frontiers | Frontiers in Computational Neuroscience

TYPE Original Research
PUBLISHED 03 October 2022
DOI 10.3389/fncom.2022.994161

OPEN ACCESS

EDITED BY
Abdul Mueed Hafiz,
University of Kashmir, India

REVIEWED BY
Asim Muhammad,
Guangdong University of Technology,
China
Atilla Göktaş,
Muğla University, Turkey

*CORRESPONDENCE
Wali Khan Mashwani
mashwanigr8@gmail.com
Mohammad Abiad
Mohammad.abiad@aum.edu.kw

RECEIVED 14 July 2022
ACCEPTED 27 July 2022
PUBLISHED 03 October 2022

CITATION
Jaffar F, Mashwani WK, Al-marzouki SM, Aamir N and Abiad M (2022) Self-decisive algorithm for unconstrained optimization problems as in biomedical image analysis. *Front. Comput. Neurosci.* 16:994161. doi: 10.3389/fncom.2022.994161

Self-decisive algorithm for unconstrained optimization problems as in biomedical image analysis

Farah Jaffar[1], Wali Khan Mashwani[2]*, Sanaa Mohammed Al-marzouki[3], Nudrat Aamir[4] and Mohammad Abiad[5]*

[1]Department of Mathematics, Shaheed Benazir Bhutto Women University, Peshawar, Pakistan, [2]Institute of Numerical Sciences, Kohat University of Science and Technology, Kohat, Pakistan, [3]Department of Statistics, Faculty of Sciences, King Abdulaziz University, Jeddah, Saudi Arabia, [4]Department of Basic Sciences and Humanities, CECOS University of IT and Emerging Sciences, Peshawar, Pakistan, [5]Department of Mathematics and Statistics, American University of the Middle East, Egaila, Kuwait

This study describes the construction of a new algorithm where image processing along with the two-step quasi-Newton methods is used in biomedical image analysis. It is a well-known fact that medical informatics is an essential component in the perspective of health care. Image processing and imaging technology are the recent advances in medical informatics, which include image content representation, image interpretation, and image acquisition, and focus on image information in the medical field. For this purpose, an algorithm was developed based on the image processing method that uses principle component analysis to find the image value of a particular test function and then direct the function toward its best method for evaluation. To validate the proposed algorithm, two functions, namely, the modified trigonometric and rosenbrock functions, are tested on variable space.

KEYWORDS

quasi-Newton method, multi-step quasi-Newton method, medical informatics, image processing, covariance matrix, eigenvalues

1. Introduction

Imaging informatics plays a significant role in medical and engineering fields. In the diagnostic application software, during the segmentation procedure, different tools are used to interact with a visualized image and a graphical user interface (GUI) is used to parameterize the algorithms and for the visualization of multi-modal images and segmentation results in 2D and 3D. Hence, different toolkits, such as Medical Interaction Toolkit (Wolf et al., 2005) or the MevisLab (Ritter et al., 2011), are used to build appropriate GUIs, yielding an interface to integrate new algorithms from science to application. To produce better results, we used different sensors to generate a perfect image for denoising the tasks, thus focusing on massive denoising, as sometimes it is difficult for humans and computers to recognize the image. Furthermore, different filters (Wang, 2018) are used to get better denoising. The commonly used mathematical method for this purpose is the quasi-Newton method, which is preferred due to its better performance than other classical methods. Schröter and Sauer (2010) investigated quasi-Newton algorithms for medical image registration. Mannel and Rund (2021) implemented a hybrid semi-smooth quasi-Newton method for the non-smooth optimal control problems and proved its efficiency. Recently, Moghrabi et al. (2022) derived self-scaled quasi-Newton methods and proved their efficiency over the non-scaled version. Among the quasi-Newton methods, Broyden-Fletcher-Goldfarb-Shanno (BFGS) is a widely used method due to its better performance. Hence, this motivated the researchers to further develop these methods, one such method is the two-step quasi-Newton method.

The two-step quasi-Newton methods are considered to minimize unconstrained optimization problems.

$$\text{minimize} f(x), \quad \text{where} \quad f: R^n \to R. \tag{1}$$

The multi-step quasi-Newton methods, which were introduced by Ford and Moghrabi (1994, 1993), obtained the best results over the single-step quasi-Newton method. In the single-step quasi-Newton method, updation of Hessian approximation (B_{i+1}) is required to satisfy the secant equation

$$\boldsymbol{B_{i+1} s_i = y_i}, \tag{2}$$

where $\boldsymbol{s_i}$ is the step size in the variable space $\boldsymbol{x_i}$

$$\boldsymbol{s_i = x_{i+1} - x_i}, \tag{3}$$

and $\boldsymbol{y_i}$ is the step size in gradient space $\boldsymbol{g(x_i)}$, such as

$$\boldsymbol{y_i = g(x_{i+1}) - g(x_i)}. \tag{4}$$

The quasi-Newton equation must satisfy the true Hessian $\boldsymbol{M_{i+1}}$ of Newton equation which is defined as follows:

$$\boldsymbol{M(x_{i+1})\frac{dx(\tau*)}{d\tau} = \frac{dg(x(\tau*))}{d\tau}} \tag{5}$$

In the case of the two-step quasi-Newton methods, the secant equation (2) is replaced by

$$\boldsymbol{B_{i+1}(s_i - \alpha_i s_{i-1}) = y_i - \alpha_i y_{i-1}}, \tag{6}$$

or

$$\boldsymbol{B_{i+1} r_i = w_i}, \tag{7}$$

which is derived by interpolating the quadratic curve $x(\tau)$ in variable space and $g(\tau)$ in gradient space

$$\boldsymbol{x(\tau_k) = x_{i-m+k+1}}, \quad \mathit{for} \quad k = 0, 1, 2. \tag{8}$$

$$\boldsymbol{g(x(\tau_k)) = g(x_{i-m+k+1})}, \quad \mathit{for} \quad k = 0, 1, 2. \tag{9}$$

for the purpose of interpolating the Lagrange polynomial that is found suitable in Jaffar and Aamir (2020). Therefore,

$$\boldsymbol{x(\tau) = \sum_{k=0}^{2} \mathcal{L}_k(\tau) x_{i+k-1}}, \tag{10}$$

$$\boldsymbol{g(x(\tau)) \approx \sum_{k=0}^{2} \mathcal{L}_k(\tau) g(x_{i+k-1})}, \tag{11}$$

the derivatives of Equations (10) and (11) are defined as

$$\boldsymbol{\frac{dx(\tau_2)}{d\tau} \equiv r_i} \tag{12}$$

$$\boldsymbol{\frac{dgx(\tau_2)}{d\tau} \equiv w_i}, \text{ and} \tag{13}$$

the above relations obtained are substituted in Equation (14), which is a two-step form of Equation (5)

$$\boldsymbol{M(x_{i+1})\frac{dx(\tau_2)}{d\tau} = \frac{dg(x(\tau_2))}{d\tau}} \tag{14}$$

The new secant condition for the two-step quasi-Newton method is obtained in the form of Equation (6/7), which should be satisfied by the updated Hessian approximation $\boldsymbol{B_{i+1}}$. The value of α_i in Equation (6) is given by

$$\alpha_i = \frac{\delta^2}{(2\delta + 1)}, \tag{15}$$

where

$$\delta = \frac{(\tau_2 - \tau_1)}{(\tau_1 - \tau_0)}. \tag{16}$$

Hence, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula for the two-step method is defined as

$$\boldsymbol{B_{i+1} = B_i + \frac{B_i r_i r_i^T B_i}{r_i^T B_i r_i} + \frac{w_i w_i^T}{r_i^T w_i}} \tag{17}$$

The standard Lagrange polynomial $\mathcal{L}_k$ depends on the values of $\{\tau_k\}_{k=0}^{m}$ and is defined as

$$\mathcal{L}_k(\tau) \equiv \prod_{j=0, j\neq k}^{m} \frac{\tau - \tau_j}{\tau_k - \tau_j}. \tag{18}$$

The parametric values τ_k, for k= 0, 1, 2....,m, used in the computation of vectors, $\boldsymbol{r_i}$ and $\boldsymbol{w_i}$, are found by the metric of the form

$$\phi_N(z_1, z_2) = ((z_1 - z_2)^T \boldsymbol{N}(z_1 - z_2))^{\frac{1}{2}}. \tag{19}$$

The matrix $\boldsymbol{N}$ is a positive definite matrix, and three choices are given for $\boldsymbol{N}$ as $\boldsymbol{I}$, $\boldsymbol{B_i}$, and $\boldsymbol{B_{i+1}}$ on variable space and $z_1, z_2 \quad \epsilon \; R^n$.

This metric between different iterates in the current interpolation is measured by fixed-point and accumulative approaches (Ford and Moghrabi, 1994). In this study, an accumulative approach is used for finding the parametric values.

• **Accumulative Methods**

These methods accumulate the distance between the consecutive iterates in their natural sequence. The latest iterate $\boldsymbol{x}_{i+1}$, corresponding to the value τ_m of τ, is considered as the origin or base point, and the other values of τ are calculated by accumulating the distance between the consecutive pairs. Therefore, we have

$$\tau_m = 0, \tag{20}$$

$$\tau_k = \tau_{k+1} - \phi_N(\boldsymbol{x_{i-m+k+2}}, \boldsymbol{x_{i-m+k+1}}), \text{for} \quad k = 0, 1, ...m-1. \tag{21}$$

In the two-step method, the accumulative type is determined as A1, A2, and A3, and the parametric values are found with the help of Equation (21) for k=0,1, where the base point will be $\tau_2 = 0$ for m=2 from Equation (20).

• **Algorithm A1**

The identity matrix $\boldsymbol{I}$ is taken as matrix $\boldsymbol{N}$ in this algorithm.

$$\tau_1 = \tau_2 - \phi_{\boldsymbol{I}}(\boldsymbol{x_{i-2+1+2}}, \boldsymbol{x_{i-2+1+1}}),$$

$$= -\phi_{\boldsymbol{I}}(\boldsymbol{x_{i+1}}, \boldsymbol{x_i}) = -\|\boldsymbol{s_i}\|_2, \tag{22}$$

$$\tau_0 = \tau_1 - \phi_{\boldsymbol{I}}(\boldsymbol{x_i}, \boldsymbol{x_{i-1}}),$$

$$= -\|\boldsymbol{s_i}\|_2 - \|\boldsymbol{s_{i-1}}\|_2. \tag{23}$$

• **Algorithm A2**

In this algorithm, matrix $\boldsymbol{N}$ is taken as the current Hessian approximation $\boldsymbol{B}_i$.

$$\tau_1 = \tau_2 - \phi_{\boldsymbol{B_i}}(\boldsymbol{x_{i+1}}, \boldsymbol{x_i}),$$

$$\Rightarrow \tau_1 = \sqrt{\boldsymbol{s_i^T B_i s_i}}, \tag{24}$$

The above equation involves a matrix vector product which is computationally expensive. For instance, with the help of search direction, we can easily compute the parameters of the same situation as

$$\boldsymbol{p_i} = -\boldsymbol{B_i^{-1} g(x_i)}, \tag{25}$$

since

$$\boldsymbol{x_{i+1}} = \boldsymbol{x_i} + \boldsymbol{t_i p_i}$$

$$= \boldsymbol{x_i} - t_i \boldsymbol{B_i^{-1} g(x_i)} \quad \text{for} \quad t_i > 0,$$

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

$$\boldsymbol{s_i} = -t_i \boldsymbol{B_i^{-1} g(x_i)},$$

$$\Rightarrow \boldsymbol{B_i s_i} = -t_i \boldsymbol{g(x_i)}, \tag{26}$$

By substituting Equation (26) in Equation (24), we get

$$\tau_1 = -\sqrt{t_i \boldsymbol{s_i^T g(x_i)}}, \tag{27}$$

The above expression is easy to calculate but the expression $\boldsymbol{s_{i-1}^T B_i s_{i-1}}$ in τ_0 is very difficult to compute in every iteration. Therefore, to lessen the computational cost, Ford and Moghrabi (1994) claimed that, in multi-step methods, when the quasi-Newton equation in Equation (2) is not satisfied, then we can consider that it satisfies approximately by replacing i+1 with i in Equation (2), where $\boldsymbol{B_i}$ is an approximation of matrix N. Therefore, we obtain

$$\boldsymbol{B_i s_{i-1}} \approx \boldsymbol{y_{i-1}} \tag{28}$$

$$\tau_0 = \tau_1 - \phi_{\boldsymbol{B_i}}(\boldsymbol{x_i}, \boldsymbol{x_{i-1}}), \text{ and}$$

using Equation (28), we have

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

• **Algorithm A3**

In this algorithm, the choice of matrix $\boldsymbol{N}$ is $\boldsymbol{B_{i+1}}$, which is the Hessian approximation at $\boldsymbol{x_{i+1}}$.

$$\tau_1 = \tau_2 - \phi_{\boldsymbol{B_{i+1}}}(\boldsymbol{x_{i+1}}, \boldsymbol{x_i}),$$

$$\tau_1 = -\sqrt{\boldsymbol{s_i^T y_i}}, \tag{29}$$

$$\tau_0 = \tau_1 - \phi_{\boldsymbol{B_{i+1}}}(\boldsymbol{x_i}, \boldsymbol{x_{i-1}}),$$

$$\tau_0 \approx -\sqrt{\boldsymbol{s_i^T y_i}} - \sqrt{\boldsymbol{s_i^T y_{i-1}}}, \tag{30}$$

Since τ_1 and τ_0 are expensive to compute, Equation (2) is used.

Aamir and Ford (2021) investigated the multi-step skipping technique in which one-step and two-step skipping strategies were experimented and produced better results than those without skipping strategy. The authors also modified the search direction, which was implemented with/without the skipping technique to achieve good performance in minimum time duration. We experimented two test functions of different dimensions by the two-step quasi-Newton methods with different techniques, i.e., one-step skipping with no modified search direction and one-step skipping with modified search direction on variable space with a high rate of computational

TABLE 1 Notation of methods with different techniques.

Notation	Discription
${}^{[1]}_{[0]}\mathbf{A}^{(2)}_{n}$	Accumulative two-step method with one update skipped and no modified search direction
${}^{[1]}_{[1]}\mathbf{A}^{(2)}_{n}$	Accumulative two-step method with one update skipped and modified search direction
	where n=1,2,3 corresponds to matrices I, B_i, B_{i+1} respectively on variable space

effort. Therefore, to lessen the computational burden and increase efficiency, an algorithm was developed to execute a particular function by using the best method only. Section 2 discusses the two-step quasi-Newton method with different techniques in detail. Section 3 proposes a self-decisive algorithm which is developed based on image processing method to find the image values of different test functions. Section 4 discusses the experimental setup of the proposed strategy. Section 5 analyzes the numerical results of one function, which can help the algorithm to execute a particular function by the best method only. In the last section, the conclusion is drawn based on different numerical simulations.

2. Two-step quasi-Newton methods with different techniques

Different techniques in two-step methods, such as the one-step skipping technique with no modified search direction and the one-step skipping technique with modified search direction, are implemented on the selected test functions for the purpose of minimization. These functions are examined by function evaluation, the number of iterations, and time in seconds. The notation of different methods on different techniques is given in Table 1.

2.1. Skipping technique

In quasi-Newton methods, updation of Inverse Hessian approximation $\boldsymbol{H_i}$ to $\boldsymbol{H_{i+1}}$ is a very expensive procedure under certain circumstances. Tamara et al. (1998) introduced the idea of skipping updates for certain steps to lessen the burden of computational cost. They investigated the question of "how much and which information can be dropped in BFGS and other quasi-Newton methods without destroying the property of quadratic termination" and called this procedure "backing up." They used this idea in the algorithm if the step length is 1.0 or the current iteration is odd.

Aamir and Ford (2021) investigated the skipping technique in single step and multi-step methods. The experimental results of comparison between the skipping and non-skipping methods revealed that skipping algorithms outperformed non-skipping algorithms.

- **Algorithm of the Multi-step Skipping Method**

The general algorithm of the skipping technique is as follows:

1. Select an initial approximation $\boldsymbol{x_0}$ and $\boldsymbol{H_0}$ and set i=1
2. For j=1: m (where m is the number of steps to be skipped)

- Calculate a search direction $\boldsymbol{p_{i+j-2} = -H_{i-1}g_{i+j-2}}$.
- Find t by giving t_{i+j-2} for executing line search $\boldsymbol{x}_{i+j-2} + t\boldsymbol{p}_{i+j-2}$.
- Calculate new approximation $\boldsymbol{x_j = x_j} + t_j\boldsymbol{p}_j$.

End for

3. By the use of different methods, update $\boldsymbol{H_{i+j-2}}$ to give $\boldsymbol{H_{i+m-1}}$.

4. If $\|\boldsymbol{g}_i\| \leq \epsilon$, then stop, else i:=i+m and go to step 2. End if.

- **Application of the skipping technique on the two-step method**

Now that we are at $\boldsymbol{x_{i+1}}$, the matrix is updated by $\boldsymbol{B_{i-1}, s_i, s_{i-1}, y_i}$, and $\boldsymbol{y_{i-1}}$, using the following steps:
1. Using the above terms, compute τ_k and then find δ.
2. By the use of all the above values, through which we find $\boldsymbol{r_i}$ and $\boldsymbol{w_i}$, we have

$$\boldsymbol{r_i = s_i} - \frac{\delta^2}{(2\delta+1)}\boldsymbol{s_{i-1}}, \tag{31}$$

$$\boldsymbol{w_i = y_i} - \frac{\delta^2}{(2\delta+1)}\boldsymbol{y_{i-1}}, \tag{32}$$

3. The Hessian approximation is updated by using all the above values.

$$\boldsymbol{B_{i+1} = BFGS(B_{i-1}, r_i, w_i)}. \tag{33}$$

Now we compute τ_k and/or δ under different methods.

2.2. The two-step method with skipping and modified search directions

Here, we explained the derivation of the modified search direction. The following notations are used during the derivation of modified search direction in the skipping technique.
$\boldsymbol{\hat{H}_i}$ represents that the matrix is never computed.
$\boldsymbol{\tilde{p}_i}$ represents modified search direction.
Now, let us consider that the single-step BFGS updated the matrix $\boldsymbol{H_{i-1}}$. The search direction $\boldsymbol{p_{i-1}}$ is defined as

$$\boldsymbol{p_{i-1} = -H_{i-1}g_{i-1}}. \tag{34}$$

By using the skipping technique, the next search direction is

$$p_i = -H_{i-1}g_i. \tag{35}$$

With the help of $\hat{H}_i$, we can find the modified search direction $\tilde{p}_i$. We define

$$\hat{H}_i = BFGS(H_{i-1}, s_{i-1}, y_{i-1})$$

$$= H_{i-1}+\lambda_{i-1}s_{i-1}s^T_{i-1}-\alpha_{i-1}(H_{i-1}y_{i-1}s^T_{i-1}+s_{i-1}y^T_{i-1}H_{i-1}), \tag{36}$$

where

$$\lambda_{i-1} = (1 + \frac{y^T_{i-1}H_{i-1}y_{i-1}}{s^T_{i-1}y_{i-1}})\frac{1}{s^T_{i-1}y_{i-1}}$$

$$\alpha_{i-1} = \frac{1}{s^T_{i-1}y_{i-1}}$$

Now, the modified search direction is

$$\tilde{p}_i = -\hat{H}_i g_i$$

$$\begin{aligned} &= H_{i-1}g_i + \lambda_{i-1}s_{i-1}[s^T_{i-1}g_i] \\ &\quad +\alpha_{i-1}(H_{i-1}y_{i-1}[s^T_{i-1}g_i] + s_{i-1}[y^T_{i-1}H_{i-1}g_i]), \end{aligned}$$

and by Equation (35), we get

$$\begin{aligned} &= p_i + \lambda_{i-1}[s^T_{i-1}g_i]s_{i-1} + \alpha_{i-1}([s^T_{i-1}g_i]H_{i-1}y_{i-1} \\ &\quad +[y^T_{i-1}p_i]s_{i-1}), \end{aligned} \tag{37}$$

From the above equation, it can be observed that $H_{i-1}y_{i-1}$ and λ_{i-1} cannot be easily computable due to the matrix vector product than other terms. However, with the help of Equation (34) and Equation (35), the expression $H_{i-1}y_{i-1}$ can be defined as

$$H_{i-1}y_{i-1} = H_{i-1}(g_i - g_{i-1})$$

$$= -p_i + p_{i-1} \tag{38}$$

Therefore, using the above equation in Equation (37), modified search direction can be calculated efficiently without explicitly computing $\hat{H}_i$.

- **Algorithm of the Multi-step Skipping Quasi-Newton method with Modified Search Direction**

The general algorithm is given as follows:

1. Select x_0 and H_0 as an initial approximation; set i=1
2. For j=1:m, where m is the number of steps,
3. Calculate $p_{i+k-2} = -H_{i-1}g_{i+k-2}$.
4. Calculate modified search direction $\tilde{p}_{i+k-2} = -\hat{H}_{i+j-2}g_{i+k-2}$.
5. Do the line search along $x_{i+j-2} + tp_{i+j-2}$ and also providing a value of t_{i+j-2} for t.
6. Calculate new approximation $x_j = x_j + t_j\tilde{p}_i$.
   End
7. Update H_{i-1} to produce H_{i+m-1} by using different methods discussed in previous sections.
8. Check for convergence, if it is not converged, then i=i+1 and go to step no: 2.

3. Image processing

In the viewpoint of image processing "an image is an array or matrix of numeric values called pixels (Picture Element) arranged in columns and rows". In mathematics an image is defined as "a graph of a spatial function" or "it is a two-dimensional function f(x,y), where x and y are the spatial (plane) coordinates, and the amplitude at any pair of coordinates (x,y) is called the intensity of the image at that level." If x,y and the amplitude values of f are finite and discrete quantities, we call the image a digital image. A digital image is composed of a finite number of pixels, each of which has a particular location and value. Image processing is a process in which different mathematical operations are performed subject to application on the image to get improved or to extract significant information from the image for subsequent processing. When this process is applied to digital images is called digital image processing.

Digital image processing has a wide scope for researchers to work on various areas of science (such as, a agriculture, biomedical, and engineering). Previous studies showed that researchers applied and investigated different techniques of image processing for analysis and problem solving, such as detection and measurement of paddy leaf disease symptoms using image processing (Narmadha and Arulvadivu, 2017), breast cancer detection using image processing techniques (Christian et al., 2000), a novel outlier detection method for monitoring data in dam engineering (Shao et al., 2022), and counterfeit electronics detection using image processing and machine learning (Navid et al., 2017).

3.1. Proposed strategy

In the proposed strategy, the algorithm is developed by which the image values of different images I(x, y) of test functions are obtained by statistical technique, and the desired objective is achieved. In the first step, the images of different test functions are obtained with a resolution of 600 × 600 pixels. In the second step, the window of a size $W \times W$ is generated around each pixel of the image I(x, y), where the suggested size of the generated window is 3 × 3 and this window is treated as matrix

S. The rows of the matrix S are considered as observations, and columns are considered as variables. Figure 1 shows the schematic diagram of the matrix S generation.

In the third step, covariance matrix $C^{(x,y)}$ of matrix S is computed with the help of the following equation:

$$C^{(x,y)} = \frac{1}{W-1}[S^t S] \tag{39}$$

In the fourth step, eigenvalues of the covariance matrix are calculated. The sum of the eigenvalues is directly proportional to edge strength, which is calculated as follows:

$$EdgeStrength = \sum_{k=1}^{W} eigen_k of C_h^{(x,y)} \tag{40}$$

The third and fourth steps are done twice, the first time for the horizontal edge strength and the second time for the vertical edge strength calculation. Therefore, Equations (39) and (40) are used for calculating horizontal and vertical edge strength generation as follows:

$$C_h^{(x,y)} = \frac{1}{W-1}[S^t S] \tag{41}$$

$$C_v^{(x,y)} = \frac{1}{W-1}[S^t S] \tag{42}$$

$$EdgeStrength_h = \sum_{k=1}^{W} eigen_k of C_h^{(x,y)} \tag{43}$$

$$EdgeStrength_v = \sum_{k=1}^{W} eigen_k of C_v^{(x,y)} \tag{44}$$

The sum of horizontal and vertical edge strength gives the value of a pixel of I(x, y). Hence, the value of each pixel of an image is calculated as

$$V(x,y) = (EdgeStrength_h) + (EdgeStrength_v), \text{ and} \tag{45}$$

the sum of all pixel values gives the value of an image I(x, y) defined as

$$V(I) = \sum_{i=1}^{n} V(x,y) \tag{46}$$

4. Experimental setup and results

Two test functions were selected from literature and were executed by using different techniques of the two-step quasi-Newton methods on variable space. The execution of test functions by every technique was computationally expensive. Therefore, an algorithm is required to enable the researchers to execute a particular function by the best method only to reduce computational cost.

Hence, our objective is to develop such an algorithm that can compute the image value of every input image of the test function and forward each function to the method by which it outperformed. The algorithm works in the following steps:

1. Obtain the images of each test function.
2. Compute the image value of each image.
3. Classify test functions by their image values.
4. Forward the particular function toward the best method.

4.1. Test functions

To check the performance of different techniques used in two-step methods, we considered two test functions of different dimensions with four different starting points and epsilon value from the literature (Hillstrom et al., 1981), which are reported in Tables 2, 3. These test functions are categorized into three classes, namely soft, medium, and hard.

1. Soft: $(2 \leq n \leq 20)$;
2. Medium: $(20 \leq n \leq 60)$;
3. Hard: $(61 \leq n \leq 150)$;
4. Combined: $(2 \leq n \leq 150)$.

The equations of both test functions are mentioned below by which 600 × 600 resolution images (displayed in Figure 2) are generated to calculate the image value of each function (as reported in Table 4) and which are programmed successfully in the self-decisive algorithm.

1. Extended Rosenbrock function:

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

2. Modified Trigonometric function:

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

4.2. Self decisive algorithm

An outline of the self decisive algorithm can be defined as follows:

Step 0: Obtain image I(x, y) to generate window/matrix S.
Step 1: Compute covariance matrix $C^{(x,y)} = \frac{1}{W-1}[S^t S]$.
Step 2: Find eigenvalues of $C^{(x,y)}$.
Step 3: Calculate the value of each pixel V(x, y)= (strength h)+ (strength v).
Step 4: Compute the value of each image $V(I) = \sum_{i=1}^{n} V(x,y)$.

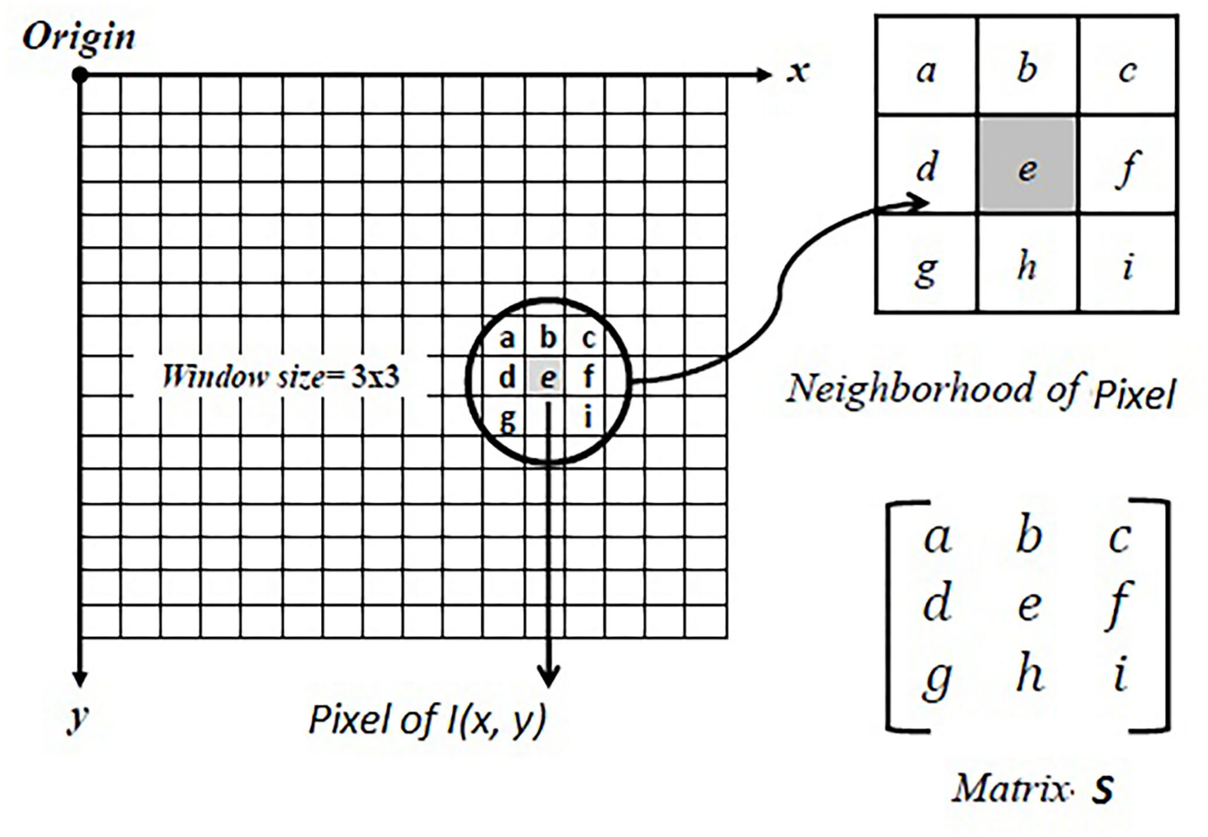

FIGURE 1
Generated window denoted as matrix S.

TABLE 2 Test problem and dimensions in different test sets.

Function Name and dimension		Starting points			
		[a]	[b]	[c]	[d]
Rosenbrock			Soft		
(2)	$e = 10^{-7}$	(-1.2, 1.0)	(-120, 100)	(20, -20)	(6.39, -0.221)
(20)	$e = 10^{-7}$	([-1.2, 1.0])	(1, 2,....,20)	([6.39, -0.221])	(-1, -1, -1, -1, -1, 1,....1)
			Medium		
(26)	$e = 10^{-7}$	([-1.2, 1.0])	([F])	([20])	([6.39, -0.221])
(40)	$e = 10^{-7}$	([-1.2, 1])	([-120, 100])	([1, -2, 3, -4,...., -10])	([20])
(60)	$e = 10^{-7}$	([-1.2, 1])	([F])	([F])	([6.39, -0.221])
			Hard		
(80)	$e= 10^{-7}$	([-1.2, -1.0])	([F])	([F])	([F])
(100)	$e=10^{-7}$	([-1.2, -1])	([F])	([F])*([F])	
(120)	$e=10^{-7}$	([-1.2, -1])	([20])	[F]	([6.39, -0.221])

TABLE 3 Test problem and dimensions in different test sets.

Function name and dimension		Starting points			
		[a]	[b]	[c]	[d]
Modified Trigonometric			**Soft**		
(16)	e = 10^{-7}	([−2, −1, 1, 2])	([−2, 1.5,.., −1.5, 2])	([0.1, 1, −0.1, −1])	([2.5, 2, 1.5, 1, 0.5], 2.5)
			Medium		
(32)	e = 10^{-7}	([−2, −1, 1, 2]	([−2, 1.5,.., −1.5, 2]	([0.1, 1, −0.1, 1]	([2.5, 2,..,0.5], 2.5, 2)
(64)	e = 10^{-6}	([−2, −1, 1, 2])	([−2, 1.5,..., −1.5, 2])	([0.1, 1, −0.1, 1])	([2.5, 2,.. 0.5], 2.5,.., 1)
			Hard		
(95)	e= 10^{-5}	([−2, −1, 1, 2],−2, −1, 1)	([−2, 1.5,.., 2])	([0.1, 1.0, −0.1, 1.0])	([2.5, 2.0, 1.5, 1.0, 0.5])
(128)	e= 10^{-6}	([−2, −1, 1, 2], −2, −1, 1)	([−2, 1.5,.., −1.5, 2])	([0.1, 1, −0.1, 1])	([2.5, 2, 1.5, 1, 0.5], 2.5, 2, 1.5)
(150)	e= 10^{-5}	([−2, −1, 1, 2], −2, −1)	([−2, 1.5, ..., 2], −2,..., −0.5, 1)	([0.1, 1, −0.1, 1], 0.1, 1)	([2.5, 2.0, 1.5, 1.0, 0.5])

A

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

B

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

FIGURE 2
Images of test functions. **(A)** I1; **(B)** I2.

TABLE 4 Image values of test functions.

Image	Test functions	Value
I1	Extended rosenbrock	0.1009
I2	Modified trigonometric	0.1004

TABLE 5 Results of rosenbrock function of all dimension problems in a two-step method of the first technique.

Method	Function Eval	Iteration	Time in sec	Failure	Dimension
${}^{[1]}_{[0]}A_1^{(2)}$	**60**	**43**	0.0243	0	
${}^{[1]}_{[0]}A_2^{(2)}$	62	45	0.0309	0	Soft
${}^{[1]}_{[0]}A_3^{(2)}$	62	45	**0.0205**	0	
${}^{[1]}_{[0]}A_1^{(2)}$	65	46	**0.0180**	0	
${}^{[1]}_{[0]}A_2^{(2)}$	**44**	46	0.0256	0	Medium
${}^{[1]}_{[0]}A_3^{(2)}$	65	**44**	0.0453	0	
${}^{[1]}_{[0]}A_1^{(2)}$	65	46	**0.0307**	0	
${}^{[1]}_{[0]}A_2^{(2)}$	**65**	**44**	0.0449	0	Hard
${}^{[1]}_{[0]}A_3^{(2)}$	65	44	0.0629	0	

The bold values indicate the good experimental results provided by the proposed methods in terms of function evaluation, number of iterations and time elapsed in seconds, which is one of our objectives in the study of this paper.

Step 5: Set threshold on V(I).

Step 6: Function execution by indicated/best method type.

TABLE 6 Results of rosenbrock function of all dimension problems in the two-step method of the second technique.

Method	Function Eval	Iteration	Time in sec	Failure	Dimension
${}^{[1]}_{[1]}A_1^{(2)}$	**62**	**30**	**0.0283**	0	
${}^{[1]}_{[1]}A_2^{(2)}$	62	34	0.0330	0	Soft
${}^{[1]}_{[1]}A_3^{(2)}$	62	31	0.0423	0	
${}^{[1]}_{[1]}A_1^{(2)}$	51	44	0.0230	0	
${}^{[1]}_{[1]}A_2^{(2)}$	**49**	**39**	**0.0158**	0	Medium
${}^{[1]}_{[1]}A_3^{(2)}$	49	39	0.0193	0	
${}^{[1]}_{[1]}A_1^{(2)}$	51	44	**0.0298**	0	
${}^{[1]}_{[1]}A_2^{(2)}$	**49**	**39**	0.0309	0	Hard
${}^{[1]}_{[1]}A_3^{(2)}$	49	39	0.0405	0	

The bold values indicate the good experimental results provided by the proposed methods in terms of function evaluation, number of iterations and time elapsed in seconds, which is one of our objectives in the study of this paper.

5. Numerical analysis of test functions

Two test functions, namely, Rosenbrock and modified trigonometric functions, are selected from the literature (Hillstrom et al., 1981) to evaluate their performance by using two different two-step techniques, i.e., one-step skipping with no modified search direction and one-step skipping with modified

TABLE 7 Results of modified trigonometric function of all dimension problems in the two-step method of the first technique.

Method	Function Eval	Iteration	Time in sec	Failure	Dimension
${}^{[1]}_{[0]}A_1^{(2)}$	**104**	**75**	**0.0312**	0	Soft
${}^{[1]}_{[0]}A_2^{(2)}$	144	95	0.0415	0	
${}^{[1]}_{[0]}A_3^{(2)}$	144	95	0.0590	0	
${}^{[1]}_{[0]}A_1^{(2)}$	170	111	**0.0500**	0	Medium
${}^{[1]}_{[0]}A_2^{(2)}$	**168**	**111**	0.0521	0	
${}^{[1]}_{[0]}A_3^{(2)}$	168	112	0.0705	0	
${}^{[1]}_{[0]}A_1^{(2)}$	199	127	**0.1747**	0	Hard
${}^{[1]}_{[0]}A_2^{(2)}$	193	**125**	0.2117	0	
${}^{[1]}_{[0]}A_3^{(2)}$	**175**	128	0.2160	0	

The bold values indicate the good experimental results provided by the proposed methods in terms of function evaluation, number of iterations and time elapsed in seconds, which is one of our objectives in the study of this paper.

TABLE 8 Results of modified trigonometric function of all dimension problems in the two-step method of the second technique.

Method	Function Eval	Iteration	Time in sec	Failure	Dimension
${}^{[1]}_{[1]}A_1^{(2)}$	110	109	**0.0400**	0	Soft
${}^{[1]}_{[1]}A_2^{(2)}$	**108**	**103**	0.0544	0	
${}^{[1]}_{[1]}A_3^{(2)}$	109	104	0.0408	0	
${}^{[1]}_{[1]}A_1^{(2)}$	137	132	0.0544	0	Medium
${}^{[1]}_{[1]}A_2^{(2)}$	**136**	**132**	**0.0516**	0	
${}^{[1]}_{[1]}A_3^{(2)}$	136	132	0.0568	0	
${}^{[1]}_{[1]}A_1^{(2)}$	**147**	146	**0.1808**	0	Hard
${}^{[1]}_{[1]}A_2^{(2)}$	147	145	0.1895	0	
${}^{[1]}_{[1]}A_3^{(2)}$	147	**145**	0.1869	0	

The bold values indicate the good experimental results provided by the proposed methods in terms of function evaluation, number of iterations and time elapsed in seconds, which is one of our objectives in the study of this paper.

search direction. These functions are of different dimensions ranging from 2 to 150.

5.1. Discussion on rosenbrock function

- It is evident from Table 5 that the function outperformed by the method $({}^{[1]}_{[0]}\mathbf{A}_1^{(2)})$ during function evaluation and number of iterations, and time reduction is noted by $({}^{[1]}_{[0]}\mathbf{A}_1^{(2)})$ in soft dimension. In medium dimension, the $({}^{[1]}_{[0]}\mathbf{A}_2^{(2)})$ and $({}^{[1]}_{[0]}\mathbf{A}_2^{(3)})$ methods outperformed in function evaluation, the number of iterations, respectively, while the $({}^{[1]}_{[0]}\mathbf{A}_1^{(2)})$ method exhibited reduction in time. In hard dimension, the $({}^{[1]}_{[0]}\mathbf{A}_2^{(2)})$ method showed better results to minimize the function evaluation and the number of iterations and the $({}^{[1]}_{[0]}\mathbf{A}_1^{(2)})$ method reduced the computational effort.
- The results of Table 6 show that the $({}^{[1]}_{[1]}\mathbf{A}_1^{(2)})$ method outperformed in terms of function evaluation, the number of iterations, and computational time in soft dimension problems. In medium dimension, the $({}^{[1]}_{[1]}\mathbf{A}_2^{(2)})$ method exhibited best results in function evaluation, the number of iterations, and computational time. The $({}^{[1]}_{[1]}\mathbf{A}_2^{(2)})$ method showed a reduction in function evaluation and the number of iterations, and the $({}^{[1]}_{[1]}\mathbf{A}_1^{(2)})$ method is executed in less time.

- **Comparative analysis of both techniques**

- The behavior of both techniques was compared, and our analysis concluded that one-step skipping with no modified search direction outperformed in function evaluation and computational time, while the second technique, i.e., one-step skipping with modified search direction, showed a reduction in the number of iterations in all dimension problems.

5.2. Discussion on modified trigonometric function

- Table 7 shows the function outperformed by the $({}^{[1]}_{[0]}\mathbf{A}_1^{(2)})$ method in function evaluation, the number of iterations, and in computational time in soft dimension. In medium dimension, the $({}^{[1]}_{[0]}\mathbf{A}_2^{(2)})$ method outperformed in function evaluation and the number of iteration while less computational time is noted by the $({}^{[1]}_{[0]}\mathbf{A}_1^{(2)})$ method. The $({}^{[1]}_{[0]}\mathbf{A}_3^{(2)})$ method showed better results in function evaluation, while the $({}^{[1]}_{[0]}A_2^{(2)})$ and $({}^{[1]}_{[0]}A_1^{(2)})$ methods outperformed in the number of iterations and time, respectively, in hard dimension.
- In Table 8, the $({}^{[1]}_{[1]}\mathbf{A}_2^{(2)})$ method showed the best results in function evaluation and the number of iterations in soft and medium dimensions, and the time reduction is observed by the $({}^{[1]}_{[1]}\mathbf{A}_1^{(2)})$ method in soft dimension and by the $({}^{[1]}_{[1]}\mathbf{A}_2^{(2)})$ method in medium dimension. In hard dimension, the $({}^{[1]}_{[1]}\mathbf{A}_1^{(2)})$ method exhibited minimum results in function evaluation and computational time, and the $({}^{[1]}_{[1]}\mathbf{A}_3^{(2)})$ method showed a reduction in the number of iterations.

- **Comparative analysis of both techniques**

- Both techniques were compared and analyzed based on experimental results. From the analysis, it can be concluded that one-step skipping with no modified search direction outperformed in function evaluation, the number of iterations, and computational time, except the one case of medium dimension, in which the second technique, i.e., one-step skipping with modified search direction, showed a reduction in function evaluation.

6. Conclusion

An algorithm was developed to compute the image value of a particular test function and direct it to its best method for execution. The two-step quasi-Newton methods with two techniques (one-step skipping with no modified search direction and one-step skipping with modified search direction) were chosen and experimented on two test functions, namely, Rosenbrock and modified trigonometric function. The best method was determined using the experimental results obtained in terms of function evaluation, the number of iterations, and computational time. This study concluded that the one-step skipping without modification in search direction technique showed superiority over the one-step skipping with modified search direction technique under both test functions. Hence, this algorithm directed all the functions having the same image value as Rosenbrock and modified trigonometric functions to the one-step skipping technique with no modified search direction.

7. Future work

To further strengthen the algorithm reported in this study, we propose to investigate the image recognition in terms of picture or graph instead of image value and then direct the reported function (or medical image) to the best method available for the obtaining solution. Based on the literary research, in the future, we are planning to collaborate with some biomedical labs to validate the practicality of the proposed algorithm.

Data availability statement

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

Author contributions

FJ, NA, and SA-m performed the main concept and experimental work. WM and MA were made critical revisions, reviewed, help in writing, analysis of this paper, and approved the final version. All authors contributed to the article and approved the submitted version.

Acknowledgments

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University for funding this work through Large Groups RGP.2/212/1443. WM was thankful to the Directorate of ORIC, Kohat University of Science and Technology for awarding the Project titled-Advanced Soft Computing Methods for Large-Scale Global Optimization Problems.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Aamir, N., and Ford, J. (2021). Two-step skipping techniques for solution of nonlinear unconstrained optimization problems. *Int. J. Eng. Works* 8, 170–174. doi: 10.34259/ijew.21.806170174

Christian, C. T., Sutton, M. A., and Bezdek, J. C. (2000). "Breast cancer detection using image processing techniques," in *Ninth IEEE International Conference on Fuzzy Systems. FUZZ-IEEE 2000 (Cat. No. 00CH37063), Vol. 2* (Vellore: IEEE), 973–976.

Ford, J., and Moghrabi, I. (1993). Alternative parameter choices for multi-step quasi-Newton methods. *Optim. Meth. Softw.* 2, 357–370. doi: 10.1080/10556789308805550

Ford, J., and Moghrabi, I. (1994). Multi-step quasi-Newton methods for optimization. *J. Comput. Appl. Math.* 50, 305–323. doi: 10.1016/0377-0427(94)90309-3

Hillstrom, K., More, J., and Garbow, B. (1981). Testing unconstrained optimization software. *ACM Trans. Math. Softw.* 7, 136–140. doi: 10.1145/355934.355943

Jaffar, F., and Aamir, N. (2020). Comparative analysis of single/multi-step quasi-newton methods at different delta values. *Punjab Univ. J. Math.* 52, 65–77.

Mannel, F., and Rund, A. (2021). A hybrid semismooth quasi-newton method for nonsmooth optimal control with pdes. *Optim. Eng.* 22, 2087U2125. doi: 10.1007/s11081-020-09523-w

Moghrabi, I. A., Hassan, B. A., and Askar, A. (2022). New self-scaling quasi-newton methods for unconstrained optimization. *Int. J. Math. Comput. Sci.* 17, 1061U1077. doi: 10.11591/ijeecs.v21.i3.pp1830-1836

Narmadha, R. P., and Arulvadivu, G. (2017). "Detection and measurement of paddy leaf disease symptoms using image processing," in *2017 International Conference on Computer Communication and Informatics (ICCCI)* (Coimbatore: IEEE), 1–4.

Navid, A., Tehranipoor, M., and Forte, D. (2017). Counterfeit electronics detection using image processing and machine learning. *J. Phys. Conf. Ser.* 787, 012023. doi: 10.1088/1742-6596/787/1/012023

Ritter, F., Laue, T. B. A. H. H., Schwier, M., Link, F., and Peitgen, H.-O. (2011). Medical image analysis. *IEEE Pulse* 2, 60–70. doi: 10.1109/MPUL.2011.942929

Schröter, M., and Sauer, O. A. (2010). Quasi-newton algorithms for medical image registration. *IFMBE Proc.* 25, 433–436. doi: 10.1007/978-3-642-03882-2_115

Shao, C., Zheng, S., Gu, C., Hu, Y., and Qin, X. (2022). A novel outlier detection method for monitoring data in dam engineering. *Expert Syst. Appl.* 193, 116476. doi: 10.1016/j.eswa.2021.116476

Tamara, D. P. O., Kolda, G., and Nazareth, L. (1998). BFGS with update skipping and varying memory. *J. Optim* 8, 1060–1083. doi: 10.1137/S1052623496306450

Wang, J. (2018). Application of improved quasi-newton method to the massive image denoising. *Multimed Tools Appl.* 77, 12157–12170. doi: 10.1007/s11042-017-4863-y

Wolf, I., Vetter, M., Wegner, I., Bottger, T., Nolden, M., Schobinger, M., et al. (2005). The medical imaging interaction toolkit. *Med. Image Anal.* 9, 594–604. doi: 10.1016/j.media.2005.04.005