Opposition-based Multi-objective ADAM Optimizer (OMAdam) for Training ANNs

Farzaneh Nikbakhtsarvestani[1], Shahryar Rahnamayan,[2] and Mehran Ebrahimi[3]

***Abstract*—Multi-loss functions are present in various aspects of deep learning. In multi-modal, cross-modal, and multi-task learning contexts, multi-loss functions are essential elements for handling complex data with diverse information sources. Different tasks or modalities may have conflicting objectives. By combining them into a single loss function, the model might struggle to strike the right balance between these objectives, leading to suboptimal performance. The Multi-objective Adam optimizer, also referred to as MAdam, is an extension of Adam optimizer that is applied for optimizing several competing loss functions in deep learning. The MAdam algorithm exhibits sensitivity to its initialization, necessitating the injection of extreme points into the initial population. Additionally, this scheme encounters difficulties in effectively capturing the disconnected and non-convex Pareto fronts. In this paper, an opposition-based scheme was introduced into MAdam framework as global search is necessary for escaping local optima in gradient-based multi-objective optimization approaches. The Opposition-based MAdam, explores multiple directions over the landscape, that leads to independence from specific initialization. In a series of experiments, we demonstrate the scalability of our method by capturing the entire Pareto front using the MNIST dataset for binary classification of digit images 2 and 3. This was achieved with a fully connected network, employing multi-objective mean absolute error and binary cross-entropy as losses. OMAdam matches Adam's F1-score in the early generations, a result to its high exploratory capacity which enhances its performance in initial stages of classification tasks. This results in a reduction of computational costs compared to both Adam and MAdam. The variation in F1-score values along the Pareto front trajectory enables practitioners to select a post hoc solution based on the trade-offs achieved among conflicting loss functions as multiple objectives. This contrasts with Adam, which offers limited options due to its single-solution approach.**

***Index Terms*—Multi-objective Optimization (MOO), Multi-objective Adam (MAdam), Opposition-based Learning (OBL), Opposition-based MAdam (OMAdam), Non-Dominated Sorting (NDS), Crowding Distance (CD), Pareto Front (PF), Fully Connected Network**

I. Introduction

The challenge of identifying optimal trade-offs between competing objective functions is pervasive in many fields of research such as machine learning. MOO gained interest in the machine learning community thanks to the development of new gradient-based MOO algorithms that facilitate faster training times. The goal of using gradient information in MOO is to determine a step direction, which guarantees simultaneous improvement of all objective functions until Pareto front is attained. The earlier attempt to tackle MOO for Deep Learning focused on addressing multi-task learning through gradient descent to find a single solution on the Pareto front. Inspired by the success of Adam optimizer [1] and the importance of using MOO in deep learning, [2] proposed hybrid algorithm, MAdam, that modified and expanded Adam algorithm to develop a Adam-based MOO optimizer. In this scheme the population-based Adam is used for the exploitation to accelerate the convergence. Accordingly, non-dominated sorting (NDS) and crowding distance (CD) are used for the exploration and to find a better distribution of Pareto front solutions. The extensions of the work have the potential to be utilized for multi-loss optimization in deep learning. Madam algorithm depends on specific initialization and for this, the extreme point created by Adam optimizer is injected to the first population. Also, in Madam scheme for each objective, one direction is applied to navigate toward promising regions which may cause premature convergence and not finding diverse and high-quality solutions. By exploring along many directions across the landscape, gradient-based algorithms can have a better chance to avoid local optima and enhance the algorithm performance to find promising regions. OBL [4] is a tool to search both a direction and its opposite simultaneously in order to obtain a better approximation for the optimal solution. OBL provides the context for finding the unknown optimal solution, searching both a direction and its opposite simultaneously which gives a higher chance to achieve the promising regions. The underlying motivation for this work lies in integrating OBL (specifically opposite directions) into the framework of hybridized Madam mechanism which represents multiple directions of the first and the second moment of the two gradients in MAdam scheme. Based on multiple directions generated by reordering the weights of mean gradient (first moment) and variance (second moment) for each input variable, our proposed approach is called opposite directions Madam (OMAdam). OMAdam makes MAdam algorithm perform better with regard to population initialization and Pareto front solutions at training time. In OMAdam scheme, the first population is generated completely random (similar to single-objective Adam) and Pareto front solutions appear in earlier stages of running the algorithm due to multiple opposite directions strategy. Similar to its parent algorithm Madam, OMAdam is designed to solve population-

[1]Farzaneh Nikbakhtsarvestani, Author is with Faculty of Science, Computer Science, Ontario Tech University, 2000 Simcoe St N, Oshawa, ON L1G 0C5 `farzaneh.nikbakhtsarvestani@ontariotechu.net`

[2]Shahryar Rahnamayan, SMIEEE, Author is with Faculty of Mathematics & Science, Brock University, 1812 Sir Isaac Brock Way, St. Catharines, ON L2S 3A1 `srahnamayan@brocku.ca`

[3]Mehran Ebrahimi, Author is with the Faculty of Science, Ontario Tech University, 2000 Simcoe St N, Oshawa, ON L1G 0C5 `mehran.ebrahimi@ontariotechu.ca`

2024 IEEE Congress on Evolutionary Computation (CEC) | 979-8-3503-0836-5/24/$31.00 ©2024 IEEE | DOI: 10.1109/CEC60901.2024.10612083

based hybrid MOO that employs NDS and CD algorithms for simultaneous exploration of objectives in order to obtain uniformly distributed MOO solutions. Experimental results show our method is an efficient MOO approach that finds Pareto front for deep models on large multi-objective datasets. The OMAdam optimizer identifies a range of solutions at each generation, contrasting with the single-objective Adam and it accomplishes this at negligible training time compared to a MAdam. To sum up, this paper introduces the following contributions:

- A hybrid population-based method for MOO in training ANN to find the well distributed full set of Pareto front through high exploratory capacity,
- Novel penalty term to MAdam scheme to be not dependent on the specific initialization to get the search direction across the objectives' space,
- Reducing computation costs by finding high-performance model parameters in the early stages of optimization,
- Achieving F1-score equal to Adam's in terms of performance for the binary classification task,
- Variation in the entries of the confusion matrix along the Pareto front's trajectory introduces trade-off options for selection, in contrast to Adam's single-solution approach, where no selection options are available.

The rest of this article is structured as follows:
We review the concept of the OMAdam algorithm, the background, and the details of the algorithm component in Section II. We provide the suggested the algorithm's details of OMAdam optimizer in Section III. In this section, the pseudo-code for the OMAdam and opposite directions are provided. In Section IV, the suggested method is assessed and competes strongly with its parent algorithm MAdam and single-objective Adam in terms of exploration and exploitation across MOO benchmarks and on a fully connected network. We provide thorough concluding remarks in Section V.

II. Background Review

1) Multi-objective Optimization: MOO is frequently encountered in real world scenarios, signifying that the problem at hand involves a minimum of two conflicting objectives. Over the past two decades, MOO has gained considerable attention. It can be mathematically represented as follows:

$$\min F(X) = (f_1(X), f_2(X), \cdots, f_m(X))^T \quad s.t. \quad X \in \mathbb{R}^n \tag{1}$$

where $X \in \mathbb{R}^n$ is a decision vector. $F(X)$ is composed of m objective functions $f_i : \Omega \to R$, $\Omega \in \mathbb{R}^n$, $1 \leq i \leq m$ and the objective space is $\mathbb{R}^m$. A feasible solution $X^* \in \Omega$ of problem (1) is called a Pareto optimal solution, if and if $\nexists X \in \Omega$ such that $F(X) \prec F(X^*)$. The set of all the Pareto optimal solutions is called Pareto set (PS) denoted as:

$$\text{PS} = \{X^* \in \Omega | \nexists X \in \Omega, F(X) \prec F(X^*).\} \tag{2}$$

The mapping of the PS in the objective space is called the Pareto front (PF)

$$\text{PF} = \{F(X^*) | X^* \in \text{PS}\}. \tag{3}$$

One of the main factors that can influence the algorithm's ability to find the optimal solution(s) is search directions over landscape. When the main goal of an algorithm is finding the optimal solutions for multiple objective functions, considering a search direction and its opposite simultaneously can be beneficial to enhance the performance of the algorithm. OBL is a novel research field which has already attracted a recognizable interest in the past decade.

2) Opposition-based learning (OBL): The concept of many entities or situations in the real world is described by using the opposition concept. OBL can potentially be used with soft computing techniques to solve engineering and science problems [4]. The main idea behind OBL is to consider an estimate and its corresponding opposite estimate synchronously in order to obtain a better approximation for the optimal solution. The first effort of utilizing the OBL concept in an optimization method was proposed by Rahnamayan et al. in 2006 [4]. Many soft computing algorithms have been enhanced by utilizing the concept of OBL such as, Reinforcement Learning (RL), Artificial Neural Networks (ANN), Fuzzy Systems. When the main goal of an algorithm is finding the optimal solution for an objective function, considering an estimate and its opposite simultaneously can be beneficial to enhance the performance of the algorithm. The computational opposition concept [3] was inspired from the opposition concept in the real world and the opposite action as follow:

Definition 1. *(Opposite Action) Suppose that an action changes the coordinates of a given state in a certain direction by the value* Δ $(s' = s \pm \Delta)$. *The opposite action* s *is able to change the coordinates to opposite direction by the same value* $(s' = s \mp \Delta)$.

Many optimization methods have been developed by utilizing the OBL concept in a) the initialization (population-level), b) during evolution, or c) designing mutation and crossover steps (operation-level) [7]. Some research works applied OBL concept in the MOO methods to enhance their performance. OBL was used in the population initialization and during the evolution process of multi-objective DE to enhance its performance in [4] and particle swarm optimization in [8] and in solving high-dimensional optimization problems.

3) MAdam: In [2] MAdam is a population-based scheme that aims to find the Pareto front via the hybridization of two optimization procedures. In this study, hybrid OMAdam is developed to enhance exploration and exploitation of Madam optimizer, robustness to local optima, handling non-convex and discontinuous landscapes.

III. Proposed Algorithm

Although the MAdam framework offers a Pareto front, this scheme suffers from poor convergence when capturing the non-convex shape of the Pareto front and relies on the injection of extreme points into the initial population. In contrast, we introduce OMAdam, designed to conduct a global search in eight directions across the landscape. This approach is aimed at escaping local optima, discovering the global optimum, and ensuring independence from the initial starting point. The choice of direction in optimization depends on both the specific problem being addressed and the current stage of the optimization process. When individuals in the population move exclusively in a single direction, as observed in MAdam, they primarily explore different problem spaces rather than different stages of optimization. On the other hand, exploring along multiple directions introduces a competitive environment where different directions contend with each other. Consequently, the optimization process follows the directions that outperform others, ultimately leading towards optimal solutions. For instance, imagine a vehicle equipped with several cameras, which grants it the flexibility to choose the most convenient path to traverse. Similarly, in the realm of optimization, the process selects the most promising directions from a pool of several possibilities, taking into account the specific problem at hand and the current stage of the optimization process. When it comes to gradient-based MOO, searching over various landscape directions can provide several benefits. First of all, in gradient-based optimization, there is a risk of getting trapped in local optima, where the objective function gradients become flat or misleading. Searching along multiple directions mitigates this issue by exploring different regions of the landscape simultaneously. It increases the chances of escaping local optima and finding globally optimal or near-optimal solutions. In addition, searching along multiple directions allows for a broader exploration of the solution space in MOO problem. It helps in discovering a diverse set of Pareto-optimal solutions spread across different regions of the landscape. By exploring multiple directions, the optimizer can uncover a wider range of trade-offs between conflicting objectives. Also, converging to the Pareto front can be challenging when using a single search direction. By searching along multiple directions simultaneously, the optimization algorithm can converge faster to a well-distributed set of Pareto-optimal solutions. This results in a more comprehensive coverage of the trade-off surface and a better representation of the solution space. Furthermore, MOO problems often involve non-convex or discontinuous landscapes with multiple local optima. By searching along multiple directions, the optimizer can effectively navigate such complex landscapes. It enables the algorithm to overcome irregularities, discontinuities, and other challenging characteristics of the objective function space. OMAdam is a hybridization of opposition-based scheme with the novel multi-objective gradient search (MAdam). The proposed algorithm incorporates multi-opposite directions into the weights of the first and the second moments of the gradients of the smooth functions in MOO problem. In this scheme the moving averages across landscape are estimates of mean gradient (first moment) and variance (second moment) with different weights of opposite directions. MAdam algorithm inherits moving average formula from Adam such that exponentially decaying mean gradient and variance gradient for each input updates through:

$$m(t) = \beta_1 * m(t-1) + (1-\beta_1) * g_j(t,i) \tag{4}$$

$$v(t) = \beta_2 * v(t-1) + (1-\beta_2) * g_j(t,i)^2, \tag{5}$$

where algorithm is executed iteratively over t as time and i for each component of gradient vector, and β_1 and β_2 are the hyper-parameters that are decayed on the schedule over the iterations of the algorithm and for number of objectives $j = 1, \cdots, M$ and a vector $Z = (z_1, \cdots, z_n)$ we have:

$$g_j = \nabla f_j = (\frac{\partial f_j}{\partial z_1}, \cdots, \frac{\partial f_j}{\partial z_n}). \tag{6}$$

In OMAdam to search over various landscape directions, moving averages are estimates of the opposite directions of the first (the mean) and the second raw moment (the variance) of the gradients. For this, OMAdam permutes the weights of these gradient vectors to have $4 * M$ (M is the number of objectives) different directions for calculating step size for each input parameter that is being optimized. Importantly, to exploit the search space, each step size is automatically adopted through search process based on $4 * M$ opposite directions and partial derivative of each objective respect to each dimension (Algorithm 1). To balance the convergence and the diversity, elitism techniques as found in NDS and crowding distance are used to partition a set of candidate solutions into different fronts and maintain diversity, respectively. OMAdam's population-based technique amplifies multiple opposite directions that help in exploration during the evolutionary process. OMAdam algorithm consists of five major components: random initialization, the core of OMAdam optimizer, finding non-dominated sorting, crowding distance calculation, and finding an elitist population for the next generation.

A. *OMAdam Initialization*

Compared to MAdam that is dependent to Adam performance to inject the extreme point, OMAdam is not sensitive to population initialization and the population is randomly initialized. Because by traversing along $4 * M$ distinct directions, the optimization process can effectively identify, at least a favorable point that aligns with the trajectory leading to the Pareto front. In this study, we consider MOO benchmarks with two-objectives and three-objectives. This results to have eight and twelve directions, respectively. For OMAdam scalability, We also investigate two conflicting parts of an aggregated objective in deep learning as two distinct, conflicting losses within the OMAdam optimization framework.

B. *OMAdam*

The scope of designing OMAdam as generalization of MAdam scheme is to develop the local search. OMAdam utilizes OBL concept during evolutionary process and searches multiple opposite directions simultaneously to find promising regions. Thus, we have assumptions about exploration of multiple directions.

Assumption 1: For gradient-base optimization moving along multiple opposite directions provides a broader view of the solution space, helping to identify promising regions more efficiently. This makes the optimization process less dependent on the first raw population and leads to faster convergence to the Pareto-optimal front compared to single-direction optimization strategies.

For the gradient of each objective, the weights of the first moment (m) and second moment (v) permutes to make opposite directions. For each objective function we have following updates for mean and variance momentum:

$$\text{D1:} \begin{cases} m(t) = \beta_1 * m(t-1) + (1-\beta_1) * g_j(t,i) \\ v(t) = \beta_2 * v(t-1) + (1-\beta_2) * g_j(t,i)^2, \end{cases} \tag{7}$$

$$\text{D2:} \begin{cases} m(t) = (1-\beta_1) * m(t-1) + \beta_1 * g_j(t,i) \\ v(t) = (1-\beta_2) * v(t-1) + \beta_2 * g_j(t,i)^2, \end{cases} \tag{8}$$

$$\text{D3:} \begin{cases} m(t) = \beta_1 * m(t-1) + (1-\beta_1) * g_j(t,i) \\ v(t) = (1-\beta_2) * v(t-1) + \beta_2 * g_j(t,i)^2, \end{cases} \tag{9}$$

$$\text{D4:} \begin{cases} m(t) = (1-\beta_1) * m(t-1) + \beta_1 * g_j(t,i) \\ v(t) = \beta_2 * v(t-1) + (1-\beta_2) * g_j(t,i)^2, \end{cases} \tag{10}$$

where $g_j(t,i)$ is selected according to each component of $\nabla f_j, \quad j = 1, \cdots, M$ (equation (6)) and $Di, \quad i = 1, \cdots, 4$ represent the directions. The schematics illustration of these directions is shown in Fig. 1. In OMAdam moving averages of each member of the population are estimates of the four means and four uncentered variances momentum of each gradient. The data frame of OMAdam stores information of each population along their moving averages. OMAdam exploits the landscape with Adam update formula and their opposite directions that comes from reordering the weights of the first and the second moments for two gradients (Algorithm 1). Our assumption is that:

Assumption 2: Searching along multiple opposite directions provide the wider spectrum of Pareto front. The broader range of solutions offered by a wider Pareto front indicates the presence of solutions that are robust and resilient across multiple objectives. While in MAdam, single direction guides the optimization process that results to narrow Pareto front trajectory. Disconnected landscapes pose challenges for MOO algorithms. Exploring multiple directions can assist in handling such complexities by capturing the local structure and non-linear relationships among the objectives. It helps avoid premature convergence to sub-optimal regions and provides a more accurate representation of the problem's characteristics.

Assumption 3: In contrast to the single solution provided by Adam, both MAdam and OMAdam generate a collection of solutions that capture the trade-off between conflicting objectives. Although the aggregation method provides a practical way to handle MOO problems using existing single-objective optimization techniques, utilizing the scalarization technique for a weighted sum of multiple objective functions does not allow controlling values of individual objective functions during the optimization process. This method typically produces a single solution that represents a specific point in the objective space. The lack of diversity can restrict the decision-maker's choices and hinder a comprehensive analysis of the problem's landscape. The schematics illustration of OMAdam according to the aforementioned directions is provided in Fig. 1. This figure demonstrates the multi-directional movement of a single candidate solution. The exploration process yields $4 * M$ updated points, each corresponding to a specific direction. Subsequently, the objective values of these updated points are calculated to be evaluated in NDS algorithm and finding elitist population for the next generation. Considering the exponential growth potential inherent in the proposed scheme during the optimization process, we adopt a fixed number of individuals in each generation to maintain control and stability. The framework of OMAdam is similar to its parent algorithm MAdam [2]. The first population is randomly generated in [0,1]. Next, the core of OMAdam is a data frame that is instantiated as the matrices with number of population and keeps the information of each direction of all individuals that consists of mean and variance moments, updated variables for each dimension, and associated objective values. This data set updates during optimization toward promising regions.

Algorithm 1 presents the scheme of the exploitation. The first step is to calculate the partial derivative of each dimension according to equation 6. Next, by reordering the weights of mean and variance of gradients, the multiple opposite direction are built (formulas (7),(8),(9), and (10)). OMAdam calculation upgrades the information data set of each population based on the two gradients and eight opposite directions. Lines 2 and 3 calculate the first and the second moments for each direction, respectively. Lines 4 and 5 indicate the bias correction of the first and the second moments. Then, in line 6 the variable values are updated. This is then repeated for each dimension. In line 7 and 8, the functions of an updated variable are evaluated. When the calculations for one direction are completed for all dimensions, its associated columns are attached to the previous data columns (Line 9).

Experimental Results and Analysis

C. Comparative Analysis: OMAdam vs. MAdam

In this section, we investigate the effect of how the proposed mechanism is able to lead to an improvement in the MOO performance of Madam on benchmarks and in a fully connected network. Also, the superiority of OMAdam over the Adam optimizer is investigated in terms of finding a set of trade-off solutions, while both demonstrate equivalent performance in classification tasks as measured by the F1-score. This comparison is conducted by considering three key components of the algorithm: the initialization of the first population, the shape of Pareto front on benchmarks and network and finally finding a set of solutions in early stages of optimization. In this analysis, Pareto front of OMAdam is compared with the Pareto front of single direction Madam on different representative kinds of bi-objective and three objective test functions (Table II). Then for examine the results scalability, we compare the Pareto front of OMAdam optimizer with the Pareto front of Madam optimizer on a network (Table I) . This comparison

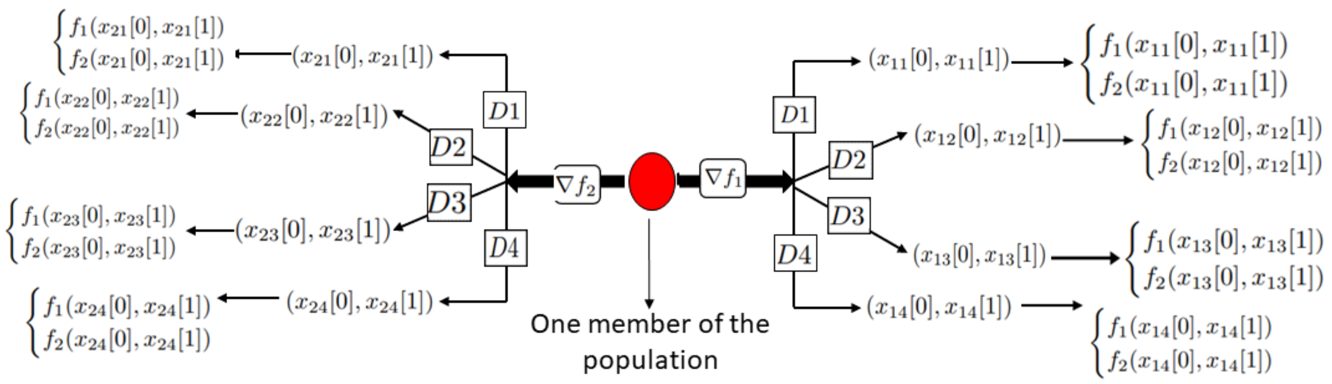

$f_1(x_{21}[0], x_{21}[1])$, $f_2(x_{21}[0], x_{21}[1])$
$(x_{21}[0], x_{21}[1])$
$D1$
$D2$
$D3$
$D4$
∇f_2
∇f_1
One member of the population
$(x_{11}[0], x_{11}[1])$
$f_1(x_{11}[0], x_{11}[1])$, $f_2(x_{11}[0], x_{11}[1])$

Fig. 1. Process flow of OMAdam

is based on the two conflicting objectives of Mean Absolute Error (MAE) and Binary Cross Entropy (BCE), as shown in equation 11 and 12, respectively, using the MNIST dataset for both training and testing. Experiments demonstrate that OMAdam provides a wider range of solutions compared to the MAdam and Adam optimizer on MOO benchmarks and better performance in binary classification task.

D. Optimization Configuration

In the repeatable optimization process, we configure the baseline hyperparameters based on the values specified in [1]. In this context, the hyperparameters for the OMAdam algorithm are established as follows: α is set to 0.02, β_1 to 0.8, β_2 to 0.999, and ϵ to 10^{-8}. We use MAE and BCE as differentiable objectives to measure the accuracy of a predictive model. In MAdam and OMAdam schemes two distinct and conflicting terms of scalarized MAE and BCE represent the multi-objective losses to compete and converge evolutionary to a set optimal trade-off weights to train a fully connected network.

$$\begin{cases} \text{objective 1: MAE}_1 & = \text{y\_true} \times (1 - \text{y\_pred}), \\ \text{objective 2: MAE}_2 & = (1 - \text{y\_true}) \times \text{y\_pred}, \end{cases} \tag{11}$$

$$\begin{cases} \text{objective 1: BCE}_1 & = -\text{y\_true} \times \log(\text{y\_pred}), \\ \text{objective 2: BCE}_2 & = -(1 - \text{y\_true}) \times \log(1 - \text{y\_pred}), \end{cases} \tag{12}$$

where y_true is the actual values and y_pred is the predicted value of images observation. In a series of experiments, we assess the comparative quality of the Pareto fronts generated by OMAdam against the prior work MAdam on binary MNIST dataset [2]. We use 14.45% of the training data as a validation split, which includes 12,089 images containing the digits 2 and 3 for training, and 2,042 images designated for validation testing. This dataset as an adaptation of the original MNIST dataset, is particularly interesting due to a unique feature: its digits are slightly offset. In this version, each digit image is manipulated to have two distinct variations – one with a slight offset towards the bottom right (BR) and the other towards the top left (TL). This variation introduces a subtle yet significant challenge in pattern recognition and image classification tasks. Table I illustrates the architecture of a used neural network designed that involves flattened input data of size 784, associated with image data of binary MNIST dataset of handwritten digits. The network comprises a sequence of layers including a flatten layer that maintains the input shape, followed by two dense layers with the first having 128 neurons and the second serving as the output layer with a single neuron.

TABLE I
MODEL SUMMARY

Layer (type)	**Output Shape**	**Param #**
flatten_input (InputLayer)	(None, 784)	0
flatten (Flatten)	(None, 784)	0
dense (Dense)	(None, 128)	100,352
dense_1 (Dense)	(None, 1)	129

This network incorporates ReLU and sigmoid activation functions to provide probabilities indicating the likelihood of belonging to one of the classes in a binary classification task. To ensure a fair comparison between the OMAdam and MAdam schemes in complex systems analysis, a population size of 50 is employed, and in epistemology of optimization a population size is 300 with random initialization of weights for OMAdam within the range [0,1]. Experiments include two-objectives and three-objectives schemes result to have eight and twelve directions, respectively. In the experimental part of this work, we use the Hyper Volume (HV) metric to assess how well an algorithm performs in finding a diverse and extensive set of optimal solutions. As the HV value increases, it signifies an expansion of the non-dominated front, indicating the presence of a more diverse and efficient set of solutions that exhibit better trade-offs between multiple conflicting objectives.

E. Ablation study: OMAdam vs. MAdam on Benchmarks

Fig. 2 illustrates in the objective space, the distribution of the final found solutions obtained by OMAdam and Madam

Algorithm 1 Pseudo-code for updating OMAdam data set for M=2 objectives

```
Input:
  • Dimension $D$, objective functions $f_j$, $j = 1, \cdots, M (M = 2)$, partial derivatives $\nabla_i f_j(x)$, $i = 1, \cdots, D$,
  • $\omega_1 = [\beta_1, (1-\beta_1), \beta_2, (1-\beta_2)]$
  • $\omega_2 = [(1-\beta_1), \beta_1, (1-\beta_2), \beta_2]$
  • $\omega_3 = [\beta_1, (1-\beta_1), (1-\beta_2), \beta_2]$
  • $\omega_4 = [(1-\beta_1), \beta_1, \beta_2, (1-\beta_2)]$
  • $\Omega = \{\omega_1, \omega_2, \omega_3, \omega_4\}$, Population Pop$_{set}$, Time $t$, Step size $\alpha$, Exponential decay rates for the moment estimate $\beta_1, \beta_2 \in [0, 1)$, $\epsilon = 10^{-8}$, bounds: define range for input population
Output: updated OMAdam data set (OMD)
for $x \in Pop_{set}$ do
    1: $g_t = \nabla_{x_k} f_j(x_{k-1})$    for  $k = 1, \cdots, D$
    for $\omega_p \in \Omega$ do
        $j = 0$
        for $i \leftarrow 0$ to $D - 1$ do
            2: $OMD[x, i+j]_t \leftarrow \omega_p[1] * OMD[x, i+j]_{t-1} + \omega_p[2] * g_t[i]$
            3: $OMD[x, i+1+j]_t \leftarrow \omega_p[3] * OMD[x, i+1+j]_{t-1} + \omega_p[4] * g_t[i]^2$
            4: $\hat{m} \leftarrow \frac{OMD[x,i+j]_t}{(1-\beta_1^{t+1})}$
            5: $\hat{v} \leftarrow \frac{OMD[x,i+1+j]_t}{(1-\beta_2^{t+1})}$
            6: $OMD[x, i+2+j]_t \leftarrow OMD[x, i+2+j]_{t-1} - \alpha . \frac{\hat{m}}{\sqrt{\hat{v}}+\epsilon}$
            $j = j + 2$
            7: $OMD[x, 6] \leftarrow f_1(OMD[x, 0], \cdots, OMD[x, D])$ (Update $f_1$-values)
            8: $OMD[x, 7] \leftarrow f_2(OMD[x, 0], \cdots, OMD[x, D])$ (Update $f_2$-values)
        9: OMAdam=OMAdam $\cup$ columns associated with $\omega_p$
    updated OMAdam
```

Algorithm 2 OMAdam Framework

```
Input: Maximum number of iteration max$_{iter}$
Output: Elitist individuals $S$
  • 1: Generate random Pop$_{set}$ in $[0, 1]$ with size $N$pop
  • 2: Define OMAdam data-set (OMD) (Algorithm 1) and initialize with Pop$_{set}$
for $t \leftarrow 1$ to $max_{iter}$ do
    for $i \leftarrow 0$ to $D$ do
        for $d \leftarrow 1$ to $D$ do
        : $S = \emptyset$
        3: /*Exploitation*/ : update OMD (Algorithm 1)
    4:/*Exploration*/ excluding objective values information from update OMD, NDS to find front, crowding distance
    9: Finding corresponding population update $S$
```

on ZDT problems (Table. II). It is visually evident that in OMAdam the Pareto front of each test function has a wider spectrum with respect to its parent algorithm Madam. By having a wider spectrum of solutions, the risk of relying on a single, narrow solution is reduced. The narrow Pareto front of MAdam may be susceptible to being dominated by a single extreme point, which could be sensitive to uncertainties or variations in the problem's parameters. In pursuit of capturing the Pareto front for the parent algorithm Madam, it is imperative to incorporate extreme points generated by the Adam optimizer into the initial raw population. Empirical experimentation depicts the limitations of this approach in finding the Pareto front with random initialization for two and three objective benchmarks. Nonetheless, the execution of the OMAdam algorithm, with or without the initialization step of MAdam, consistently demonstrates the robustness of the proposed OMAdam method to initialization and the algorithm exhibits independence from specific extreme points, and the initial population is randomly generated within the range of (0,1). In Fig. 2 we see that MAdam catches a narrow spectrum of Pareto front which aligns with the subset of solutions obtained by the OMAdam algorithm. However, due to the comparatively lower rate of convergence to dis-connected Pareto front of ZDT3, higher rank of fronts are included for this benchmark. This shows MAdam struggling to capture the disconnected Pareto front. Because of this, solutions from the prior Pareto fronts are added to the last state of the Pareto front, which results in a comparatively denser population. But comparing the HV values for both algorithms (as shown in Table III) shows ODMadam performs better than Madam on four test problems. It becomes apparent that while both algorithms yield the Pareto optimal set, OMAdam exhibits a superior ability to explore and produce a larger number of solutions on the Pareto front compared to its parent algorithm. To scale up the performance of OMAdam in large number of objectives, we use DTLZ2 as a generic sphere problem (Table II). The Pareto-optimal solutions must lie inside the first octanes of the unit sphere in a three-objective plot with f_3 as one of the axes. Fig. 3 a, and Fig. 3 b, show the distribution of the obtained solution on DTLZ2 generated by OMAdam and MAdam schemes, respectively. As it is shown in OMAdam the solutions are well distributed than MAdam. To test the performance of OMAdam on disconnected and non-convex benchmarks, we use the test problem that belongs to Bottom-up approach [5]. In the bottom-up approach the construction procedure begins by assuming a mathematical formulation of the Pareto-optimal front. Such function is embedded in the overall test problem design so that two different types of difficulties of converging to the Pareto-optimal front and maintaining a diverse set of solutions can be introduced (Table II, 6,7,8). One of such problems is DTLZ7 (Table II, function number 6) that has a disconnected set of Pareto-optimal regions. This problem tests an algorithm's ability to

TABLE II
MOO BENCHMARKS DEFINITIONS

MOO benchmarks			
	Name	Problem	Domain
1	ZDT1	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{D-1}\Sigma_{i=2}^{D} x_i$ $h(f_1, g) = 1 - \sqrt{f1/g}$	$[0, 1]$
2	ZDT2	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{D-1}\Sigma_{i=2}^{D} x_i$ $h(f_1, g) = 1 - (f1/g)^2$	$[0, 1]$
3	ZDT3	$f_1(x) = x_1$ $g(x) = 1 + \frac{9}{D-1}\Sigma_{i=2}^{D} x_i$ $h(f_1, g) = 1 - \sqrt{f_1/g} - (f_1/g)\sin(10\pi f_1)$	$[0, 1]$
4	ZDT4	$f_1(x) = x_1$ $g(x) = 1 + 10(D-1) + \Sigma_{i=2}^{D}(x_i^2 - 10\cos(4\pi x_i))$ $h(f_1, g) = 1 - \sqrt{f_1/g}$	$x_1 \in [0, 1]$ $x_i \in [-10, 10]$, for $i > 1$
5	DTLZ2	$f_1(x) = (1+\mathrm{g}(\mathrm{x}_M))cos(x_1\pi/2)cos(x_2\pi/2)$ $\mathrm{f}_2(x) = (1+\mathrm{g}(\mathrm{x}_M))cos(x_1\pi/2)sin(x_2\pi/2)$ $\mathrm{f}_3(x) = (1+\mathrm{g}(\mathrm{x}_M))sin(x_1\pi/2)$ $g(x_M) = \sum_{x_i}(x_i - 0.5)^2$	$x_i \in [0, 1]$ for $i = 1, \cdots, n$
6	DTLZ7	$f_1(x) = \mathrm{x}_1$ $\mathrm{f}_2(x) = \mathrm{x}_2$ $\mathrm{f}_3(x) = (1+\mathrm{g}(\mathrm{x}_M))h(f_1, f_2, g))$ $g(x_M) = 1 + \frac{9}{\lvert x_M \rvert}\sum_{x_i} x_i$ $h(f_1, f_2, g) = 3 - \sum_{i=1}^{2}[\frac{f_i}{1+g}(1 + sin(\pi f_i))]$	$x_i \in [0, 1]$, for $i = 1, \cdots, n$
7	non-convex Pareto-optimal front	$f_1(x) = x_1$ $f_2(x) = \mathrm{x}_2$ $\mathrm{f}_3(x) = (\mathrm{g}(\mathrm{x}_M))h(f_1(x_1), f_2(x_2), g(x_M)))$ $g(x_M) = 1 + \frac{9}{\lvert x_M \rvert}\sum_{x_i} x_i$ $\mathrm{h}(\mathrm{f}_1, f_2) = 1 - (\frac{\sum_{i=1}^{2} f_i}{\beta})^{\alpha}$	$\alpha = 2$, $\beta = 0.5$, $M = 3$
8	disconnected Pareto-optimal surfaces	$f_1(x) = x_1$ $f_2(x) = \mathrm{x}_2$ $\mathrm{f}_3(x) = (\mathrm{g}(\mathrm{x}_M))h(f_1(x_1), f_2(x_2), g(x_M)))$ $\mathrm{h}(\mathrm{f}_1, f_2) = 2M - \sum_{i=1}^{M-1}(2f_i + sin(3\pi f_i))$	$M = 3$

maintain subpopulation in different Pareto-optimal regions. Fig. 3 c show OMAdam and MAdam after 100 generations. It is clear that OMAdam can capture the fronts by searching along twelve directions. While PF coverage of MAdam is very poor. Fig. 3 e, and Fig. 3 f shows a non-convex Pareto-optimal front of three objective benchmark that belongs to bottom-up approach (Table II, function number 7). Despite the fact that OMAdam demonstrates a superior coverage of the Pareto front in this benchmark, the performance of this algorithm remains competitive with its parent algorithm. Another generic benchmark that is affiliated to bottom-up approach is the function with disjoint set of Pareto-optimal front (Table II, function number 8). Fig. 3 d shows the out-performance of OMAdam respect to its Parent algorithm.

F. Ablation study: OMAdam vs. MAdam on Training ANNS

Fig. 4 depicts the comparison of the MAdam and OMAdam schemes with multi objective MAE and BCE losses on both training and test datasets, evaluating the quality of the Pareto front using the hyper-volume metric. In each generation, both optimizers presents a set of potential solutions. Each point on the Pareto front corresponds to a unique set of model parameters, or weights configuration within the model. These weights are crucial because they adjust how the model behaves and makes predictions. The visual observation reveals that MAdam and OMAdam exhibits in forming the final state of the Pareto front with train and test datasets. For test data, objectives are selected through elitist filter (NDS). OMAdam schemes captures Adam single solution as one point on its Pareto front spectrum, demonstrating the potential for offering solutions to conflicting losses, unlike a single solution Adam which might not precisely reflect the actual preferences or trade-offs. This affirms that numerous solutions in the optimization process remain unseen when utilizing Adam's single solution approach. MAdam and OMAdam Pareto front trajectory depicts equal hyper-volume while OMAdam has well-distributed spectrum of validation dataset. Also, OMAdam captures the trajectory of the Pareto front at earlier stages of optimization respect to its parent algorithm Madam on train and test data. It becomes apparent from Fig. 4 that while both algorithms yield the Pareto optimal set, OMAdam exhibits a superior ability to explore and produce a larger number of solutions on the Pareto front compared to its parent algorithm. Table V presents the range of extreme losses, from the initial optimization stages to the model optimized against two opposing objectives, indicated by MAE1 and MAE2. This is shown for both the MAdam and OMAdam optimization methods. MAE_1 and MAE_2 denote the range of the assessed losses for the initial and optimized weights respectively, corresponding to each optimizer. In the initial stages of optimization using MAdam, the focus is predominantly on reducing MAE2, rather than MAE1. This approach gradually shifts the model's performance, moving from excelling primarily in one loss to enhancing performance in the other (Fig. 4). As this process evolves, we see a contrasting point at the other end of the spectrum, where the emphasis significantly switches to improving MAE2, the second loss. The trajectory from one end of the Pareto front to the other demonstrates a shift in priority from one objective to the other, making it evident that decreasing in one loss invariably leads to increasing in the other. This trade-off causes the practitioner to select a post hoc solution based on the achieved trade-offs among loss functions as multiple objectives. The range of possible solutions with OMAdam extends further from the lower bound. As the framework focuses on minimization-based optimization, OMAdam demonstrates superior performance compared to MAdam. Similarly, this optimizer seeks to find the most balanced set of weights that will minimize the competing objectives, which in the context of binary classification, could be related to minimizing false positives for one class while

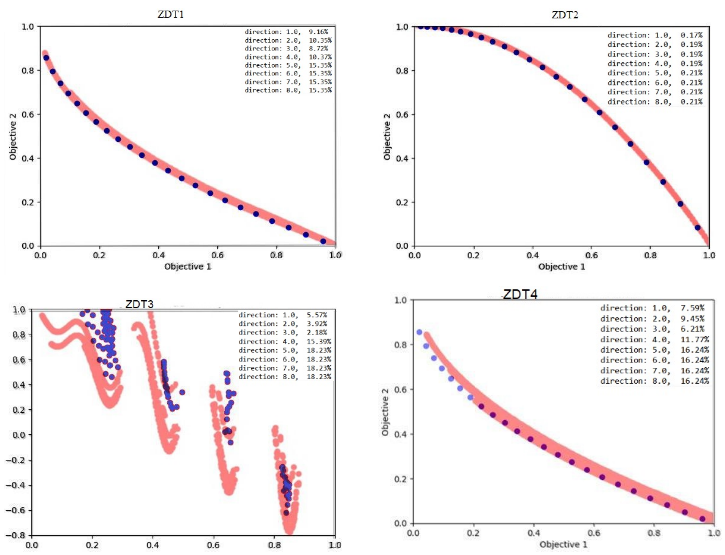


Fig. 2. Comparison of Pareto fronts of ZDT1, ZDT2, ZDT3, and ZDT4, resulted from OMAdam and MAdam. The blue traverse represent the PF of MAdam and the red spectrum shows PF of OMAdam. The contribution of eight directions is reported to evaluate the success of each path over all iterations.

also minimizing false negatives for the other. In evolutionary algorithms, the F1-score is calculated after the optimization process, alongside the Pareto front, to assess the balance between precision and recall for each model. In MAdam and OMAdam schemes, each point on the Pareto front is indicative of a unique combination of parameters for the model, which are utilized to make predictions on a set of validation data and subsequently evaluate the F1-score. The highest F1-score among the solutions on the last Pareto front is reported in Table IV. This table showcases the maximum F1-score derived from the array of F1 scores corresponding to the individuals located on the Pareto front for MAdam and OMAdam. Both optimizers capture Adam's F1-score, despite Adam's method of combining two losses of MAE and BCE into a single objective and provide one solution, which results in the loss of granularity and information about each loss individually. This approach can be problematic when analyzing and interpreting the performance of a classifier in different contexts. In contrast, the OMAdam scheme provides the F1-score for each solution on the Pareto front, offering insights into confusion matrix entries and facilitating a reasonable trade-off between them. This enables decision-makers to explore various options and consider a range of trade-offs between objectives in the classification task. OMAdam achieves high-performance model parameters by the second generation. This confirms the computational efficiency of OMAdam compared to MAdam and Adam. Such high-performance models are found after 30 iterations through MAdam and Adam optimizers. This result aligns with OMAdam's high and rapid exploratory capacity to capture the spectrum of the Pareto front in benchmarks. Additionally, OMAdam outperforms MAdam due to variations in the confusion matrix, where false positives and false negatives compete along the trajectory of the Pareto front. This competition results in more optimal performance in classification tasks.

TABLE III
COMPARISON OF OMADAM AND MADAM ON HV OF BENCHMARKS

Functions	HV	
	MAdam	OMAdam
ZDT1	0.43	**0.66**
ZDT2	0.39	**0.56**
ZDT3	0.38	**0.69**
ZDT4	0.43	**0.66**

TABLE IV
COMPARISON F1-SCORE OF THREE OPTIMIZERS OF MEAN-ABSOLUTE ERROR (MAE) AND BINARY CROSS ENTROPY (BCE)

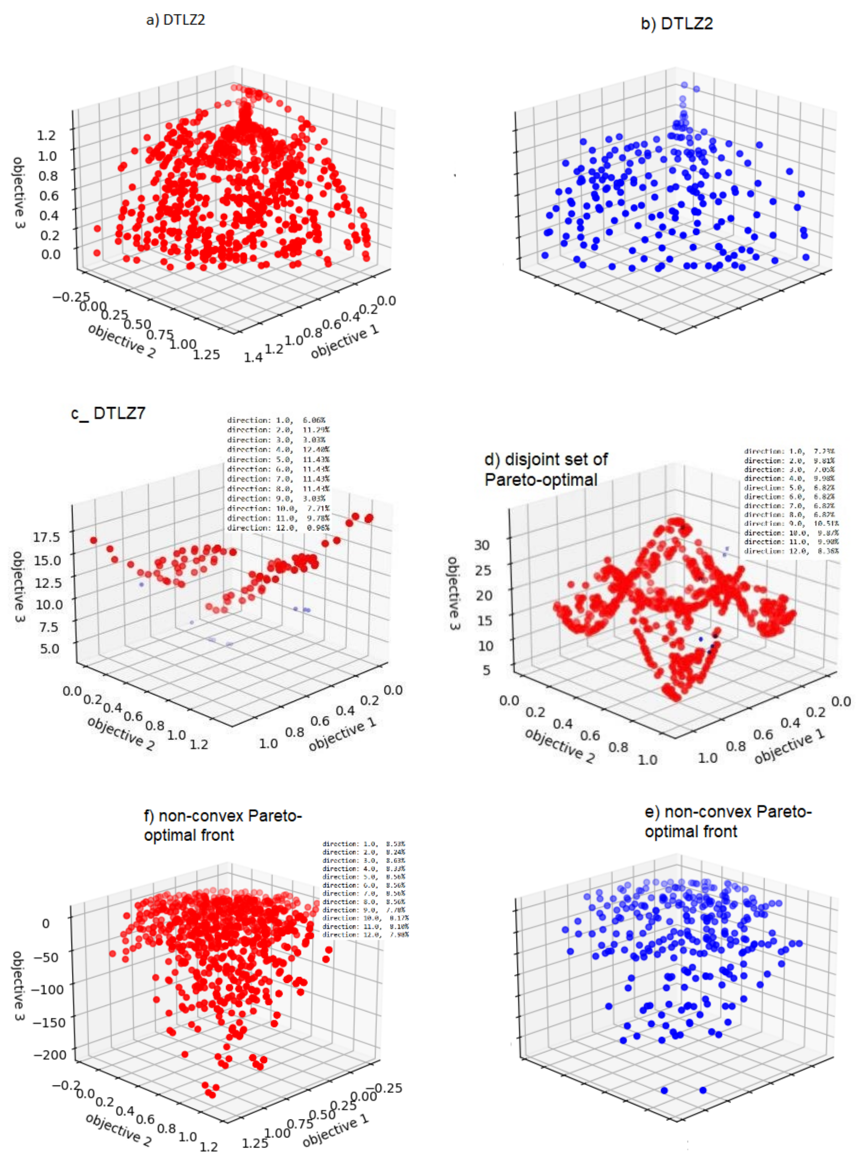
Optimizer	**Loss functions**	**F1-score**
OMAdam	multi-objective MAE	0.97
OMAdam	multi-objective BCE	0.97
MAdam	multi-objective MAE	0.97
MAdam	multi-objective BCE	0.96
Adam	MAE	0.97
Adam	BCE	0.97

TABLE V
REPORTING LOSSES MAE1 AND MAE2

Optimizer	**Initial Weight Loss**	**Optimized Weight Loss**
MAdam	MAE1: 0.03 MAE2: 0.48	MAE1: 0.48 MAE2: 0.01
OMAdam	MAE1: 8.34×10^{-7} MAE2: 0.32	MAE1: 0.2 MAE2: 7.14×10^{-7}

IV. CONCLUSION

In this paper, we introduce OMAdam, a gradient-based, multi-objective optimizer capable of producing a qualitative



```
direction: 1.0,  6.06%
direction: 2.0,  11.29%
direction: 3.0,  3.03%
direction: 4.0,  12.40%
direction: 5.0,  11.43%
direction: 6.0,  11.43%
direction: 7.0,  11.43%
direction: 8.0,  11.43%
direction: 9.0,  3.03%
direction: 10.0,  7.71%
direction: 11.0,  9.78%
direction: 12.0,  0.96%
```

```
direction: 1.0,  7.73%
direction: 2.0,  9.81%
direction: 3.0,  7.05%
direction: 4.0,  9.98%
direction: 5.0,  6.82%
direction: 6.0,  6.82%
direction: 7.0,  6.82%
direction: 8.0,  6.82%
direction: 9.0,  10.51%
direction: 10.0,  9.87%
direction: 11.0,  9.98%
direction: 12.0,  8.36%
```

```
direction: 1.0,  8.53%
direction: 2.0,  8.24%
direction: 3.0,  8.63%
direction: 4.0,  8.33%
direction: 5.0,  8.56%
direction: 6.0,  8.56%
direction: 7.0,  8.56%
direction: 8.0,  8.56%
direction: 9.0,  7.78%
direction: 10.0,  8.17%
direction: 11.0,  8.10%
direction: 12.0,  7.98%
```

Fig. 3. Comparison of Pareto fronts of DTLZ2, DTLZ7, non-convex Pareto front and disconnected Pareto front. The blue traverse represent the PF of MAdam and the red spectrum shows PF of OMAdam

Pareto front through multiple directions to search globally across landscape. The trajectory from one end of the Pareto front to the other highlights a transition in performance priorities between objectives, offering practitioners a context for evaluating the trade-offs among competing objectives. Unlike the single-solution optimizer Adam with subjective weights that leads to limited insights into the satisfaction of each conflicting objective. Multi-directional exploration with OMAdam leads to independence from specific initializations, with respect to MAdam scheme and is more likely to discover good solutions regardless of the starting point. In addition, in OMAdam scheme the optimizer can uncover a wider range of trade-offs between conflicting objectives, while, Adam occupies a relatively smaller region within the OMAdam spectrum. This affirms Adam optimizer's incapacity to capture numerous solutions, resulting in suboptimal solutions or premature convergence. variation of F1-score values that changes along the Pareto front trajectory, enables a dynamic selection of F1-scores and confusion matrix entries, which is crucial in scenarios where the decision-maker needs to prioritize one feature over another. OMAdam matches Adam's F1-score in the early generations, a result of its high exploratory capacity that boosts its early-stage performance in classification tasks. This leads to a significant reduction in computational costs compared to the Adam optimization process. We emphasize that our gradient-based scheme utilizes smooth objectives for its analytical gradients. For non-continuous objectives, the use of numerical approximation of gradients or gradient-free approaches is necessary to guide the optimization procedure towards optimal Pareto front solutions across the landscape.

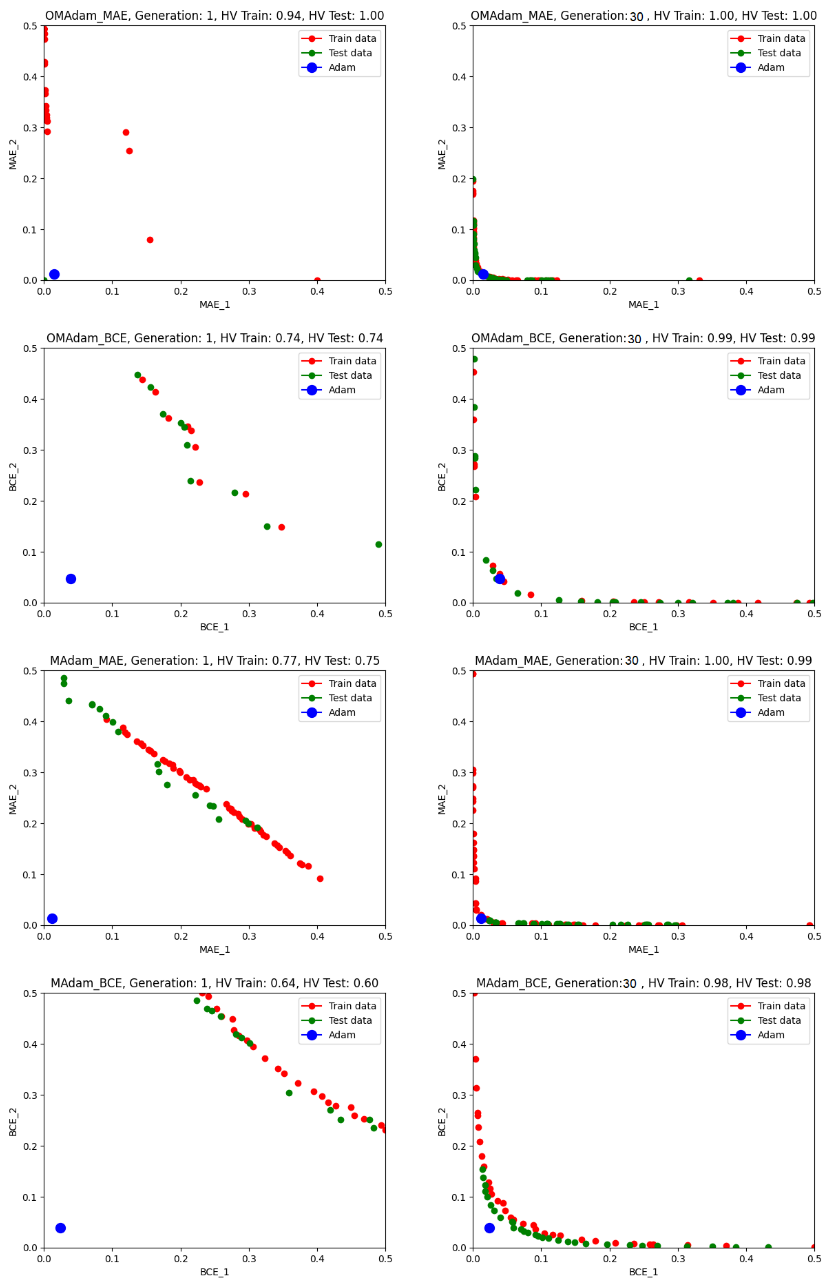


Fig. 4. Progression of the Pareto fronts over generations in the OMAdam and MAdam schemes on train and test data vs. Adam + HV. Comparison optimizer OMAdam, MAdam.

References

[1] Kingma, D. P., and Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

[2] Nikbakhtsarvestani, F., Ebrahimi, M., Rahnamayan, S. (2023, October). Multi-objective ADAM Optimizer (MAdam). In 2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3860-3867). IEEE.

[3] Tizhoosh, H. R. (2005, November). Opposition-based learning: a new scheme for machine intelligence. In International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06) (Vol. 1, pp. 695-701). IEEE.

[4] Rahnamayan, S., Tizhoosh, H. R., & Salama, M. M. (2008). Opposition-based differential evolution. IEEE Transactions on Evolutionary computation, 12(1), 64-79.

[5] Deb, K., Thiele, L., Laumanns, M., & Zitzler, E. (2005). Scalable test problems for evolutionary multiobjective optimization. In Evolutionary multiobjective optimization: theoretical advances and applications (pp. 105-145). London: Springer London.

[6] Deng, Li. "The mnist database of handwritten digit images for machine learning research [best of the web]." IEEE signal processing magazine 29.6 (2012): 141-142.

[7] Mahdavi, S., Rahnamayan, S., & Deb, K. (2018). Opposition based learning: A literature review. Swarm and evolutionary computation, 39, 1-23.

[8] Wang, H., Wu, Z., Rahnamayan, S., Liu, Y., & Ventresca, M. (2011). Enhancing particle swarm optimization using generalized opposition-based learning. Information sciences, 181(20), 4699-4714.