &= \min f_n(x)
\end{align*}
\]

During optimization, the Pareto technique maintains the components of the solution vectors separately (independently), and the idea of dominance is used to distinguish between dominated and non-dominated solutions. In MOO, the dominant solution and optimal value are often reached when one objective function cannot be increased without decreasing the other objective function; this state is known as Pareto optimality. A Pareto optimum solution is the name that is given to a collection of ideal solutions in MOO. A non-dominated solution (often known as Pareto efficient) is a concept in mathematics. Non-Pareto optimum solutions are those in which one objective function can be increased in such a way that the other’s objective function is not affected. This answer is referred to as a dominating solution (worse). Once a Pareto optimal solution is obtained, it can be solved mathematically [23]. This approach requires taking notice of a number of terms in the Pareto optimum solution; these are the terms:

(a) **Anchor Point**: through use of objective function, anchor points can be found;
(b) **Utopia Point**: meeting point of maximum/minimum value of one objective function with maximum/minimum value of another objective function yields utopia point.
A Pareto optimum front (POF) on a two-dimensional surface can illustrate the optimization with two objective functions and the non-dominated solution [11]. Consider optimizing the $f_1(x)$ and $f_2(x)$ objective functions.

Figure 4 shows the dominated solution (p7, p8,..., p21) and the non-dominated solution (p1, p2, p3, p4, p5, and p6) [32, 73]. As shown in Figure 5 under curves (a), (b), and (c), respectively, the POF may also exist in three distinct combinations for minimizing $f_1(x)$ and maximizing $f_2(x)$, maximizing $f_1(x)$ and minimizing $f_2(x)$, and maximizing both $f_1(x)$ and $f_2(x)$. The Pareto optimum solutions’ solution sites are depicted in Figure 4.

![Figure 4](image1.png)

**Figure 4.** POF of two objective functions (Guntantara, [30])

![Figure 5](image2.png)

**Figure 5.** Two other objective functions’ POFs (Guntantara, [30])

Dominated solutions and non-dominated solutions can be found through a comparison of two solution points for all solution points. For example, the p3
solution is said to be dominant over the p9 solution if the following two conditions are met [16]:

(a) for all objective functions, p3 solution is not significantly worse than p9 solution;
(b) in terms of one or more than one objective function, p3 solution triumphs over p9 solution.

Once the utopia point has been identified, the shortest Euclidean distance may be used to estimate the POF’s ideal value [71]. The POF form in three-dimensional space can be used to express non-dominated solutions for three objective functions. The non-dominated solutions, however, cannot be seen in the POF if the optimization is comprised of more than three objective functions [73]. Use the ‘continuously updated’ approach to look for non-dominated solutions. The quest for non-dominated solutions is ongoing with this strategy. The ‘continuously updated’ technique can be explained as follows [16]:

(a) path chosen for start is not dominated by $P' = 1$, put counter $i$ to 2
(b) put $j = 1$
(c) in order to find solution that is more dominant, solutions $i$ and $j$ from $P'$ were compared
(d) if solution $i$ outperforms solution $j$, then remove number of $-j$ from $P'$; if $j$ is smaller than $P'$, multiply it by one and return to step c; return to step e if contrary is true; if member number $-j$ from $P'$ dominates member number $i$, add one to $i$ and return to step b
(e) replace $P'$ with answer $i$ or make $P' = P'i$; add $i$ with one and return to step b if $i < N$ (where $N$ is number of solutions); if not, procedure halts and $P'$ is designated as non-dominated set (POF is composed of non-dominated set)

The utopia point is then discovered following the completion of the Continuously Updated algorithm. The ideal value is determined using the utopia point. The shortest Euclidean distance based on Equation 4 may be used to obtain the ideal value [12]. The following equation may be used to obtain the shortest Euclidean distance between the utopia point and the POF’s points [12]. After the Continuously Updated algorithm is completed, the utopia point can be reached. This point can be used to establish the ideal value. Based on Equation 4, the shortest Euclidean distance may be determined [12] to obtain the best value. The equation may be applied to get the shortest Euclidean distance between the utopia point and the POF’s points [12]:

$$d_E = \min \sqrt{\left(\frac{Q_1 - Q_1^*}{Q_{1\text{norm}}}\right)^2 + \left(\frac{Q_2 - Q_2^*}{Q_{2\text{norm}}}\right)^2}$$  \hspace{1cm} (4)$$

where (using Figure 4 as an example) $(Q_1^*, Q_2^*)$ are the point coordinates on the POF, $(Q_1, Q_2)$ are the normalization point coordinates in the problematic regions, and $(Q_{1\text{norm}}, Q_{2\text{norm}})$ are the coordinates for the normalization point in the utopia points of the objective function $f_1(x)$ whose minimum value is sought. Based on the minimal values of $Q_1$ and $Q_2$, respectively, $Q_{1\text{norm}}$ and $Q_{2\text{norm}}$ are calculated.
3.2. Scalarization method

The multi-objective function generates a single solution using the scalarization technique. Before the optimization process, the weight is chosen. This approach integrates multiple objective functions into a scalar fitness function using the following equation [68]:

\[ F(x) = w_1 f_1(x) + w_2 f_2(x) + \cdots + w_n f_n(x) \] (5)

An objective function weight can determine the fitness function solution and can show the performance priority [21]. The weight that is assigned to each objective function in a scalar fitness function determines the priority of each function in the solution. When an objective function is given a higher weight, it gains a greater priority when compared to items of lesser weights. There are three ways to determine the scalarization weight: equal weights, ROC weights, and RS weights [27]. Equal weights can be calculated by using the following equation [14]:

\[ w_i = \frac{1}{n} \] (6)

where \( i = 1, 2, 3, \ldots, n \); given that \( n \) is the number of objective functions. For ranking different criteria, rank-order centroid (ROC) weights are computed using the equation in [24]:

\[ w_i = \frac{1}{n} \sum_{k=i}^{n} \frac{1}{k} \] (7)

Each criterion is given a proportionate weight using rank-sum (RS) weights. The equation below can be used to calculate RS weights [24]:

\[ w_i = \frac{2(n + 1 - i)}{n(n + 1)} \] (8)

In this technique, the minimizing and maximizing functions are marked as negative and positive, respectively. To make objective functions fair in the scalarization method, it is important to normalize them using the root mean square [33]. For the three objective functions, a scalarization example is given below:

\[ F(x) = -\frac{w_1 f_1(x)}{\sqrt{E(f_1^2(x))}} + \frac{w_2 f_2(x)}{\sqrt{E(f_2^2(x))}} - \frac{w_3 f_3(x)}{\sqrt{E(f_3^2(x))}} \] (9)

where \( F(x) \) is the fitness function, \( f_1(x), f_2(x), f_3(x) \) are objective functions 1, 2, and 3, respectively, and \( w_1, w_2, w_3 \) are the corresponding weights. For checking the overall solution, the exhaustive method is used in MOO to determine the optimal value. Certain algorithms (such as ant colony optimization, particle swarm optimization [PSO], and meta-heuristic algorithms [like GA, etc.]) can determine the optimal value in order to assist in the optimal solution-finding process for a large solution.
4. Some recent works of deep-learning methods with MOO-based parameter optimization

In this section, we provide a summary of various works of literature as case studies. Deep-learning methods and their corresponding frameworks using MOO strategies have been classified in Figure 6.

![Figure 6. Classification of deep-learning model-based MOO methods](image)

The following sections provide an overview of all of the methods that are discussed. We surveyed about 23 papers that were mainly obtained from two major sources (including Google Scholar and Scopus) during the period of 2016 through 2022. We filtered the commonly appearing works in both search spaces based on the following keywords: deep learning, neural networks, multi-objective optimization, and parameter optimization. We classified the models using single-model architectures, followed by ensemble and surrogate models. The figure contains the key features and applications of the methods (sorted chronologically).

4.1. Single model-based architectures

In this section, stand-alone DNN architectures that use MOO-based methods are provided in Table 1, followed by two detailed discussions of a few of the methods.
4.1.1. Multi-objective-based differential optimization for CNN models

The manual tuning of the parameters forms one of the drawbacks for CNN models in achieving high performance. MODE-CNN [40] focuses on the CNN-based optimization
of parameters using the multi-objective differential evolution (MODE) algorithm. This has mainly been developed for image analysis in the healthcare domain. The accuracy of the model is regulated by three parameters, which include patch accuracy, general stride, and neighbor distance. Segmentation loss $SL$ is calculated by the following:

$$SL = (1 - \frac{1}{m} \sum_{i=1}^{m} \frac{1}{a} \sum_{j=1}^{a} (IOU_j))$$

(10)

where $IoU$ is the ratio of the total number of pixels in the image to the number of pixels where the item that is estimated by using ground truth intersects, $a$ is the total count of objects in the image, $m$ is the total number of images, and $a$ is the total count of objects in the image.

What follows is the algorithm that is provided.

### Algorithm of MODE-CNN

1: MODE-CNN initial values are created
2: Appropriate values of general stride (ST), neighbor distance (DIS), and patch accuracy (PAC) with MODE-CNN are created
3: Segmentation error and test time with CNN-based method are discovered
4: if the desired iteration has been reached then
5: return ST, DIS, and PAC values
6: else
7: go to 2
8: end if

### Characteristics

- The score for each individual in a population is calculated using the crowding distance and Pareto front numbers.
- Roulette wheel-selection technique is used for parent selection, which in turn helps to achieve fast convergence.
- Minimum segmentation loss and minimal test time have been achieved based on the above-mentioned parameters using this algorithm.
- Optimization using this algorithm is achieved in fewer iterations.
- This was demonstrated to be a robust and competitive algorithm when compared to other popular multi-objective optimization algorithms like NSGA-II, MODE, and others.

### 4.1.2. MOO chaotic butterfly optimization with DNN

To perform clustering for diagnosing diseases, the MOO-based chaotic butterfly optimization algorithm with deep neural network (MOCBOA-DNN) technique was developed [1]. First, a fitness function is used to choose the best set of cluster heads (CHs) and arrange the clusters while clustering the IoT medical devices. Second, the
cloud server receives the gathered medical data for further analysis. The healthcare data is then analyzed by the DNN model to determine whether a condition truly exists.

What follows is the butterfly optimization algorithm, which is used to find clusters for the IOT devices. The fragrance \( f \) estimate for the optimization strategy can be written as in Equation 11, where \( c \) is the sensory modality, \( I \) is the intensity of the stimuli, and \( \beta \) is the power exponent value. These parameters play a significant role where \( \beta \) varies between 0 and 1. A value of 1 denotes that the neighboring butterfly can sense full fragrance. In an ideal environment, there is no absorption of fragrance:

\[
f = cI^\beta
\]  

(11)

The global search where the butterflies move toward the best butterfly can be defined as in Equation 12, where \( g^* \) is the overall highest value that is obtained among all of the solutions in the current iteration, and \( r \) is a generated random number:

\[
y_i^{t+1} = y_i^t + (r^2 \times g^* - y_i^t) \times f
\]  

(12)

**Algorithm** of BOA

1: Generate a population of \( n \) butterflies \( y_i = (i = 1, 2, \ldots, n) \).
2: Initialize sensor modality \( c \), switch probabilities \( p \), and power exponents \( \beta \).
3: while end criteria remains unsatisfied do
4: for all butterflies \( b_f \) in the population do
5: Do fragrance estimation as in Equation 11
6: Choose best butterfly based on the best solution \( (g^*) \)
7: end for
8: for all butterflies \( b_f \) in the population do
9: Generate \( r \) between 0 and 1
10: if \( r < p \) then
11: Moves the optimum butterfly near best butterfly \( (g^*) \) as in Equation 12.
12: else
13: Moves arbitrarily
14: end if
15: Evaluate for a novel butterfly
16: Upgrade the population when optimum is obtained
17: end for
18: Upgrade the value of \( c \)
19: Assign the best overall solution \( (g^*) \)
20: end while
21: Print the optimum solutions \( (g^*) \)

**Characteristics**

- Using gathered healthcare data, the MOCBOA-DNN technique may cluster IoT devices for healthcare and diagnose diseases, which can lead to more efficient and effective healthcare management systems.
- The best set of cluster heads is chosen using a fitness function for the clustering of IoT medical devices.
• The collected healthcare data present in the cloud storage is analyzed further to check for the presence of any disease.
• When it comes to a diverse variety of evaluation components, the MOCBOA-DNN technique can outperform other current techniques, which indicates its effectiveness.

4.1.3. MOO using grid-search optimization with DNN

Based on deep-learning optimization models, this framework was created for medical data sets with high percentages of missing values [66]. The data missing care (DMC) framework (which addresses the issue of excessive missing data in medical records) is used to increase the model’s resilience. By adjusting multiple hyperparameters, ANN (artificial neural network), CNN (convolutional neural network), and recurrent neural network-based (RNN) deep-learning algorithms are tuned, and grid-search optimization is utilized to create a better deep predictive training model for patients with COVID-19.

What follows is an algorithm for resolving the issue of a large amount of missing data in medical databases. In the pre-processing stage, the data is normalized using the following Equation 13, where $p_i$ denotes the original feature value, and the minimum and maximum feature values are indicated by $\min(p)$ and $\max(p)$, respectively:

$$p_i^{\text{new}} = \frac{p_i^{\text{old}} - \min(p)}{\max(p) - \min(p)}$$

Algorithm of Multi-Objective Deep-learning Framework
1: Data normalized using Equation 13
2: Implement ANN, CNN, and RNN models
3: Perform hyperparameter tuning using grid search optimization
4: Evaluate prediction results using following:
5: $\text{Accuracy} = \frac{TP + TN}{(TP + TN) + FP + FN} \times 100$
6: $\text{Precision} = \frac{TP}{(TP + FP)} \times 100$
7: $\text{Recall} = \frac{TP}{(TP + FN)}$
8: $F1 \text{ Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
9: Terminate

Characteristics
• Using the grid optimization technique, the deep-learning models’ hyperparameters were optimized.
• The analysis used the following hyperparameters: the number of layers, neurons, activation function, momentum, loss function, learning rate, batch size, and epochs.
• This method can handle the imbalanced data sets that are commonly found in medical data sets, and the F1 score is used as the ultimate accuracy measuring metric.
The RNN model performed best when compared to the CNN and ANN models.

This technique may be utilized in future studies in the fields of traffic control, electric power networks, financial companies, and other sectors that work with high-dimensional data sets and need rapid data processing.

4.1.4. Multi-objective evolution of ANN for medical diagnosis

A selection mechanism that is based on the hypervolume indicator is combined with a quick approximate algorithm in the covariance matrix adaptation Pareto achieved evolution strategy with hypervolume sorted adaptive grid algorithm (CMA-PAES-HAGA) [75] to explore a problem space. With the use of variation operators, CMA-PAES-HAGA is able to converge toward the true Pareto-optimal front [81] while preserving a diverse population of solutions during the optimization process.

What follows is the covariance matrix adaptation Pareto achieved evolution strategy with hypervolume-sorted adaptive grid algorithm:

\begin{algorithm}
\textbf{Algorithm of CMA-PAES-HAGA}
\begin{algorithmic}
\STATE Generation counter and extreme value vector are initialized
\STATE $t \leftarrow 0$
\STATE $Z \leftarrow (\varepsilon_1 = 0, \varepsilon_2 = 0, \cdots, \varepsilon_M = 0)$
\STATE Parent population is initialized, where X contains the solutions in the search space and Y contains the vectors of the objective values
\STATE Initialize parent population $Y, X$
\WHILE {not met termination criteria}
\FOR {$j = 1, \cdots, \lambda$}
\STATE $X'_j \leftarrow X_j$
\STATE $X'_j \leftarrow X'_j + \sigma_j \cdot N(0, C_j)$
\STATE Check if solution is within bounds
\IF {$X_i^{(L)} \leq X'_{ij} \leq X_i^{(U)}$}
\IF {$X'_{ij} > X_i^{(U)}$}
\STATE $X'_{ij} = X_i^{(U)}$
\ELSE
\STATE $X'_{ij} = X_i^{(L)}$
\ENDIF
\ELSE
\ENDIF
\ENDFOR
\STATE Evaluate solution
\STATE $Y'_j \leftarrow f(X'_j)$
\STATE $Y^* = Y \cup Y'$
\STATE Update extreme values
\FOR {$m = 1, \ldots, M$}
\IF {$Y^*_{mj} > \varepsilon_m$}
\STATE $\varepsilon_m = Y^*_{mj}$
\ENDIF
\ENDFOR
\STATE Selection routine
\STATE $Y, X \leftarrow \text{HypervolumeSortedAGA}(Y^*, Z)$
\STATE \text{// variation routine}
\STATE \text{CMAParameterUpdate}()
\STATE $t \leftarrow t + 1$
\ENDWHILE
\end{algorithmic}
\end{algorithm}
Characteristics

- This method addresses class-imbalance concerns without requiring the prior integration of knowledge that is particular to the problem.
- In the case of a multi-class classification problem, it takes the trade-offs between the classification accuracy of each class into account.
- In the case of a multi-class classification problem, it takes the trade-offs between the classification accuracy of each class into account.
- The minority class recognition is also improved without making assumptions about misclassification costs.
- This also presents a decision-maker with a variety of trained ANNs with trade-offs that are evenly dispersed throughout the Pareto front.

4.2. Ensemble model-based architectures

In this section, we discuss the different ensemble models that use MOO-based parameter optimization. Table 2 enlists the ensemble methods that use MOO-based optimization along with deep-learning algorithms in a sequential or integrated form. We discuss some of the methods in detail with their working frameworks and areas of application, while all if the other methods are included in Table 2.

<table>
<thead>
<tr>
<th>Year</th>
<th>Algorithm/Model</th>
<th>Key Features</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019</td>
<td>Ensemble deep learning with MOO for prediction of treatment outcomes [93]</td>
<td>- Trains with deep perceptron models to handle issues of EHR data</td>
<td>- Prediction of treatment risks after radiotherapy for lung cancer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Ensemble strategy with adaptive multi-objective optimization and evidential reasoning (ER) fusion used</td>
<td>- Identifies features like tumor size, regional dose, T staging, N staging through feature-importance analysis</td>
</tr>
<tr>
<td>2021</td>
<td>Multi-objective ensemble deep learning for prognosis of rotating machinery [57]</td>
<td>- Monitoring data collected by prognostics and health management (PHM)</td>
<td>- Monitoring data used to calculate remaining useful life (RUL) of mechanical components</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Applied in PHM systems using smart sensing techniques and the Internet of Things (IoT)</td>
</tr>
<tr>
<td>2021</td>
<td>Character recognition-based aquila optimizer and DNN [83]</td>
<td>- Fuzzy filtering technique for image pre-processing</td>
<td>- Character recognition, where handwritten and printed text coexist in same document</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Fusion of EfficientNet and CapsuleNet DNN models for feature extraction</td>
<td>- Digitization of historical documents that contain Telugu characters</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Bi-directional long short-term memory (BiLSTM) model with aquila optimizer is used</td>
<td>- Accessibility for visually impaired</td>
</tr>
<tr>
<td></td>
<td>Breast lesion assessment using combined DNNs and MOO-based seagull optimization [77]</td>
<td>- Convolutional and recursive neural networks are combined</td>
<td>- License plate recognition, signboard recognition, and text translation</td>
</tr>
<tr>
<td>2022</td>
<td></td>
<td></td>
<td>- Identification and categorization of breast lesions</td>
</tr>
</tbody>
</table>
Table 2 (con’t)

<table>
<thead>
<tr>
<th>Year</th>
<th>Algorithm Description</th>
<th>Task Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2022</td>
<td>Multi-objective grasshopper optimization algorithm (MOGOA) [28]</td>
<td>- Two pre-trained CNN architecture (such as InceptionV3 and ResNet50) were used - Classifying non-COVID-19, COVID-19, and pneumonia patients using chest X-ray images</td>
</tr>
<tr>
<td>2022</td>
<td>Short-term wind-speed forecasting based on deep learning and multi-objective parameter optimization [92]</td>
<td>- Integrates several single-model forecasting results through weight-optimization operator - Able to provide both point prediction and uncertainty forecasting - Provides accurate and real-time wind power information - Useful for planning and managing wind power projects - May be applicable in aviation or marine transportation</td>
</tr>
<tr>
<td>2022</td>
<td>Multi-objective mayfly optimization with DenseNet [84]</td>
<td>- Involves DenseNet-169 as feature extractor - Uses functional link neural network (FLNN) as classification model - Assistive technologies for visually impaired people - Automated registry for business documents - Real-time handwriting recognition in smartphone environment</td>
</tr>
<tr>
<td>2022</td>
<td>Multi-objective quantum swarm optimization with DNN [2]</td>
<td>- Involves optimized region growing-based segmentation - Involves capsule network-based (CapsNet) feature extraction - Involves extreme learning machine-based (ELM) classification - Diagnoses dystrophinopathies using muscle magnetic resonance imaging (MRI) images</td>
</tr>
</tbody>
</table>

4.2.1. Multi-objective mayfly optimization with DenseNet

As a feature extractor, a DenseNet-169 model is utilized in order to generate a set of beneficial feature vectors in the multi-objective mayfly optimization with deep learning (MOMFO-DL) technique, and a functional link neural network (FLNN) is used for classification to identify and categorize the handwritten characters [84]. The DenseNet and FLNN models’ parameters are optimized using the MOMFO method. The overall architecture is shown in Figure 7.

![Figure 7](image-url)
Recognition of handwritten characters has gained prominent attention in recent times due to its wide application in various technologies, like providing assistance to visually impaired people, automating the registry of business documents, and others. This particular work focuses on Telugu handwritten character recognition.

What follows is the mayfly optimization algorithm:

\textbf{Algorithm} of Mayfly Optimization

1: Male and female mayfly population is initialized
2: Upgrades of the velocities and solution
3: Upgrades for male mayflies
4: \textbf{if} $f(b_i) > f(b_{hi})$ \textbf{then}
5: \hspace{1em} $s_i(t+1) = g \cdot s_i(t) + a_1 e^{-\beta r^2_p} [b_{hi} - b_i(t)] + a_2 e^{-\beta r^2_q} [b_g - b_i(t)]$
6: \textbf{else}
7: \hspace{1em} $s_i(t+1) = g \cdot s_i(t) + d \cdot r_1$
8: \textbf{end if}
9: where
10: $f(b_i)$ indicates present fitness value
11: $f(b_{hi})$ indicates optimal fitness value in trajectories
12: $g$ indicates linear declination of small one from the maximal values
13: $a_1, a_2,$ and $\beta$ are constants that are used to balance the values
14: $r_p$ and $r_g$ denote the parameters that are used for calculating Cartesian distance among the male individuals and their finest global and previous positions in swarm
15: $d$ is an arbitrary dance coefficient
16: $r_1$ indicates a random quantity in uniform distributions from $[1, 1]$
17: Upgrades for female mayflies
18: \textbf{if} $f(c_i) < f(b_i)$ \textbf{then}
19: \hspace{1em} $s_i(t+1) = g \cdot s_i(t) + a_3 e^{-\beta r^2_m} [b_i(t) - c_i(t)]$
20: \textbf{else}
21: \hspace{1em} $s_i(t) = g \cdot s_i(t) + f_l \cdot r_2$
22: \textbf{end if}
23: where
24: $a_3$ indicates additional constant
25: $r_m$ indicates Cartesian distance among female individuals for and their finest global and previous positions in swarm
26: $f_l$ is additional arbitrary dance
27: $r_2$ indicates a random quantity in uniform distributions from $[1, 1]$
28: Mayflies are ranked
29: Mayflies are mated, and offspring are evaluated
30: Worst solutions are replaced with best new one
31: \textbf{if} Met is terminated \textbf{then}
32: \hspace{1em} End
33: \textbf{else}
34: \hspace{1em} go to 2
35: \textbf{end if}
Characteristics
- The new MOMFO-DL model attempts to enhance the understanding of handwritten Telugu characters.
- Pre-processing, feature extraction, classification, and parameter optimization are only a few of the various steps of the operations that the model entails.
- Most of this approach’s inspiration comes from the behaviors of mayflies. The functioning of the female and male mayflies is implemented for the optimizations, much like the swarm optimization that is connected to swarm individuals. As a result, the classification process has been improved by the application of the mayfly optimization method.
- The results of the experiment showed better recognition performance with high accuracy. These findings might be applied to feature-selection and segmentation strategy-design in the future; they may also be useful in assisting smartphone users.

4.2.2. Multi-objective quantum swarm optimization with DNN
The MOQTSO-DL (multi-objective quantum tunicate swarm optimization with deep learning) model by [2] includes four steps: segmentation (based on optimized regional growth), feature extraction (based on CapsNet), classification (based on an extreme-learning machine), and parameter optimization (based on MOQTSO). This algorithm is mostly used to diagnose dystrophinopathies; these form one of the most commonly inherited muscular diseases across the globe. For this study, muscle MRI images are utilized, and the region of interest (RoI) detection method is predominantly carried out by using an optimized region-growing approach. The feature vectors are extracted by utilizing the CapsNet model. The ELM classifier then uses the retrieved feature vectors as inputs to derive the appropriate class labels. The MOQTSO technique is primarily used to choose the first RoI detection seed sites and to fine-tune the ELM model’s parameters. Figure 8 refers to the overall process of the MOQTSO-DL model.

![Overall framework of MOQTSO-DL method](image-url)
Algorithm of MOQTSO

1: Initialization of population of tunicates $P^*_p$
2: Original value for parameter and maximum iterations is set
3: Fitness value of each exploration agent is calculated
4: After fitness evaluation, find optimal entity in search space Calculate state fitness cost of upgraded search agent
5: Each exploration agent’s location is upgraded
6: Novel upgraded agent is returned to its borders
7: Fitness cost of upgraded search agent is calculated
8: if solution is more optimal than previously then
9: $P^*_p$ is upgraded, and optimal solution is stored in $X_{best}$
10: end if
11: if end condition is not encountered then
12: Iterate Steps 5 to 8
13: end if
14: Optimum solution has been attained thus far ($X_{best}$) is stated

Characteristics

• The entire method starts with region of interest (RoI) detection, followed by feature extraction using the CapsNet model.

• This method helps emphasize the relationships among the regions of the image and is, hence, very useful for feature extraction.

• The feature vectors that are extracted are then fed to the ELM model for classifying the muscle MRI images.

• Finally, the tunicate swarm optimization (TSO) method is used for parameter selection. The tunicate swarm optimization algorithm is described below.

• The MOQTSO algorithm reduces the time complexity as compared to traditional random parameter-searching processes.

• This results in maximal performance with high accuracy when classifying Duchenne muscular dystrophy (DMD) and Becker muscular dystrophy (BMD) versus non-dystrophinopathies.

• This method can be extended for designing lightweight deep-learning models and hyper-parameter optimizers like Adam and others to reduce space-time complexity.

4.3. Surrogate Model-based Architectures

In this section, we mention surrogate model-based frameworks using parameter optimization. Table 3 contains the various methods used in the different applications of machinery.
<table>
<thead>
<tr>
<th>Year</th>
<th>Algorithm/Model</th>
<th>Key Features</th>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019</td>
<td>DL-based aerodynamic design optimization [89]</td>
<td>- Multi-fidelity-based optimization is used</td>
<td>- Aerodynamic design optimization under uncertainty of Mach number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Deep belief network is employed as low-fidelity model</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>- K-step contrastive divergence algorithm is used for training</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2020</td>
<td>Design of magnet synchronous motor for electric vehicle based on MOO and DL [100]</td>
<td>- Uses multi-layer perceptron (MLP) for shape optimization of motors</td>
<td>- Shape optimization of permanent magnet synchronous motor (PMSM)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Utilizes finite element analysis to design experiments</td>
<td>- Maximize PMSM performance</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2021</td>
<td>Aerodynamic prediction using dual CNN and optimization of turbine rotor [94]</td>
<td>- Dual-CNN for aero-engine turbines</td>
<td>- Field reconstruction and performance prediction for compact turbine rotor in aerospace</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Gradient-based MOO with efficiency and torque as objective functions</td>
<td>- Real-time adjustment direction for operation and maintenance of rotor</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>- Predicts pressure and temperature distribution in large gradient areas at leading and trailing edges of blade</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2022</td>
<td>MOO of machining process parameters with DNN [96]</td>
<td>- Uses DL for data-driven prediction of optimized objectives</td>
<td>- Making improved product quality, efficiency, and reduced environmental impact</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Uses technique for order preference by similarity to ideal solution (TOPSIS)</td>
<td>- Guide operator for machining process parameter selection</td>
</tr>
</tbody>
</table>

### 4.3.1. MOO of machining process parameters with DNN

Production efficiency and the environmental impact of the machining process are hugely impacted by machining process parameters [96]. The existing research employs costly physical models and computationally intensive numerical simulations that are ineffective and inaccurate during the real exploitation stage. For the multi-objective optimization of machining process parameters, a deep learning-based genetic algorithm as well as TOPSIS (technique for order preference by similarity to the ideal solution) is employed. The overall framework is shown in Figure 9. DNN first creates the data-driven prediction function for various optimized objectives. The created prediction function is subsequently transformed into a surrogate model and combined with the genetic algorithm in order to generate the Pareto set. The optimal processing parameter is then automatically found from the generated Pareto set using TOPSIS. The algorithm is provided below.

**Characteristics**

- NSGA-III is used as the MOO-based genetic algorithm is used for generating Pareto optimal solutions [19] [42].
- The machining-based optimization objectives involve conflicting goals, including the roughness of the surface, the rate of production, the maximal cutting force, and the energy consumption.
• Its primary advantages include cost-efficient generic objective functions.
• An end-to-end structure that minimizes human interference and provides fast optimization speeds that take only a few minutes.
• The future scope would include performance degradation, which would be invariable if using such advanced optimization strategies.

![Diagram of overall framework](image)

**Figure 9.** Diagram of overall framework [96]

**Algorithm of TOPSIS**

1: Normalized decision matrix (NDM) is constructed using \( r_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n} x_{ij}}} \), where \( r_{ij} \) indicates elements of NDM
2: Weighted NDM is constructed using \( c_{ij} = r_{ij} \times \omega_j \forall i, j \), where \( \omega_j \) indicates assigned weight to attribute \( j \)
3: Idea(\( X^+ \)) and negative-idea(\( X^- \)) solutions are determined using
4: \( \left\{ \left( \max_{j} c_{ij} \right) | i \in Y, \left( \min_{j} c_{ij} \right) | i \in Y' ; \forall j \right\} = \left\{ c_1^+, c_2^+, \ldots \right\} \)
5: \( \left\{ \left( \min_{j} c_{ij} \right) | i \in Y, \left( \max_{j} c_{ij} \right) | i \in Y' ; \forall j \right\} = \left\{ c_1^-, c_2^- , \ldots \right\} , \)
6: where \( Y \) and \( Y' \) are associated with benefit and cost attributes, respectively
7: Separation measure is calculated using \( S_i^+ = \sqrt{\sum_{i=1}^{n} (c_{ij} - c^+)^2} \forall j \), \( S_i^- = \sqrt{\sum_{i=1}^{n} (c_{ij} - c^-)^2} \forall j \)
8: Relative closeness to ideal solutions is calculated using \( C_j^+ = \frac{S_j^+}{S_j^+ + S_j^-} \), alternative rankings based on \( C_j^+ \) values
5. Applications

Multi-objective optimization is incorporated with deep-learning models in various fields; these are summarized below.

5.1. Text recognition

Aquila optimization using the adaptive fuzzy filtering [83] and multi-objective mayfly optimization with deep learning (MOMFO-DL) [84] techniques aims to detect and recognize handwritten Telugu characters. Character recognition is performed in the presence of both handwritten and printed text that coexist in the same document. It is also potentially helpful for creating assisted technologies for visually impaired people. The scope of such works may also be extended to the automated registration of documents, signboards, and license plate recognition.

5.2. Traffic management

In the case of traffic management, the main purpose is to detect traffic incidents automatically and actively control traffic based on predictions. A MOO framework that uses the particle swarm algorithm achieves traffic-flow forecasting for the next day to actively control and adjust traffic flow as well as develops plans for managing traffic [49].

5.3. Wind speed forecasting

The interval forecast model with improved NSGA plays a crucial role in operating and dispatching contemporary power systems by effectively quantifying uncertainties in wind power forecasting [101]. A MOO framework that uses the binary backtracking search algorithm is used in real-world wind speed forecasting [56]. A MOO framework that uses a data denoising strategy and a grey wolf optimizer assists in accurately forecasting short-term wind speeds, which mitigates the effects of wind speed fluctuations [92]. This aids decision makers in their planning and the operators of power grid systems in their dispatching of power systems in a timely manner; it also reduces the risk of failure in wind power systems and improves the overall power quality.

5.4. Mobile and embedded systems

The time-to-market and NRE costs for mobile and embedded systems promote the use of deep learning-based technologies. The design space exploration method with response surface modeling automates the artificial neural network’s design process, reducing time and costs [82]. A MOO framework with neural network pruning for exploring a design space is used to optimize DNNs in embedded systems, which are mostly restricted by constraints in memory and energy resources [53].
5.5. Machinery

By analyzing the monitored data that is collected by PHM systems, a MOO framework that employs GA is used to predict the remaining useful life (RUL) of mechanical components [57]. Surrogate-based modeling system-based frameworks are used mainly for aerodynamics and motor systems. A MOO framework that uses improved particle swarm optimization is utilized for the optimization of aircraft airfoils and wings when the Mach number is uncertain [89]. Again, finite element analysis that employs the metamodeling technique optimizes the design of permanent magnet-based synchronous motors for electric vehicles [100]. Another MOO framework that uses the automatic differentiation method optimizes turbine rotors by aerodynamic-performance prediction for aerospace engineering applications [94]. The data-driven genetic algorithm is used in mechanical manufacturing systems to select optimal machining process parameters, which can effectively enhance the production efficiency of the process and reduce its environmental impact [96].

5.6. Environmental control

Several environmental hazards can be overcome with the use of parameter optimization-based deep-learning methodologies. The deep data-driven framework with the swarm intelligence method improves combustion efficiency in coal-fired thermal power plants [88]. It also reduces NOx emissions, thereby reducing environmental pollution.

5.7. Healthcare

In healthcare, different types of analyses are involved based on the multi-modal data that is available. The covariance matrix adaptation-based evolution strategy with adaptive grid algorithm was used to classify fetal cardiocograms by optimizing the overall classification accuracy and individual target-class accuracy in [81]. This method has been used for the development of a support-based system for decision management for the computerized analysis of fetal cardiocograms. The valuable patient-specific data that is found in electronic health records (EHRs) can be used to enhance outcome prediction. Multi-objective ensemble deep learning (MoEDL) is devised to predict the probability of significant treatment failure following radiation in lung cancer patients [93]. Specifically, deep perceptron networks are utilized as base learners throughout the training phase to address a variety of EHR (electronic health record) data-related issues. For devices in healthcare, machine-learning algorithms have widespread applications. Clustering using the butterfly optimization technique has the objective of categorizing healthcare IoT devices into clusters while analyzing the collected healthcare data to diagnose diseases [1]. Missing data forms one of the commonly faced issues in the medical area of research. A MOO framework that uses grid-search optimization is utilized to address the shortcomings of high levels of missing data in medical data sets [66]. These methodologies have also been used in
speech-enhancement technologies. A MOO framework that uses an ideal binary mask and DNN is used to enhance speech signals [98].

5.8. Image processing

A MOO framework that uses differential evolution is used in the segmentation and classification of medical images [40]. Swarm optimization that uses quantum computing is used to diagnose dystrophinopathy diseases by employing a magnetic resonance imaging (MRI) tool [2]. Grasshopper optimization that uses a support vector machine automatically uses chest X-ray images to categorize patients under Non-COVID-19, COVID-19, and pneumonia categories [28]. CT scans and MRI images are also used for image analysis. Based on a combined CNN and RNN framework with optimized parameters, the multi-objective seagull optimization algorithm (BLIC-CRNN-MOSOA) can be used in mammography screening to identify and classify breast lesions. It provides three categories of output: (i) normal, (ii) benign, and (iii) malignant tumors [77]. The MOSOA method can be used to help CRNN in finding the best parameters and also for fine-tuning them.

6. Discussion and conclusion

The performance advantages that can be obtained by applying multi objective-based parameter optimization can be more fully utilized while deep learning is an active research topic. Deep learning currently has two significant limitations: problems with training data come first, followed by the explainability of these black-box models. The difficulty in mapping the input to the output results in identifying such DNNs as black boxes. Other challenges include noisy data, the problem of missing or incomplete data, and others.

Presently, there are many ongoing works for addressing these drawbacks. Many interpretation- and attribution-based explanation strategies are being proposed in the recent literature. In this survey, it is discussed how useful multi-objective optimization is for deep-learning parameter optimization. Some of the benefits of multi-objective optimizations include acquiring improved results with low errors, using smaller data sets for training phases, optimal architectures, and obtaining multiple final solutions where it is not dependent on an external constraint for only one solution to be deemed the best solution of the Pareto front. It has been seen that using such MOO-based parameter-optimization strategies in deep-learning models can improve their performance – especially for those cases with incomplete, missing, and noisy data.

To achieve performance gains, parameter optimization must be implemented correctly. As compared to single objective optimization methods, MOO methods overcome the problems by having lower variances, as they intelligently search parameter spaces. The widespread applications (particularly in the domains of healthcare and machinery) provide the very positive future utilities of such methods –
especially with embedded and surrogate models. Therefore, future research can focus on enhancing the performance of deep-learning models via MOO-based parameter optimization.

Acknowledgements

Sanghamitra Bandyopadhyay acknowledges JC Bose Fellowship Grant No. JBR/2021/000036 from SERB, Govt of India, in support of the partial fulfillment of this research work.

References


**Affiliations**

**Mrittika Chakraborty**
Jadavpur University, Kolkata, India, mrittikac.cse.rs@jadavpuruniversity.in

**Wreetbhas Pal**
Jadavpur University, Kolkata, India, wreetbhaspal@gmail.com

**Sanghamitra Bandyopadhyay**
Indian Statistical Institute, Kolkata, India, sanghami@isical.ac.in

**Ujjwal Maulik**
Jadavpur University, Kolkata, India, ujjwal.maulik@jadavpuruniversity.in, corresponding author

**Received**: 12.05.2023  
**Revised**: 13.05.2023  
**Accepted**: 13.05.2023