AI-POWERED SELF-DECISIVE ALGORITHM FOR TWO-STEP QUASI-NEWTON METHODS

Farah Jaffar[1], Nudrat Aamir[2], Syed Ibrahim[*3], Roohi Laila[4], Sidra Aman[5], Muhammad Touseef Irshad[6]

[1,2] Department of Basic Sciences and Humanities, CECOS University of IT and Emerging Sciences, Peshawar, Pakistan
[3] Riphah Institute of Informatics, Riphah international university Islamabad, Pakistan
*[*3] Hebei Foreign Studies University, Hebei, Shijiazhuang, Qiaoxi District, China*
[4] Department of Artificial Intelligence, Institute of Management Sciences, Peshawar, Pakistan
[5] School of Computer Science and IT, Institute of Management Sciences (IMSciences), Peshawar
[6] Department of Computer Science, National University of Modern Languages, Peshawar, Pakistan

[1]farah.jaffer@cecos.edu.pk, [2]nudrat@cecos.edu.pk, [*3]ibrahimmathsicp@gmail.com, [4]roohilaila98@yahoo.com, [5]sidraaman9@gmail.com, [6]m.touseef.irshad@gmail.com

DOI: https://doi.org/10.5281/zenodo.20152352

Keywords
Two-step quasi-Newton method, fixed-point approach, image value, skipping technique, search-direction modification.

Article History
Received: 14 March 2026
Accepted: 22 April 2026
Published: 13 May 2026

Copyright @Author
Corresponding Author: *
Syed Ibrahim

Abstract
The rapid evolution of machine learning has introduced a wide range of challenging and significant optimization problems. Various algorithms have been developed and trained to obtain optimal solutions for diverse problems in science, engineering, medicine, and related fields through machine learning techniques. In this context, fast gradient-driven optimization algorithms have become essential for computationally efficient model training. This study investigates an AI-powered self-decisive algorithm based on image-processing techniques for solving nonlinear unconstrained optimization problems. Different skipping strategies and search-direction modification techniques are incorporated within the framework of two-step quasi-Newton methods. Two test functions with different dimensions and initial points are examined using the fixed-point approach. The numerical simulations support the selection of the most suitable strategy for the proposed self-decisive algorithm in two-step quasi-Newton methods.

1. INTRODUCTION

Conventional optimization approaches often become insufficient when addressing the challenges associated with big data. In contrast, modern machine learning (ML) techniques provide powerful tools for processing large-scale datasets, identifying meaningful patterns, and generating accurate and actionable insights that support intelligent decision-making. ML techniques use large sets of input and output data to recognize patterns and effectively "learn" to make autonomous recommendations or decisions [11]. Optimization plays a fundamental role in the analysis of physical phenomena and decision sciences. Artificial intelligence may be viewed as a continuous pursuit of optimization, since the intelligence we aim to build ultimately depends on solving optimization problems. Whether in traditional machine learning, deep learning, or reinforcement learning, the underlying principles are consistently connected to optimization [10]. The main objective is either to minimize the required effort or to maximize the desired benefit, thereby enabling the solution of a given problem in an effective and optimal manner.

Among the wide range of optimization techniques, quasi-Newton methods have been established as efficient methods for problems of different dimensions [9, 3]. Further development in the form of multi-step quasi-Newton methods was introduced in [6, 5] and later enhanced in [2, 13] for solving unconstrained optimization problems using different techniques. Several strategies, including implicit updates [4, 8], minimum curvature approaches [7], and the exploitation of function values [15], have been investigated within the framework of multi-step quasi-Newton methods, producing promising results. Multi-step quasi-Newton methods approximate the current Hessian by using information from more than one previous step. Ford and Moghrabi [5] reported significant improvements of the multi-step quasi-Newton method over the classical quasi-Newton method. The method satisfies the secant equation for updating the Hessian approximation (B_{i+1}), defined as

$$B_{i+1}s_i = y_i. \tag{1}$$

where

$$s_i = x_{i+1} - x_i, \tag{2}$$

$$y_i = g(x_{i+1}) - g(x_i). \tag{3}$$

Two-step quasi-Newton methods satisfy the relation

$$B_{i+1}(s_i - \mu_i s_{i-1}) = y_i - \mu_i y_{i-1} \tag{4}$$

or

$$B_{i+1}r_i = w_i. \tag{5}$$

where μ_i is computed by δ as

$$\mu_i = \frac{\delta^2}{2\delta+1} \tag{6}$$

and

$$\delta = \frac{\tau_2-\tau_1}{\tau_1-\tau_0}. \tag{7}$$

The secant equation (5) is derived by constructing an interpolating polynomial $x(\tau)$, which interpolates in the variable space, while $g(x(\tau))$ interpolates in the gradient space as follows:

$$x(\tau_j) = x_{i+j-1}, \text{ for } j = 0,1,2, \tag{8}$$

$$g\left(x(\tau_j)\right) = g(x_{i+j-1}), \text{ for } j = 0,1,2. \tag{9}$$

By applying the chain rule to $g(x(\tau))$, the Hessian matrix $G(x_{i+1})$ satisfies the equation

$$G(x_{i+1})x'(\tau_2) = g'(x_{\tau_2}). \tag{10}$$

If the derivatives are given by

$$x'(\tau_2) = r_i \tag{11}$$

and

$$g'(x_{\tau_2}) = w_i, \tag{12}$$

then, after substituting into Eq. (9), Eq. (5) is obtained. A suitable updating formula for B_{i+1} is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula, defined as

$$B_{i+1} = B_i + \frac{B_i r_i r_i^T B_i}{r_i^T B_i r_i} + \frac{w_i w_i^T}{r_i^T w_i}. \tag{13}$$

Multi-step quasi-Newton methods were further improved in [1, 2] through the use of skipping techniques and modified search directions. A self-decisive algorithm [13] was successfully developed to reduce the computational burden during the execution process and was validated through numerical experiments on two test functions. In [13], two techniques, namely skipping and modified search direction in two-step quasi-Newton methods, were implemented using the accumulative approach. In this research, the fixed-point approach is applied to the two-step quasi-Newton method to identify the most suitable method for a particular type of function and to assist the algorithm in selecting the best-fit strategy. The remainder of the paper is organized as follows. First, a review of the self-decisive algorithm is presented. Next, the proposed fixed-point approach is discussed, and the parameters required for vector computation are calculated. This is followed by numerical results based on skipping and modified search-direction techniques. Finally, a numerical comparison is presented, leading to the conclusion of the study.

2. Self-Decisive Algorithm

In the present era, modern digital technology enables the efficient processing and manipulation of multidimensional signals through computational systems. When signal-processing techniques are applied to image data, the process is generally referred to as **image processing**. Image-processing techniques have been widely applied and investigated in various fields, including medicine, engineering, agriculture, and related scientific domains. In [14], an extended split Bregman approach was introduced for image deblurring. Similarly, graph Laplacian regularization was applied in [16] for inverse imaging. Graph-based representations and techniques for image perceptual grouping, object recognition, and image segmentation were investigated in [17]. In [13], an image-processing-based technique was employed to determine the image values of test functions. These image values guide the algorithm to select and execute the most suitable method for a particular function. The working procedure of the algorithm is described as follows.

2.1. Self-Decisive Algorithm

An outline of the self-decisive algorithm is given below:

Step 0: Obtain the image $I(x, y)$to generate the window/matrix S.

Step 1: Compute the covariance matrix $C_{(x,y)} = \frac{1}{W-1}[S^T S]$.

Step 2: Find the eigenvalues of $C_{(x,y)}$.

Step 3: Calculate the value of each pixel as $V(x, y) = (\text{strength } h) + (\text{strength } v)$.

Step 4: Compute the value of each image as $V(I) = \sum_{i=1}^{n} V(x, y)$.

Step 5: Set the threshold value on $V(I)$.

Step 6: Execute the function using the indicated or best-suited method type.

2.2. Test Functions

Two test functions are considered in this study: one quadratic-type function and one trigonometric-type function, namely the **Extended Rosenbrock function** and the **Modified Trigonometric function**, respectively. These functions are selected from the literature [12] with different dimensions ranging from 2 to 150. The corresponding dimensions are summarized in **Table 8**, while the starting points are provided in **Tables 9 and 10**. An epsilon value of $\varepsilon = 10^{-7}$ is considered for both functions. The dimensions are classified into the following categories:

1. **Low dimension:** $2 \leq n \leq 20$
2. **Medium dimension:** $20 < n \leq 60$
3. **High dimension:** $61 \leq n \leq 200$
4. **Combined dimension:** $2 \leq n \leq 200$

To compute the image values of both test functions, images with a resolution of 600×600 are generated. The images and their corresponding image values are shown in **Figure 1**. The mathematical expressions of the two test functions are given below.

a. **Extended Rosenbrock Function**

$$f(x) = \sum_{i=1}^{n/2} \left[100\left(x_{2i} - x_{2i-1}^2\right)^2 + (1 - x_{2i-1})^2\right].$$

b. **Modified Trigonometric Function**

$$f(x) = n^2 - \sum_{i=1}^{n} [\cos(x_i) + i(1 - \cos x_i) - \sin(x_i) + e^{x_i} - 1]^2.$$

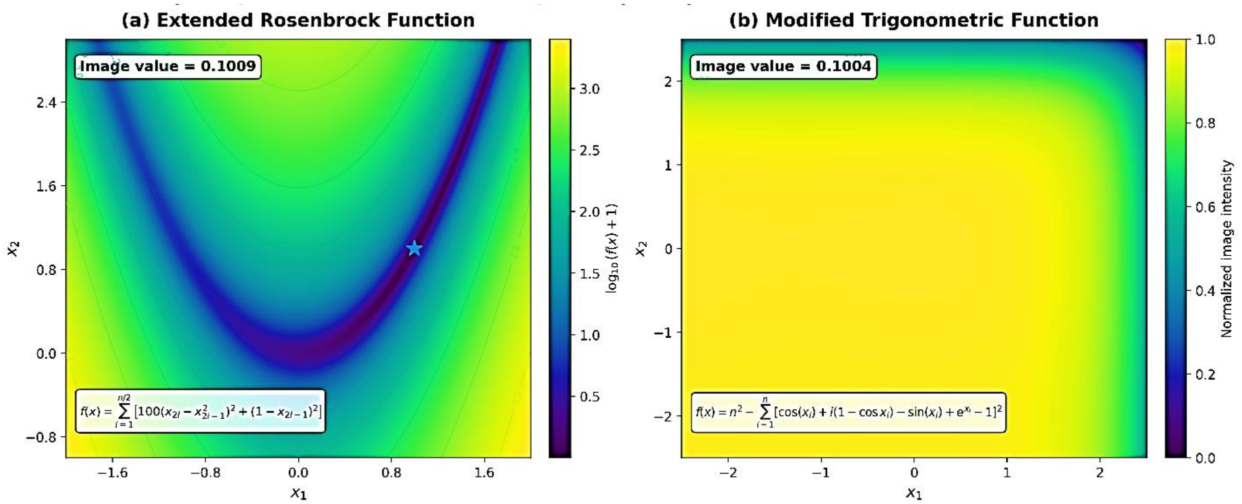

Figure 1: Image representations and corresponding image values of the selected test functions: (a) Extended Rosenbrock function with image value 0.1009; (b) Modified Trigonometric function with image value 0.1004.

3. **Proposed Approach**

The parameters τ given in Eq. (7) were computed by Ford and Moghrabi [5] using two metric-based approaches, among which the fixed-point approach was explored as an effective method. Aamir and Ford [2] also investigated the two-step quasi-Newton method using the fixed-point approach. In [13], the results of the accumulative approach were obtained for two-step quasi-Newton methods. In this research, the fixed-point approach is applied to different techniques of the two-step quasi-Newton method to calculate the parametric values. The obtained results are then compared with those of the accumulative approach [13] to identify the best-fit method.

3.1. Fixed-Point Approach

In this approach, the distance of each iterate is computed from a fixed point. The latest iterate x_{i+1} is kept as the fixed point, while $\tau_m = 0$ is taken as the base point or origin. The remaining parameters τ_k are then computed accordingly. The distance from x_{i+1} to x_i can be measured using the following relation:

$$\tau_j = -\phi_N\left(x_{i+1}, x_{i+j-m+1}\right), \text{ for } j = 0,1, \dots, m. \quad (14)$$

The algorithms based on the fixed-point approach are represented by F_1, F_2, and F_3. Here, $\tau_2 = 0$ is taken as the base point, while τ_1 and τ_0 are obtained from Eq. (14) for $j = 0,1$.

- **Algorithm F_1**

In this algorithm, the matrix N is chosen as the identity matrix I. Therefore,

$$\tau_1 = -\phi_I(x_{i+1}, x_i) = -\| s_i \|_2, \quad (15)$$

and

$$\tau_0 = -\phi_I(x_{i+1}, x_{i-1}) = -\| s_i + s_{i-1} \|_2. \quad (16)$$

- **Algorithm F_2**

In this algorithm, the matrix N is chosen as the current Hessian approximation B_i. Thus,

$$\tau_1 = -\phi_{B_i}(x_{i+1}, x_i),$$

$$\tau_1 = -\sqrt{s_i^T B_i s_i}. \quad (17)$$

The above equation is computationally expensive because it involves a matrix-vector product. However, to compute the parameters more conveniently, the search direction is defined as

$$p_i = -B_i^{-1} g(x_i). \quad (18)$$

Since

$$x_{i+1} = x_i + t_i p_i, \quad (19)$$

we have

$$x_{i+1} - x_i = -t_i B_i^{-1} g(x_i), \quad (20)$$

and therefore,

$$s_i = -t_i B_i^{-1} g(x_i). \quad (21)$$

This implies that

$$B_i s_i = -t_i g(x_i). \quad (22)$$

Using Eq. (22) in Eq. (17), we obtain

$$\tau_1 = -\sqrt{-t_i s_i^T g(x_i)}. \quad (23)$$

The computation of the above relation is relatively simple. However, in the computation of τ_0, the term $s_{i-1}^T B_i s_{i-1}$ is computationally expensive to evaluate at each iteration. Therefore, Ford and Moghrabi [6] showed that, in the case of multi-step quasi-Newton methods, when Eq. (1) is not exactly satisfied, replacing $i+1$ with i in Eq. (1) allows the relation to be approximately satisfied. Hence, we obtain

$$B_i s_{i-1} \approx y_{i-1}. \quad (24)$$

Now,

$$\tau_0 = -\phi_{B_i}(x_{i+1}, x_{i-1}),$$

$$\tau_0 = -\sqrt{(s_i + s_{i-1})^T B_i (s_i + s_{i-1})}.$$

Using Eq. (22) and Eq. (24), we obtain

$$\tau_0 = -\sqrt{-t_i s_i^T g(x_i) - 2t_i s_{i-1}^T g(x_i) + s_{i-1}^T y_{i-1}}.$$

- **Algorithm F_3**

In F_3, the matrix N is chosen as the updated Hessian approximation B_{i+1} at x_{i+1}. Therefore, the parameters τ_1 and τ_0 are computed as follows:

$$\tau_1 = -\phi_{B_{i+1}}(x_{i+1}, x_i),$$

$$\tau_1 \approx -\sqrt{s_i^T y_i}. \quad (25)$$

Similarly,

$$\tau_0 = -\phi_{B_{i+1}}(x_{i+1}, x_{i-1}),$$

$$\tau_0 \approx -\sqrt{s_i^T y_i + 2s_{i-1}^T y_i + s_{i-1}^T y_{i-1}}.$$

4. Numerical Analysis of Test Functions

Different methods are applied to two test functions for minimization purposes. To examine the performance of these functions, two techniques of the two-step quasi-Newton method based on the fixed-point approach are considered:

a. skipping with no modified search direction;
b. skipping with modified search direction.

The results are evaluated in terms of the number of iterations, function evaluations, computational time, and failure rate. The notation of the methods used for different techniques is given in Table 1.

Table 1. Notation of Methods with Different Techniques

Notation	Description
${}^{[1]}_{[0]}F^{(2)}_{n}$	Fixed-point two-step method with one update skipped and no modified search direction
${}^{[1]}_{[1]}F^{(2)}_{n}$	Fixed-point two-step method with one update skipped and modified search direction

where $n = 1,2,3$corresponds to the matrices I, B_i, and B_{i+1}, respectively.

Table 2. Results of the Rosenbrock Function for All-Dimensional Problems Using the First Technique in the Two-Step Method

Method	Function Eval.	Iteration	Time (sec)	Failure	Dimension
${}^{[1]}_{[0]}F^{(2)}_{1}$	52	41	0.0203	0	Soft
${}^{[1]}_{[0]}F^{(2)}_{2}$	65	49	0.0271	0	Soft
${}^{[1]}_{[0]}F^{(2)}_{3}$	61	47	0.0265	0	Soft
${}^{[1]}_{[0]}F^{(2)}_{1}$	64	46	0.0171	0	Medium
${}^{[1]}_{[0]}F^{(2)}_{2}$	61	43	0.0184	0	Medium
${}^{[1]}_{[0]}F^{(2)}_{3}$	64	45	0.0184	0	Medium
${}^{[1]}_{[0]}F^{(2)}_{1}$	64	46	0.0278	0	Hard
${}^{[1]}_{[0]}F^{(2)}_{2}$	61	43	0.0222	0	Hard
${}^{[1]}_{[0]}F^{(2)}_{3}$	64	45	0.0233	0	Hard

4.1. Experimental Results and Discussion on the Rosenbrock Function

From Table 2, it is observed that the method ${}^{[1]}_{[0]}F^{(2)}_{1}$ performs better in terms of function evaluations, number of iterations, and computational time for low-dimensional problems. As the dimension increases, the method ${}^{[1]}_{[0]}F^{(2)}_{2}$ shows improved performance in terms of function evaluations and iterations. However, the minimum computational time is observed for the method ${}^{[1]}_{[0]}F^{(2)}_{1}$. The results of the second technique, presented in Table 3, show that for low-dimensional problems, the method ${}^{[1]}_{[1]}F^{(2)}_{1}$ performs well in terms of iterations, function evaluations, and computational time. As the dimension increases, the best results in terms of function evaluations and iterations are obtained by the method ${}^{[1]}_{[1]}F^{(2)}_{3}$, while the minimum computational time is noted for ${}^{[1]}_{[1]}F^{(2)}_{1}$. The analysis of the Rosenbrock function reveals encouraging results for the second technique, namely one-step skipping with modified search direction.

Table 3. Results of the Rosenbrock Function for All-Dimensional Problems Using the Second Technique in the Two-Step Method

Method	Function Eval.	Iteration	Time (sec)	Failure	Dimension
${}_{[1]}^{[1]}F_1^{(2)}$	48	33	0.0185	0	Soft
${}_{[1]}^{[1]}F_2^{(2)}$	60	43	0.0251	0	Soft
${}_{[1]}^{[1]}F_3^{(2)}$	54	33	0.0211	0	Soft
${}_{[1]}^{[1]}F_1^{(2)}$	49	39	0.0166	0	Medium
${}_{[1]}^{[1]}F_2^{(2)}$	49	43	0.0189	0	Medium
${}_{[1]}^{[1]}F_3^{(2)}$	45	39	0.0171	0	Medium
${}_{[1]}^{[1]}F_1^{(2)}$	49	43	0.0210	0	Hard
${}_{[1]}^{[1]}F_2^{(2)}$	49	43	0.0231	0	Hard
${}_{[1]}^{[1]}F_3^{(2)}$	45	39	0.0353	0	Hard

4.2. Experimental Results and Discussion on the Modified Trigonometric Function

It is evident from Table 4 that the method ${}_{[0]}^{[1]}F_1^{(2)}$ produces the best results in terms of function evaluations, iterations, and computational time for low-dimensional problems. For medium-dimensional problems, ${}_{[0]}^{[1]}F_1^{(2)}$ gives the minimum number of function evaluations, while ${}_{[0]}^{[1]}F_3^{(2)}$ performs better in terms of iterations and computational time. As the dimension increases, ${}_{[0]}^{[1]}F_1^{(2)}$ continues to show promising results in terms of iterations, function evaluations, and computational time. Table 5 reveals that the method ${}_{[1]}^{[1]}F_2^{(2)}$ outperforms the other methods for all-dimensional problems in terms of function evaluations, iterations, and computational time. The numerical results for the Modified Trigonometric function show that, under the first technique, namely one-step skipping with no modification in the search direction, minimum values are obtained in terms of iterations and computational time. On the other hand, the second technique gives the best results in terms of function evaluations.

4.3. Comparison of the Two-Step Quasi-Newton Method with Different Techniques Using the Metric-Based Method

The two-step quasi-Newton method with two different techniques is investigated using two metric based approaches. In [13], the accumulative approach was used and produced good results. In this paper, the fixed-point approach is implemented to obtain numerical results in terms of function evaluations, iterations, and computational time for two test functions. The best results of both approaches for the two functions are reported in Tables 6 and 7. The comparison of the two test functions under both techniques is summarized as follows:

- From Table 6, it is observed that the best method for executing the Rosenbrock function is the skipping technique with modified search direction using the accumulative approach.
- Table 7 shows that the Modified Trigonometric function performs better with the skipping technique without modification in the search direction using the fixed-point approach.
- Moreover, the approximate Hessian matrix B_i performs significantly better than the other two choices used for the weighted matrix N.

Table 4. Results of the Modified Trigonometric Function for All-Dimensional Problems Using the First Technique in the Two-Step Method

Method	Function Eval.	Iteration	Time (sec)	Failure	Dimension
${}^{[1]}_{[0]}\boldsymbol{F}^{(2)}_{1}$	138	89	0.0306	0	Soft
${}^{[1]}_{[0]}\boldsymbol{F}^{(2)}_{2}$	150	91	0.0366	0	Soft
${}^{[1]}_{[0]}\boldsymbol{F}^{(2)}_{3}$	104	73	0.0247	0	Soft
${}^{[1]}_{[0]}\boldsymbol{F}^{(2)}_{1}$	137	97	0.0428	0	Medium
${}^{[1]}_{[0]}\boldsymbol{F}^{(2)}_{2}$	144	97	0.0439	0	Medium
${}^{[1]}_{[0]}\boldsymbol{F}^{(2)}_{3}$	141	89	0.0414	0	Medium
${}^{[1]}_{[0]}\boldsymbol{F}^{(2)}_{1}$	153	108	0.1324	0	Hard
${}^{[1]}_{[0]}\boldsymbol{F}^{(2)}_{2}$	194	127	0.1626	0	Hard
${}^{[1]}_{[0]}\boldsymbol{F}^{(2)}_{3}$	198	127	0.1670	0	Hard

Table 5. Results of the Modified Trigonometric Function for All-Dimensional Problems Using the Second Technique in the Two-Step Method

Method	Function Eval.	Iteration	Time (sec)	Failure	Dimension
${}^{[1]}_{[1]}\boldsymbol{F}^{(2)}_{1}$	105	102	0.0296	0	Soft
${}^{[1]}_{[1]}\boldsymbol{F}^{(2)}_{2}$	90	83	0.0335	0	Soft
${}^{[1]}_{[1]}\boldsymbol{F}^{(2)}_{3}$	109	104	0.0391	0	Soft
${}^{[1]}_{[1]}\boldsymbol{F}^{(2)}_{1}$	135	131	0.0489	0	Medium
${}^{[1]}_{[1]}\boldsymbol{F}^{(2)}_{2}$	124	109	0.0466	0	Medium
${}^{[1]}_{[1]}\boldsymbol{F}^{(2)}_{3}$	136	131	0.0508	0	Medium
${}^{[1]}_{[1]}\boldsymbol{F}^{(2)}_{1}$	148	147	0.1687	0	Hard
${}^{[1]}_{[1]}\boldsymbol{F}^{(2)}_{2}$	124	116	0.1298	0	Hard
${}^{[1]}_{[1]}\boldsymbol{F}^{(2)}_{3}$	144	143	0.1687	0	Hard

Table 6. Best Results of the Rosenbrock Function for All-Dimensional Problems

Technique	Method	Function Eval.	Iteration	Time (sec)	Dimension
First technique	${}^{[1]}_{[0]}A^{(2)}_{1}$	60	43	0.0243	Soft
First technique	${}^{[1]}_{[0]}F^{(2)}_{1}$	52	41	0.0203	Soft
First technique	${}^{[1]}_{[0]}A^{(2)}_{1}$	65	46	0.0180	Medium
First technique	${}^{[1]}_{[0]}A^{(2)}_{2}$	44	46	0.0256	Medium
First technique	${}^{[1]}_{[0]}F^{(2)}_{1}$	64	46	0.0171	Medium
Second technique	${}^{[1]}_{[1]}A^{(2)}_{1}$	62	30	0.0283	Soft
Second technique	${}^{[1]}_{[1]}F^{(2)}_{1}$	48	33	0.0185	Soft
Second technique	${}^{[1]}_{[1]}A^{(2)}_{2}$	49	39	0.0158	Medium

Second technique	${}^{[1]}_{[1]}F^{(2)}_{1}$	49	39	0.0166	Medium
Second technique	${}^{[1]}_{[1]}F^{(2)}_{3}$	45	39	0.0171	Medium
Second technique	${}^{[1]}_{[1]}A^{(2)}_{2}$	49	39	0.0309	Hard
Second technique	${}^{[1]}_{[1]}F^{(2)}_{3}$	45	39	0.0353	Hard

Table 7. Best Results of the Modified Trigonometric Function for All-Dimensional Problems

Technique	Method	Function Eval.	Iteration	Time (sec)	Dimension
First technique	${}^{[1]}_{[0]}A^{(2)}_{1}$	104	75	0.0312	Soft
First technique	${}^{[1]}_{[0]}F^{(2)}_{1}$	52	41	0.0203	Soft
First technique	${}^{[1]}_{[0]}F^{(2)}_{3}$	104	73	0.0247	Soft
Second technique	${}^{[1]}_{[1]}A^{(2)}_{1}$	110	109	0.0400	Soft
Second technique	${}^{[1]}_{[1]}A^{(2)}_{2}$	108	103	0.0544	Soft
Second technique	${}^{[1]}_{[1]}F^{(2)}_{1}$	105	102	0.0296	Soft
Second technique	${}^{[1]}_{[1]}F^{(2)}_{2}$	90	83	0.0335	Soft

Table 8. Test Problems and Dimensions [18]

Function Name	Dimensions
Extended Rosenbrock	2, 20, 26, 40, 60, 80, 100, 120
Modified Trigonometric Function	16, 32, 64, 95, 128, 150

Table 9. First Test Problem with Different Dimensions and Starting-Point Sets

Dimension Class	Dimension	Starting Point [a]	Starting Point [b]	Starting Point [c]	Starting Point [d]	Function Name
Soft	2	(−1.2,1.0)	(−120,100)	(20, −20)	(6.39, −0.221)	Rosenbrock

Rosenbrock	Rosenbrock	Rosenbrock	Rosenbrock
$(-1,-1,-1,-1,-1,1,\ldots,1)$	$[6.39,-0.221]$	$[20]$	$[6.39,-0.221]$
$[6.39,-0.221]$	$[20]$	$[1,-2,3,-4,\ldots,-10]$	$[F]$
$(1,2,\ldots,20)$	$[F]$	$[-120,100]$	$[F]$
$[-1.2,1.0]$	$[-1.2,1.0]$	$[-1.2,1]$	$[-1.2,1]$
20	26	40	60
Soft	**Medium**	**Medium**	**Medium**

ISSN (e) 3007-3138 (p) 3007-312X Volume 4, Issue 5, 2026

Hard	80	$[-1.2, -1.0]$	$[F]$	$[F]$	$[F]$	Rosenbrock
Hard	100	$[-1.2, -1]$	$[F]$	$[F]$	$[F]$	Rosenbrock
Hard	120	$[-1.2, -1]$	$[20]$	$[F]$	$[6.39, -0.221]$	Rosenbrock

IEER Institute for Excellence in Education & Research

Table 10. Second Test Problem with Different Dimensions and Starting-Point Sets

Dimension Class	Dimension	ε	Starting Point [a]	Starting Point [b]	Starting Point [c]	Starting Point [d]	Function Name

Modified Trigonometric	[2.5,2,1.5,1,0.5],2.5	[0.1,1,−0.1,−1]	[−2,1.5, … , −1.5,2]	[−2, −1,1,2]	10^{-7}	16	Medium
Modified Trigonometric	[2.5,2, … ,0.5],2.5,2	[0.1,1,−0.1,1]	[−2,1.5, … , −1.5,2]	[−2, −1,1,2]	10^{-7}	32	Medium
Modified Trigonometric	[2.5,2, … ,0.5],2.5, … ,1	[0.1,1,−0.1,1]	[−2,1.5, … , −1.5,2]	[−2, −1,1,2]	10^{-6}	64	Medium
Modified Trigonometric	[2.5, 2.0, 1.5, 1.0, 0.5]	[0.1,1.0,−0.1,1.0]	[−2,1.5, … ,2]	[−2, −1,1,2],−2, −1,1	10^{-5}	95	Hard

Modified Trigonometric	[2.5,2,1.5,1,0.5],2.5,2,1.5	[0.1,1,−0.1,1]	[−2,1.5,…,−1.5,2]	[−2,−1,1,2],−2,−1,1	10^{-6}	128	Hard
Modified Trigonometric	[2.5, 2.0, 1.5, 1.0, 0.5]	[0.1,1,−0.1,1],0.1,1	[−2,1.5,…,2],−2,…,−0.5,1	[−2,−1,1,2],−2,−1	10^{-5}	150	Hard

5. Machine-Learning Perspective of the Proposed Self-Decisive Framework

The proposed self-decisive algorithm can be interpreted as an AI-assisted decision mechanism within the framework of numerical optimization. In machine learning, optimization algorithms play a central role in minimizing loss functions, updating model parameters, and improving prediction accuracy. Therefore, the selection of an efficient optimization method is directly connected to the computational performance of machine-learning models. In this study, image-processing features are used to represent the structural behavior of the selected test functions. The computed image value $V(I)$serves as an informative descriptor that helps the algorithm identify the function pattern and select the most suitable two-step quasi-Newton technique. In this sense, the proposed approach reflects a machine-learning-inspired strategy, where useful features are extracted from function images and used to support method selection. Unlike conventional optimization procedures, where a fixed method is often applied to all problems, the self-decisive framework introduces an adaptive selection mechanism. This makes the algorithm more intelligent, since it can distinguish between different function patterns and guide the execution toward the most appropriate technique. Such a mechanism is particularly useful in large-scale optimization and machine-learning applications, where computational cost, number of iterations, function evaluations, and execution time are important performance indicators. The role of the image value is similar to that of a feature descriptor in machine learning. It provides compact information about the behavior of the objective function and assists in selecting between skipping and modified search-direction strategies. Thus, the proposed method establishes a meaningful connection between image-based feature extraction and numerical optimization, offering a pathway toward more adaptive and intelligent quasi-Newton methods.

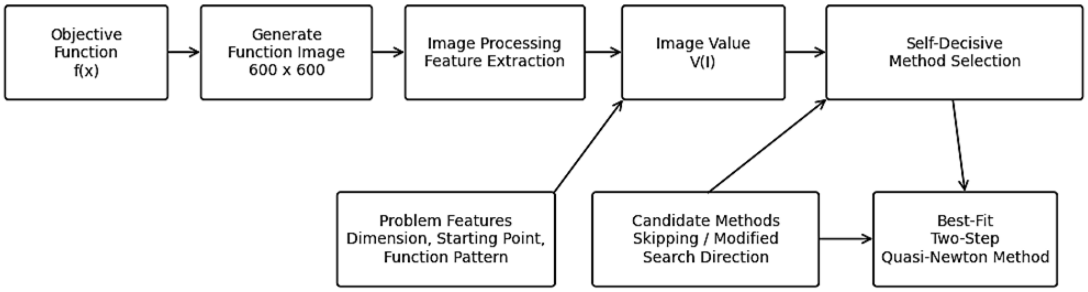

Figure 2: Machine-learning perspective of the self-decisive optimization framework. Image-based features and numerical problem information guide the adaptive selection of the most suitable two-step quasi-Newton technique.

5.1. Future Machine-Learning Extension

The current self-decisive framework is based on image-value-guided decision-making. In future work, this mechanism may be extended into a supervised machine-learning model. For example, a classifier or regression model may be trained using previous optimization experiments. The input features may include image value, function type, dimension, initial point, number of iterations, function evaluations, computational time, and failure rate. Based on these features, the trained model may predict the most efficient optimization strategy for a new objective function.

Such an extension would transform the current rule-based self-decisive mechanism into a data-driven optimizer selection framework. This would allow the algorithm to learn from previous numerical experiments and automatically recommend the best two-step quasi-Newton variant for a given problem. Consequently, the proposed framework may serve as a foundation for developing intelligent optimization systems for machine learning, engineering computation, and large-scale nonlinear unconstrained optimization.

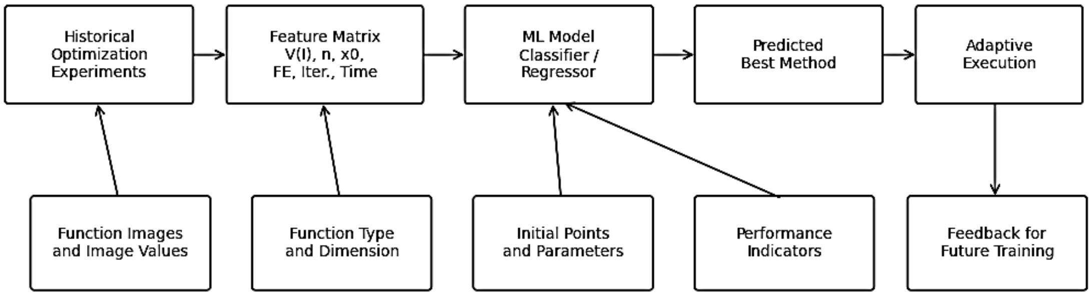

Figure 3: Possible future machine-learning extension for optimizer selection. A supervised-learning model can learn from previous optimization experiments and predict the most efficient method automatically.

6. Conclusion

Machine learning algorithms process large sets of input and output data to identify patterns, learn from experience, and generate autonomous recommendations or decisions. In the same spirit, this study introduced an AI-assisted self-decisive framework for selecting suitable optimization strategies within the structure of two-step quasi-Newton methods. In this work, two techniques, namely skipping and search-direction modification, were investigated using the fixed-point approach. During the computation of the parameters τ, three different choices of the weighted matrix Nwere considered. Among these choices, the best performance was achieved when $N = B_i$, which produced remarkable results, particularly for the quadratic test function compared with the trigonometric function. The numerical evidence obtained from the two selected test functions shows that image values can serve as meaningful decision indicators for identifying the most appropriate method for a particular function. These image values assist the self-decisive algorithm in selecting the best-fit optimization technique, thereby reducing unnecessary computational effort and improving the efficiency of the execution process.

The proposed algorithm provides a promising pathway toward more intelligent and adaptive quasi-Newton optimization methods. By connecting image-based feature extraction with numerical optimization, this study opens the door for future machine-learning-based optimizer selection frameworks, where function characteristics, image values, dimensions, and computational performance measures can be used to automatically predict the most efficient optimization strategy.

REFERENCES

[1] Aamir, N., & Ford, J. (2013). *Multi-step skipping methods for unconstrained non-linear optimization* [PhD thesis, Department of Mathematical Sciences, University of Essex, UK].

[2] Aamir, N., & Ford, J. (2021). Two-step skipping techniques for solution of nonlinear unconstrained optimization problems. *International Journal of Engineering Works, 8*(06), 170–174.

[3] Albaali, M., & Khalfan, H. (2007). An overview of some practical quasi-Newton methods for unconstrained optimization. *SQU Journal for Science, 12*(2), 199–209.

[4] Ford, J. (2001). Implicit updates in multi-step quasi-Newton methods. *Journal of Computational Mathematics and Applications, 42*, 1083–1091.

[5] Ford, J., & Moghrabi, I. (1993). Alternative parameter choices for multi-step quasi-Newton methods. *Optimization Methods and Software, 2*, 357–370.

[6] Ford, J., & Moghrabi, I. (1994). Multi-step quasi-Newton methods for optimization. *Journal of Computational and Applied Mathematics, 50*, 305–323.

[7] Ford, J., & Moghrabi, I. (1996). Minimum curvature multi-step quasi-Newton methods. *Computers & Mathematics with Applications, 31*, 179–186.

[8] Ford, J., & Tharmlikit, S. (2003). New implicit updates in multi-step quasi-Newton methods for unconstrained optimization. *Journal of Computational and Applied Mathematics, 152*, 133–146.

[9] Griva, I., Nash, S., & Sofer, A. (2009). *Linear and nonlinear optimization* (2nd ed.). Society for Industrial and Applied Mathematics.

[10] Guo, T.-D., Liu, Y., & Han, C.-Y. (2023). An overview of stochastic quasi-Newton methods for large-scale machine learning. *Journal of the Operations Research Society of China, 11*, 245–275.

[11] Helm, J. M., Swiergosz, A. M., Haeberle, H. S., Karnuta, J. M., Schaffer, J. L., Krebs, V. E., Spitzer, A. I., & Ramkumar, P. N. (2020). Machine learning and artificial intelligence: Definitions, applications, and future directions. *Current Reviews in Musculoskeletal Medicine, 13*, 69–76.

[12] Hillstrom, K., More, J., & Garbow, B. (1981). Testing unconstrained optimization software. *ACM Transactions on Mathematical Software, 7*(1), 136–140.

[13] Jaffar, F., Mashwani, W. K., Al-Marzouki, S. M., Aamir, N., & Abiad, M. (2022). Self decisive algorithm for unconstrained optimization problems as in biomedical image analysis. *Frontiers in Computational Neuroscience, 16*.

[14] Liu, X., & Huang, L. (2010). Split Bregman iteration algorithm for total bounded variation regularization based image deblurring. *Journal of Mathematical Analysis and Applications, 372*(2), 486–495.

[15] Moghrabi, I. (2006). Exploiting function values in multi-step methods. *Journal of Pure and Applied Mathematics, 28*(2), 187–196.

[16] Pang, J., & Cheung, G. (2017). Graph Laplacian regularization for inverse imaging: Analysis in the continuous domain. *IEEE Transactions on Image Processing, 26*(4), 1770–1785.

[17] Sanfeliu, A., Alquezar, R., Andrade, J., Climent, J., Serratosa, F., & Verges, J. (2002). Graph-based representations and techniques for image processing and image analysis. *Pattern Recognition, 35*, 639–650.

[18] Toint, P. (1987). On large scale nonlinear least squares calculation. *SIAM Journal on Scientific Computing, 8*(3), 416–435.

IEER
Institute for Excellence in Education & Research