DOI: 10.24996/ijs.2024.65.3.25

ISSN: 0067-2904

# A Modified Davidon-Fletcher-Powell Method for Solving Nonlinear Optimization Problems

**Ali Joma'a Al-Issa[1], Basim A. Hassan[2], Issam A R Moghrabi[3]*, Ibrahim M. Sulaiman[4]**

*[1]Department of Mathematics, College of Computer Sciences and Mathematics University of Aleppo, Aleppo*
*[2]Department of Mathematics, College of Computers Sciences and Mathematics, University of Mosul, IRAQ*
*[3]Department of Computer Science, College of Arts and Sciences, University Central Asia, 310 Lenin Street, Naryn, 722918, Kyrgyz Republic*
*[4]Institute of Strategic Industrial Decision Modelling (ISIDM), School of Quantitative Sciences, Universiti Utara Malaysia, Sintok, 06010, Kedah, Malaysia,*

Received: 22/11/2022 Accepted: 26/3/2023 Published: 30/3/2024

**Abstract**

One of the quasi-Newton update formulae, namely the Davidon-Fletcher-Powell method, is crucial for resolving nonlinear programming optimization problems. In order to achieve a Newton-like condition that depends on the function values and gradient vectors at each iteration, we construct an alternative positive-definite Hessian approximation in this study. The essential theorems are established to study algorithm convergence. The proposed approach is then tested on well-known test problems and then compared to the standard DFP method. The numerical outcomes demonstrate the effectiveness of the newly developed method.

**Keywords:** Quasi-Newton Methods, Nonlinear optimization, Unconstrained optimization, DFP update.

## 1. Introduction

Nonlinear programming is an important method since so many objects in our environment do not act linearly. A drug's efficacy needs not double just because the dosage is doubled. A project may not be finished twice as quickly because twice as many individuals are working on it. In research and engineering, nonlinear models are often used. Nonlinear models may also be used in commercial applications, notwithstanding their rarity, such as the administration of investment portfolios. In this situation, the objective may be to choose an investment mix that maximizes return while minimizing risk. The model's nonlinearity results from the consideration of risk.

The most known class of problems in optimization takes the form:

$$\min\ f(x), x \in \mathbb{R}^n, \tag{1}$$

where $f(x)$ is twice continuously differentiable. To solve the unconstrained optimization problem (1), several methods have been proposed such as Newton's methods [1]–[3], quasi-Newton methods [4], [5], trust-region methods [6], [7], and conjugate gradient methods [8], [9]. One of Newton's method [1] benefits is that it can only discover a solution rapidly and correctly if the function $f(x)$ is quadratic. However, if the function is not quadratic, the method has flaws as it may fail, especially if the Hessian $H(x)$ of the objective function being minimized

*Email: i.mohammed.sulaiman@uum.edu.my

is not positive definite or if the starting point $x_0$ is not close to the solution to a sufficient degree. Newton's method computational cost dramatically increases with the increasing number of variables of the function $f(x)$. To eliminate the difficulties mentioned above, quasi-Newton methods are a better option. [4]–[6], [10], [11] and such methods differ from each other through the alternative update formulas developed to approximate the Hessian matrix (or its inverse). These methods compute successive iterates using the following formula:

$$x_{k+1} = x_k + s_k, \quad s_k = \alpha_k d_k, \qquad k \geq 0, \tag{2}$$

where a step size $\alpha_k > 0$ is exactly defined as [1]:

$$\alpha_k = -\frac{g_k^T d_k}{d_k^T \mathrm{Q}\, d_k} \tag{3}$$

for quadratic functions with a positive definite matrix $Q$, $g_k = g(x_k)$ is the gradient of $f$ (x) at $x_k$, and $d_k$ is the search direction computed by quasi- Newton methods using [7]–[9]:

$$B_k d_k = -g_k, \quad k \geq 0\ , \tag{4}$$

where $B_k$ approximates the Hessian matrix $H_k$. For non-quadratic functions, $\alpha_k$ is computed using some line search algorithm that satisfies certain criteria to be stated later. Quasi-Newton updates generally satisfy the relation:

$$B_{k+1} s_k = y_k, \tag{5}$$

for $y_k = g_{k+1} - g_k$ and $s_k$ as in (2). One of the early updating formulae is due to Davidon, Fletcher and Powell (DFP) [12] and is given by:

$$B_{k+1}^{DFP} = \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) B_k \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) + \frac{y_k y_k^T}{s_k^T y_k}.$$

A crucial part of employing the quasi-Newton approach is the update formula, which creates a matrix $B_{k+1}$ from the existing matrix $B_k$ using the data available at the *k*th iteration. The numerical behavior of the approaches is influenced by this formula. It is intended that the matrices that are subsequently constructed would come as near as they can to the real Hessian. In the literature, there are a number of well-known updates of B k that meet the traditional Secant equation (1), but numerically, none of them have been able to match the well-known Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula which is described as follows [11], [13], [14]:

$$B_{k+1}^{BFGS} = B_k + \frac{y_k\, y_k^T}{y_k^T s_k} - \frac{B_k\, s_k s_k^T B_k^T}{s_k^T B_k s_k}.$$

The BFGS update formula is the most successful of all the quasi-Newton approaches, according to reported numerical results. The method converges finitely in a maximum of *n* iterations on quadratic functions if and only if $y_k^T s_k > 0$, for $y_k$ and $s_k$ as in (5). Convergence is still a problem for generic functions even though global and superlinear convergence is demonstrated for convex functions [9], [13]. Demonstrate that the conventional BFGS method might not converge on generic functions even with precise line search. When dealing with non-convex functions, the standard BFGS approach may fail when the accurate line search is being employed as illustrated in [14] with the aid of an example.

In many optimization algorithms, scholars often use the Wolfe-Powell (WP) line search technique to find step length, see [1], [15]. The Wolfe-Powell (WP) line search technique is determined by:

$$f(x_k+\alpha_k d_k) \le f(x_k) + \delta\alpha_k g_k^T d_k,$$
$$g(x_k+\alpha_k d_k)^T d_k \le \sigma g_k^T d_k \ , \tag{6}$$

where $0<\delta<\sigma<1$. The above discussion motivates us to seek an improved optimization algorithm that may obtain better numerical performance.

In this paper, we suggest a new quasi-Newton method which is a variant of the DFP formula. This is derived in Section 2. The convergence analysis of the method follows in Section 3. Numerical test results are presented in section 4. The conclusions follow in Section 5.

## 2. Development of a New Matrix with an Alternative Update to the Hessian Matrix

Zengxin et al. [16] presented a variant update of the BFGS formula as follows:

$$B_{k+1} = B_k + \frac{y_k^{m*}\, y_k^{m*T}}{s_k^T y_k^{m*}} - \frac{B_k\, s_k s_k^T B_k^T}{s_k^T B_k s_k},$$
$$\mathrm{y}_\mathrm{k}^{\mathrm{m}*} = y_k + \frac{\rho_k}{\|s_k\|^2}.s_k, \quad \rho_k = 2(f_k - f_{k+1}) + (g_k + g_{k+1})^T s_k \ . \tag{7}$$

Since the matrix $B_{k+1}$ approximates the true Hessian matrix $H_{k+1}$ for which the following holds:

$$s_k^T H_{k+1} s_k = y_k^T s_k + 2(f_k - f_{k+1}) + (g_k + g_{k+1})^T s_k, \tag{8}$$
$$\frac{1}{n} s_k^T H_{k+1} s_k = \frac{1}{n} y_k^T s_k + \frac{2}{n}(f_k - f_{k+1}) + \frac{1}{n}(g_k + g_{k+1})^T s_k, \quad n>0, \tag{9}$$

and

$$s_k^T H_{k+1} s_k = \frac{1}{n} y_k^T s_k + \frac{2}{n}(f_k - f_{k+1}) + \frac{1}{n}(g_k + g_{k+1})^T s_k + \left(\frac{n-1}{n}\right) s_k^T H_{k+1} s_k. \tag{10A}$$

Substituting (3) and (10) in (9), we get:

$$s_k^T H_{k+1} s_k = \frac{1}{n} y_k^T s_k + \frac{2}{n}(f_k - f_{k+1}) + \frac{1}{n} g_{k+1}^T s_k + \left(\frac{2-n}{n}\right) g_k^T s_k. \tag{10B}$$

Since $B_{k+1}$ approximates the Hessian matrix $H_{k+1}$, (11) can be replaced with:

$$s_k^T B_{k+1} s_k = \frac{1}{n} y_k^T s_k + \frac{2}{n}(f_k - f_{k+1}) + \frac{1}{n} g_{k+1}^T s_k + \left(\frac{2-n}{n}\right) g_k^T s_k.$$

According to the quasi- Newton condition (5), the following formula then follows from the above expression for $s_k^T B_{k+1} s_k$:

$$A_k = \frac{1}{n} y_k + \frac{\frac{2}{n}(f_k - f_{k+1}) + \frac{1}{n} g_{k+1}^T s_k + \left(\frac{2-n}{n}\right) g_k^T s_k}{s_k^T y_k} y_k, \quad s_k^T y_k \ne 0\ . \tag{11}$$

We use the proposed formula $A_k$ to develop the following DFP-variant and thus we get a new update matrix as follows:

$$M_{k+1}^{HASDFP} = \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) M_k \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) + \frac{y_k y_k^T}{s_k^T A_k} \ . \tag{12}$$

Based on the above discussion, we describe the algorithmic outline of the proposed method as follows:

**The IHMSDFP algorithm**
Step 0: $x_0 \in \mathbb{R}^n$, $M_0 = I_{n*n}$, $\varepsilon > 0$, set $k = 0$.
Step 1: if $\|g(x_k)\| \leq \varepsilon$ then stop, otherwise go to step2.
Step 2: Compute $d_k M_k = -g_k$.
Step 3: Compute the step size $\alpha_k$ along direction $d_k$ so that the Wolf-Powell conditions (6) are satisfied.
Step 4: Let $x_{k+1} = x_k + \alpha_k d_k$, if $\|g_{k+1}\| \leq \varepsilon$ then stop.
Step 5: set $s_k = x_{k+1} - x_k$, $y_k = g_{k+1} - g_k$.
Step 6: find $A_k$ from (11).
Step 7: update $M_{k+1}$ by the formula(12).
Step 8: set $k = k+1$, go to step2.

**Theorem 2.1**.
The IHMSDFP method presented with the search direction and the step length satisfies the Wolf-Powell conditions (6), then $M_{k+1}^{HASDFP}$ generates positive definite updates, for all $k \geq 0$, provided the following condition is fulfilled:

$$s_k^T A_k > 0 \quad ,\forall n, k > 0 . \tag{13}$$

**Proof:**

$$s_k^T A_k = \frac{1}{n} y_k^T s_k + \frac{2}{n}(f_k - f_{k+1}) + \frac{1}{n} g_{k+1}^T s_k + \left(\frac{2-n}{n}\right) g_k^T s_k \tag{14}$$

$$= \frac{1}{n}(g_{k+1} - g_k)^T s_k + \frac{2}{n}(f_k - f_{k+1}) + \frac{1}{n} g_{k+1}^T s_k + \left(\frac{2-n}{n}\right) g_k^T s_k ,$$

which yields:

$$s_k^T A_k = \frac{2}{n}(f_k - f_{k+1}) + \frac{2}{n} g_{k+1}^T s_k + \left(\frac{1-n}{n}\right) g_k^T s_k. \tag{15}$$

From (12) and (13), it follows that $f_k - f_{k+1} \geq -\delta \alpha_k g_k^T d_k = -\delta g_k^T s_k$, when substituted in (15) yields:

$$s_k^T A_k \geq -\frac{2}{n}\delta g_k^T s_k + \frac{2}{n}\sigma g_k^T s_k + \left(\frac{1-n}{n}\right) g_k^T s_k = c g_k^T s_k \tag{16}$$

where $c = \frac{-2\delta + 2\sigma + 1 - n}{n} < 0$. But we also have:
$s_k^T A_k \geq c g_k^T s_k > 0$.
Hence, the proof is complete.

In the next section, we show that our method is globally convergent even without convexity assumptions on the objective function.

## 3. CONVERGENCE ANALYSIS

Throughout this section, we assume that the gradient is not equal to zero for all $k \geq 1$, otherwise a stationary point is found. In order to proceed with the convergence analysis, the following basic assumptions on the objective function are considered.

**Assumptions**

i. The level set $\Psi = \{x \in \mathbb{R}^n | \, f(x) \leq f(x_0)\}$ is bounded.

ii. The gradient vector satisfies the Lipchitz condition [2] and is continuous on the neighborhood $N$ of the group $\Psi$, and there is a constant $L > 0$ such that:
$\|\nabla f(x) - \nabla f(y)\| \leq L\|x - y\|$ for all $x$ and $y$ from $N$, there exist constants $u, b \geq 0$ such that $\|g_k\| \leq u$, for all $x$, and $\|x\| \leq b$. Since $\{f_k\}$ is a declining series, it is clear that the series $\{x_k\}$ generated by a new Algorithm is found in $\Psi$, and there exists a constant $f^*$ such that:

$$\lim_{k\to\infty} f_k = f^* \tag{17}$$

See [17], [18].

**Theorem 3.1.**

Let $\{x_k\}$ be generated by the new method, and the following inequality holds:

$$\|B_k s_k\| \leq m_1\|s_k\|,\ \ s_k^T B_k s_k \geq m_2\|s_k\|^2, \quad m_1 > 0\ ,\ m_2 > 0\ , \tag{18}$$

then the following holds

$$\lim_{k\to\infty} \inf\|g_k\| = 0. \tag{19}$$

**Proof:**

Since $B_k d_k = -g_k$, we have:

$$\|B_k d_k\| \leq m_1\|d_k\|,\ \ d_k^T B_k d_k \geq m_2\|d_k\|^2, \tag{20}$$

By using (6) and assumption (ii), we have:

$$-(1-\sigma)g_k^T d_k \leq (g_{k+1} - g_k)^T d_k \leq L\alpha_k\|d_k\|^2, \tag{21}$$

This implies that:

$$\alpha_k \geq \frac{-(1-\sigma)g_k^T d_k}{L\|d_k\|^2} = \frac{(1-\sigma)d_k^T B_k d_k}{L\|d_k\|^2} \geq \frac{(1-\sigma)\ m}{L} = \tilde{\alpha}_k \tag{22}$$

on the other hand, from (17), we obtain:

$$\sum_{k=1}^{\infty}(f_k - f_{k+1}) = \lim_{N\to\infty}\sum_{k=1}^{\infty}(f_k - f_{k+1}) = \lim_{N\to\infty}(f_1 - f_{k+1}) = f_1 - f^*. \tag{23}$$

which yields:

$$\sum_{k=1}^{\infty}(f_k - f_{k+1}) \leq +\infty, \tag{24}$$

Using (16), we get:

$$\sum_{k=1}^{\infty} -\alpha_k g_k^T d_k \leq +\infty \tag{25}$$

which ensure:

$$\lim_{k\to\infty} -\alpha_k g_k^T d_k = 0 \tag{26}$$

this together with (22) leads to:

$$\lim_{k\to\infty} d_k^T B_k d_k = \lim_{k\to\infty} - g_k^T d_k = 0 \tag{27}$$

Which along with (20) yields (19).

## 4. Numerical results and discussions

The numerical experimental findings on 48 test functions discussed in [18] are provided in this section. We compare and contrast the two approaches, DFP and IHMSDFP. The stopping condition is set to $\|g_k\| \leq 10^{-6}$ for all methods. A list of the test functions is presented in Table 1. All the problems are solved utilizing MATLAB on a Intel (R) Core TM i3-4005U (1.70GHz) CPU, with 4 GB RAM. In some cases, the computation was terminated due to the failure of the line search to find an acceptable step size and thus those were considered failures. Numerical results are compared in terms of the number of evaluations of the function (NF) and the number of iterations. Figures 1 and 2 are performance profiles created with a performance profile tool developed by Dolan and More (see [19]–[20]). Table 2 reports the numerical results.

**Table 1:** A list of test functions

| P.NO | FUNCTION | $x_0$ |
|---|---|---|
| 1 | Extended Trigonometric | [0.2,0.2] |
| 2 | Extended Rosenbrock | [-1.2,1] |
| 3 | Extended White & Holst | [-1.2,1] |
| 4 | Extended Beale | [1,0.8] |
| 6 | Extended Penalty | [1,2] |
| 7 | Perturbed Quadratic | [0.5,0.5] |
| 8 | Raydan 1 | [1,1] |
| 9 | Raydan 2 | [1,1] |
| 10 | Diagonal 2 | [1,0.5] |
| 11 | Diagonal 3 | [1,1] |
| 12 | Hager | [1,1] |
| 13 | Generalized Tridiagonal 1 | [2,2] |
| 14 | Extended TET | [0.1,0.1] |
| 15 | Diagonal 4 | [1,1] |
| 16 | Diagonal 5 | [1.1,1.1] |
| 17 | Extended Himmelblau | [1,1] |
| 18 | Generalized PSC1 | [3,0.1] |
| 19 | Extended PSC1 | [3,0.1] |
| 20 | Full Hessian FH1 | [0.01,0.01] |
| 21 | Full Hessian FH2 | [0.01,0.01] |
| 22 | Extended BD1 | [0.1,0.1] |
| 23 | Extended Cliff | [0,−1] |
| 24 | Perturbed quadratic diagonal | [0.5,0.5 |
| 25 | Extended Wood | [−3,−1,−3,−1] |
| 26 | Quadratic QF1 | [1,1] |
| 27 | Extended quadratic penalty QP1 | [1,1] |
| 28 | Quadratic QF2 | [0.5,0.5] |
| 29 | Extended Tridiagonal 2 | [1,1] |
| 30 | FLETCHCR | [0,0] |
| 31 | ARGLINB | [1,1] |
| 32 | Partial Perturbed Quadratic | [0.5,0.5] |
| 33 | Broyden Tridiagonal | [−1,−1] |
| 34 | Almost Perturbed Quadratic | [0.5,0.5] |
| 35 | Perturbed Tridiagonal Quadratic | [0.5,0.5] |
| 36 | Staircase 1 | [1,1] |
| 37 | Staircase 2 | [4,4] |
| 38 | LIARWHD | [4,4] |
| 39 | POWER | [1,1] |
| 40 | ENGVAL1 | [2,2] |
| 41 | EDENSCH | [0,0] |
| 42 | CUBE | [−1.2,1] |
| 43 | NONSCOMP | [3,3] |
| 44 | QUARTC | [2,2] |
| 45 | SINQUAD | [0.1,0.1] |
| 46 | COSINE | [1,1] |
| 47 | SINE | [1,1] |
| 48 | Generalized Quartic | [1,1] |

**Table 2:** test results

| P.No | DFP | | | | IHMSDFP | | | |
|---|---|---|---|---|---|---|---|---|
| | **IT** | **NF** | **f** | **norm** | **IT** | **NF** | **f** | **norm** |
| P1 | 8 | 33 | 2.65428e-16 | 4.72e-08 | 7 | 24 | 1.3093e-24 | 1.3125e-12 |
| P2 | F | F | F | F | 21 | 66 | 2.5091e-19 | 4.8622e-09 |
| P3 | F | F | F | F | 14 | 45 | 1.0356e-20 | 5.5971e-09 |
| P4 | 24 | 78 | 3.15035e-13 | 5.71e-06 | 9 | 30 | 3.4371e-18 | 1.8316e-08 |
| P5 | 28 | 87 | 0.14722 | 1.55e-05 | 7 | 24 | 0.1472 | 0 |
| P6 | 6 | 21 | 2.59322e-14 | 3.1e-07 | 4 | 15 | 8.2874e-25 | 3.6666e-13 |
| P7 | 12 | 39 | 0.3 | 5.81e-07 | 4 | 15 | 0.3 | 0 |
| P8 | 2 | 6 | 2 | 0 | 2 | 6 | 2 | 0 |
| P9 | 9 | 30 | 1.84657 | 1.15e-06 | 4 | 15 | 1.8466 | 0 |
| P10 | 9 | 30 | 1.68773 | 4.47e-08 | 4 | 15 | 1.6877 | 0 |
| P11 | 6 | 21 | 1.92408 | 4.47e-08 | 4 | 15 | 1.9241 | 0 |
| P12 | 24 | 75 | 2.02173e-12 | 2.74e-06 | 12 | 39 | 1.6560e-14 | 3.3263e-07 |
| P13 | 13 | 45 | 2.55927 | 6.26e-07 | 4 | 15 | 2.5593 | 0 |
| P14 | 5 | 21 | 4.51506e-10 | 4.57e-05 | 3 | 12 | 7.8236e-22 | 4.4551e-10 |
| P15 | 5 | 18 | 1.38629 | 2.98e-08 | 2 | 6 | 1.3863 | 0 |
| P16 | F | F | F | F | 6 | 21 | 2.4966e-15 | 6.3672e-07 |
| P17 | 47 | 144 | 1 | 0.000105 | 4 | 15 | 1 | 0 |
| P18 | 37 | 114 | 0.773199 | 2.38e-05 | 5 | 18 | 0.7732 | 0 |
| P19 | 13 | 51 | 6.56476e-10 | 9.57e-07 | 16 | 51 | 1.0291e-13 | 7.5226e-07 |
| P20 | 11 | 36 | 1.78114e-11 | 9.39e-06 | 4 | 15 | 1.2836e-19 | 1.1110e-09 |
| P21 | 5 | 27 | 2.35923e-16 | 1.83e-09 | 2 | 9 | 0 | 5.6579e-12 |
| P22 | F | F | F | F | F | F | F | F |
| P23 | 6 | 30 | 1.38121e-15 | 2.36e-08 | 3 | 12 | 1.5463e-19 | 5.4698e-10 |
| P24 | F | F | F | F | 26 | 81 | 1.2454e-16 | 3.7932e-07 |
| P25 | 6 | 21 | -0.25 | 6.93e-07 | 5 | 18 | -0.25 | 0 |
| P26 | 7 | 27 | 1.1250 | 4.72e-06 | 6 | 21 | 1.1250 | 0 |
| P27 | 7 | 30 | -1.0562 | 7.67e-07 | 4 | 15 | -1.0562 | 0 |
| P28 | 3 | 15 | 0.3897 | 1.86e-08 | 2 | 6 | 0.3897 | 0 |
| P29 | 4 | 18 | 2.06283e-11 | 0.000171 | 3 | 12 | 6.4794e-24 | 2.8467e-10 |
| P30 | 12 | 42 | 2.77382e-07 | 9.67e-05 | 2 | 6 | 2.7079e-10 | 5.7528e-07 |
| P31 | 3 | 15 | 5.99569e-15 | 2.48e-07 | 4 | 15 | 3.0643e-21 | 1.5662e-10 |
| P32 | 11 | 36 | 6.5969e-13 | 6.87e-06 | 5 | 18 | 3.6895e-23 | 2.9279e-11 |
| P33 | 6 | 21 | 2.59322e-14 | 3.1e-07 | 4 | 15 | 8.2874e-25 | 3.6666e-13 |
| P34 | 2 | 9 | 0 | 1.49e-08 | 2 | 9 | 0 | 1.0054e-12 |
| P35 | 9 | 30 | 9.17158e-13 | 2.72e-06 | 4 | 15 | 3.0161e-19 | 1.7749e-09 |
| P36 | 6 | 21 | 2.53319e-16 | 1.62e-08 | 4 | 15 | 2.2232e-21 | 1.1363e-10 |
| P37 | 26 | 81 | 1.40912e-09 | 0.000258 | 12 | 39 | 1.7510e-23 | 3.4298e-11 |
| P38 | 7 | 24 | 1.69318e-13 | 1.53e-06 | 4 | 15 | 1.8782e-22 | 5.0504e-11 |
| P39 | 20 | 63 | 2.05565e-10 | 6.24e-05 | 6 | 21 | 5.5604e-11 | 0 |
| P40 | 18 | 57 | 16 | 3.58e-06 | 3 | 12 | 16 | 0 |
| P41 | F | F | | | 14 | 45 | 1.0356e-20 | 5.5971e-09 |
| P42 | 6 | 21 | 2.95023e-16 | 1.13e-06 | 3 | 12 | 3.8181e-28 | 5.1030e-11 |
| P43 | 2 | 6 | 0 | 0 | 2 | 6 | 1.2492e-16 | 3.9745e-12 |
| P44 | 39 | 138 | 1.20636e-09 | 3.01e-06 | 13 | 42 | 3.0856e-15 | 3.1180e-07 |
| P45 | 5 | 24 | -1 | 2.24e-08 | 2 | 9 | -1 | 0 |
| P46 | 5 | 24 | -1 | 2.24e-08 | 2 | 9 | -1 | 0 |
| P47 | 10 | 33 | 5.75437e-12 | 3.88e-06 | 6 | 21 | 9.6051e-22 | 6.2954e-11 |
| P48 | 37 | 114 | 0.773199 | 2.38e-05 | 5 | 18 | 0.7732 | 0 |

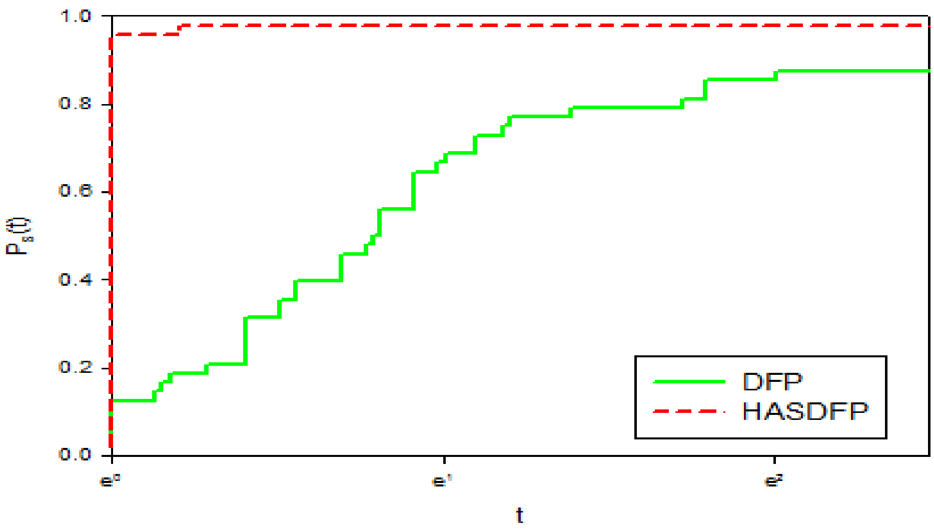


**Figure 1:** Performance profile relative to the number of iterations for DFP, IHMSDFP

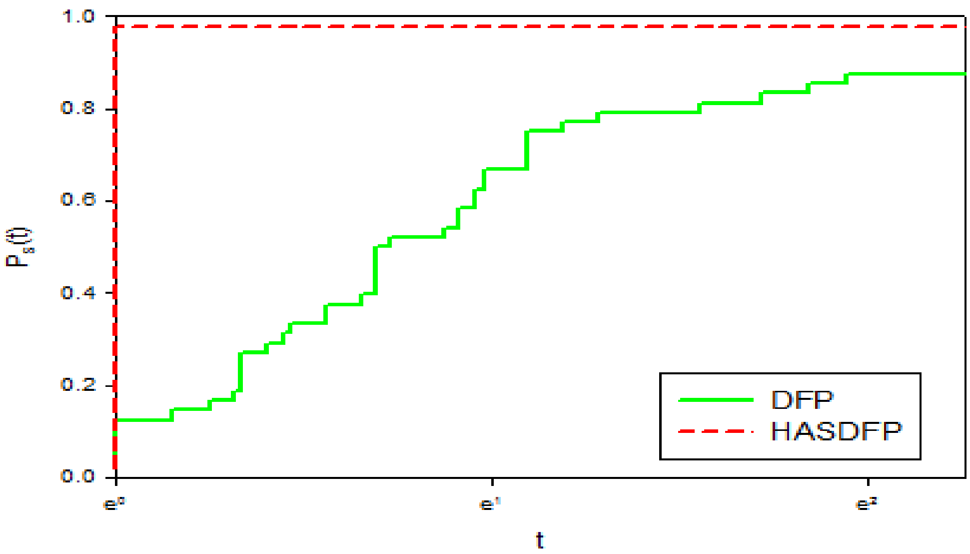


**Figure 2:** Performance profile relative to number of evaluations of the function for DFP, IHMSDFP

Figures 1 and 2 clearly show that the IHMSDFP approach is superior to the standard DFP method. It is more efficient in terms of the number of iterations and number of evaluations where our method solved 98 % of the tested problems, in comparison with the 87% rate for the DFP formula.

## 5. Conclusions

In this study, we modify quasi-Newtonian techniques by updating DFP and proposing a new approach called HASDFP, in which the new matrices used to approximate the Hessian matrix are always positive-defined and satisfied the Newton-like requirement. Moreover, a key feature of our proposed approaches is global convergence with inexact line search. For a set of unconstrained optimization problems, numerical comparisons are done between our suggested approach and the DFP method. The computational tests reveal that our novel approach is both efficient and effective, outperforming the HASDFP. As a potential future project, using search direction vector evolution, the hybridization of conjugate gradient techniques with Hessian matrix approximations HASDFP is worth investigating. It is also worth examining the impact of other approaches for determining the step length on the convergence of our suggested method.

**References**

**[1]** J. Nocedal and S. J. Wright, “Numerical optimization,” in Springer Series in Operations Research and Financial Engineering," *Springer Science & Business Media,* pp. 1–664,2006.

**[2]** B. A. Hassan, F. Alfarag, A. Ibrahim, and A. Abubakar, “An improved quasi-Newton equation on the quasi-Newton methods for unconstrained optimizations,” *Indones. J. Electr. Eng. Comput.* Sci., vol. 22, pp. 389–397, doi: 10.11591/ijeecs.v22.i2..

**[3]** B. A. Hassan and M. A. Kahya, “A new class of quasi-Newton updating formulas for unconstrained optimization,” J. Interdiscip. Math., vol. 24, no. 8, 2021, doi: 10.1080/09720502.2021.1961980.

**[4]** N. Andrei, “A diagonal quasi-Newton updating method for unconstrained optimization,” *Numer. Algorithms,* vol. 81, no. 2, 2019, doi: 10.1007/s11075-018-0562-7.

**[5]** A. S. Berahas and M. Takáč, “A robust multi-batch L-BFGS method for machine learning*,” *Optim. Methods Softw.,* vol. 35, no. 1, 2020, doi: 10.1080/10556788.2019.1658107.

**[6]** S. Rezaee and S. Babaie-Kafaki, “An adaptive nonmonotone trust region algorithm,” *Optim. Methods Softw.,* vol. 34, no. 2, 2019, doi: 10.1080/10556788.2017.1364738.

**[7]** Z. Sheng, G. Yuan, and Z. Cui, “A new adaptive trust region algorithm for optimization problems,” *Acta Math. Sci.,* vol. 38, no. 2, 2018, doi: 10.1016/S0252-9602(18)30762-8.

**[8]** Z. Dai and H. Zhu, “A modified hestenes-stiefel-type derivative-free method for large-scale nonlinear monotone equations,” Mathematics, vol. 8, no. 2, 2020, doi: 10.3390/math8020168.

**[9]** S. S. Mahmood and J. H. Eidi, “New Class of Rank 1 Update for Solving Unconstrained Optimization Problem,” *Iraqi J. Sci.,* vol. 63, no. 2, 2022, doi: 10.24996/ijs.2022.63.2.25.

**[10]** B. A. Hassan, K. Muangchoo, F. Alfara, A. H. Ibrahim, and A. B. Abubakar, “An improved quasi-Newton equation on the quasi-Newton methods for unconstrained optimizations,” *Indones. J. Electr. Eng. Comput.* Sci., vol. 22, no. 2, pp. 389–397, 2020, doi: 10.11591/ijeecs.v22.i2.pp389-397.

**[11]** R. Fletcher, “New approach to variable metric algorithms,” Comput. J., vol. 13, no. 3, 1970, doi: 10.1093/comjnl/13.3.317.

**[12]** W. C. Davidon, “Variable metric method for minimization,” *SIAM Journal on Optimization,* vol. 1, no. 1. 1991, doi: 10.1137/0801001.

**[13]** D. Goldfarb, “A family of variable metric updates derived by variational means. Mathematics of Computing,” 1970.

**[14]** D. F. Shanno, “Conditioning of quasi-Newton methods for function minimization,” *Math. Comput.,* vol. 24, no. 111, 1970, doi: 10.1090/s0025-5718-1970-0274029-x.

**[15]** P. Wolfe, “Convergence Conditions for Ascent Methods. II: Some Corrections,” *SIAM Rev.,* vol. 13, no. 2, 1971, doi: 10.1137/1013035.

**[16]** Z. Wei, G. Yu, G. Yuan, and Z. Lian, “The superlinear convergence of a modified BFGS-type method for unconstrained optimization,” *Comput. Optim. Appl.,* vol. 29, no. 3, 2004, doi: 10.1023/B:COAP.0000044184.25410.39.

**[17]** M. Hamoda, M. Mamat, M. Rivaie, and Z. Salleh, “A conjugate gradient method with Strong Wolfe-Powell line search for unconstrained optimization,” *Appl. Math. Sci.,* vol. 10, no. 13–16, 2016, doi: 10.12988/ams.2016.56449.

**[18]** I.A.M. Moghrabi, “A non-Secant quasi-Newton Method for Unconstrained Nonlinear Optimization", Cogent Engineering, vol. 9, no. 1, 2022, doi: 10.1080/23311916.2021.2018929,” *J. Eng. Appl. Sci.,* vol. 12, no. 17, 2017, doi: 10.3923/jeasci.2017.4504.4507.

**[19]** H. N. Jabbar and B. A. Hassan, “Two-versions of descent conjugate gradient methods for large-scale unconstrained optimization,” *Indones. J. Electr. Eng. Comput. Sci.,* vol. 22, no. 3, 2021, doi: 10.11591/ijeecs.v22.i3.pp1643-1649.

**[20]** E. D. Dolan and J. J. Moré, “Benchmarking optimization software with performance profiles,” *Math. Program. Ser. B,* vol. 91, no. 2, pp. 201–213, 2002, doi: 10.1007/s101070100263.