Iranian Journal of Numerical Analysis and Optimization

Ferdowsi University of Mashhad

Reaserch Article

Enhancing conjugate gradient performance via a diagonal quasi-Newton strategy: Applications in adaptive filtering and power allocation

N. Bidabadi

DOI:10.22067/ijnao.2026.97071.1808

Vol. 17, No. 1, March 2027

pp. 359–379

Ferdowsi University of Mashhad, Iran

Iranian Journal of Numerical Analysis and Optimization
Vol. 17, No. 1, 2027, pp 359–379
https://doi.org/10.22067/ijnao.2026.97071.1808
https://ijnao.um.ac.ir/

CC BY NC

Research Article

Enhancing conjugate gradient performance via a diagonal quasi-Newton strategy: Applications in adaptive filtering and power allocation

N. Bidabadi*,

Abstract

Unconstrained optimization is fundamental to scientific computing. While nonlinear conjugate gradient (CG) methods are valued for their low memory requirements and simplicity, their convergence may be slow. Conversely, quasi-Newton methods like Broyden–Fletcher–Goldfarb–Shanno (BFGS) achieve faster superlinear convergence but become computationally prohibitive for large-scale problems. To address this trade-off, this paper introduces a novel hybrid algorithm that combines a diagonal quasi-Newton approach with a CG method. This framework leverages the complementary strengths of both techniques, maintaining modest memory usage while accelerating convergence. Numerical experiments on several benchmark problems, adaptive filter design and power allocation, demonstrate the proposed method's effectiveness. Our MATLAB-based evaluations show that the hybrid approach consistently outperforms standard CG and BFGS methods.

AMS subject classifications (2020): Primary 90C30; Secondary 65K05.

Keywords: Conjugate gradient; Diagonal quasi-Newton; Adaptive filtering; Power allocation

*Corresponding author

Received 18 December 2025; revised 18 February 2026; accepted 27 March 2026

Narges Bidabadi
Department of Mathematical Sciences, Yazd University, Yazd, Iran. e-mail: n_bidabadi@yazd.ac.ir

How to cite this article
Bidabadi, N., Enhancing conjugate gradient performance via a diagonal quasi-Newton strategy: Applications in adaptive filtering and power allocation. *Iran. J. Numer. Anal. Optim.*, 2027; 17(1): 359-379. https://doi.org/10.22067/ijnao.2026.97071.1808

1 Introduction

Unconstrained optimization, the problem of minimizing a nonlinear function

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

where $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable, forms the computational backbone of countless applications in engineering, data science, and computational physics. As the scale of these problems continues to increase, the development of efficient, scalable, and robust iterative algorithms remains a central pursuit in numerical optimization. Iterative methods for minimization are broadly characterized by their trade-off between computational cost per iteration and convergence rate. Two of the most prominent families of algorithms in this domain are quasi-Newton methods and nonlinear conjugate gradient (CG) methods.

Quasi-Newton methods emerged in the 1950s and 1960s as a revolutionary class of algorithms for unconstrained optimization, bridging the gap between simple gradient methods and computationally expensive Newton's method. The fundamental idea is to approximate the Hessian matrix (or its inverse) using only gradient information, thus avoiding the costly computation of second derivatives.

The first significant breakthrough came with the Davidon–Fletcher–Powell method proposed by Davidon [6] in 1959 and later popularized by Fletcher and Powell [10] in 1963. This method introduced the concept of updating an inverse Hessian approximation iteratively through rank-two updates that satisfy the secant condition.

However, the most influential development came with the Broyden–Fletcher–Goldfarb–Shanno (BFGS) method, which was independently developed around 1970 [4, 9, 11, 20]. BFGS quickly became the gold standard for quasi-Newton methods due to its numerical stability and superior performance in practice. The BFGS update maintains positive definiteness of the Hessian approximation under appropriate conditions and demonstrates robust convergence behavior.

A major weakness of the BFGS method is that as the dimension of the problem increases, its computational cost and memory requirements also increase. Furthermore, especially if the approximation of the Hessian matrix becomes ill-conditioned, a suitable solution may not be obtained. Numerous diagonal quasi-Newton (DQN) methods have been proposed in recent years to address these issues. For example, Enshaei, Leong, and Farid [8] (2016) proposed a DQN

method for solving unconstrained optimization problems. Assuming that the Hessian matrix at iteration k is diagonal and positive definite, they derived the Hessian matrix for the next iteration such that it has the shortest distance from the current iteration matrix in the Frobenius norm while satisfying the weak secant condition. Similarly, Sim et al. [21] developed a diagonally updated quasi-Newton formula using the log-determinant norm, ensuring that it satisfies the weak secant equation. More recently, in 2024, an accelerated structured DQN method was proposed for solving nonlinear least squares problems [13], followed in 2025 by a DQN method designed for nonlinear equations [5].

A fundamental contribution to optimization theory was made by Polyak [18], who systematically analyzed the CG method for extremal problems. While nonlinear CG methods are valued for their low memory requirements and simplicity, they are often hampered by slow convergence and sensitivity to problem conditioning. This stems from their reliance on gradient information alone, which can lead to inefficient zigzagging search paths, especially on ill-conditioned or non-quadratic landscapes.

To address this weakness, many improvements to the CG algorithm have been proposed. For example, in 2009, Andrei proposed a fundamentally different CG method in which the conjugacy parameter is computed by approximating the Hessian/vector product through finite differences. Additionally, for the step length computation, he suggested an acceleration scheme capable of improving the efficiency of the algorithm [3]. More recently, Zhu, Zhu, and Tan [22] proposed an adaptive two-parameter accelerated CG method that satisfies the sufficient descent condition for the search direction.

The complementary strengths and weaknesses of these two families suggest a compelling opportunity for a hybrid approach. Aini et al. [1] used the quasi-Newton direction instead of the steepest descent direction when calculating the search direction in the CG method. In 2023, Osinuga and Stanimirovic [17] proposed a modification of hybrid CG methods by appropriately combining the BFGS update with hybridized CG parameters.

In this paper, we propose a new hybrid algorithm that integrates a DQN method and a CG method. The rest of this paper is organized as follows: Section 2 reviews necessary background on CG and quasi-Newton methods. Section 3 details the derivation and mechanics of our proposed hybrid algorithm. Section 4 presents the convergence analysis. Section 5 presents the numerical results, and Section 6 concludes the paper.

2 Preliminaries

Iterative methods for solving problem (1), by using initial point x_0, generate a sequence of iterates x_k according to the update formula:

$$x_{k+1} = x_k + \alpha_k d_k, \tag{2}$$

where $\alpha_k > 0$ denotes the step length and d_k represents the search direction at the kth iteration. In quasi-Newton methods, the search direction d_k is computed by solving the linear system:

$$B_k d_k = -g_k, \tag{3}$$

where $g_k = \nabla f(x_k)$ is the gradient and B_k is a symmetric matrix that approximates the Hessian of f at x_k. The construction of B_k relies on the secant condition, which requires the matrix to satisfy

$$B_k s_{k-1} = y_{k-1}, \tag{4}$$

where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. This condition ensures that B_k mimics the behavior of the true Hessian along s_{k-1}.

Among the most important quasi-Newton methods is the BFGS method, in which the Hessian matrix is updated at each iteration using the following relation:

$$B_{k+1} = B_k + \frac{y_k y_k^\top}{y_k^\top s_k} - \frac{B_k s_k s_k^\top B_k}{s_k^\top B_k s_k}. \tag{5}$$

In contrast, the nonlinear CG method computes the search direction recursively as

$$d_k = \begin{cases} -g_k & k = 0, \\ -g_k + \beta_{k-1} d_{k-1}, & k = 1, 2, \ldots, \end{cases} \tag{6}$$

where the scalar β_k is a conjugacy parameter that distinguishes different CG variants. The Rivaie–Mamat–Ismail–Leong (RMIL) formula is given by [19]

$$\beta_{k-1} = \frac{g_k^T y_{k-1}}{d_{k-1}^T d_{k-1}}. \tag{7}$$

Note: The step length $\alpha_k \in (0, \alpha_{\max})$ in this study is determined by the Wolfe conditions [16]:

$$\begin{aligned} f(x_k + \alpha_k d_k) &\leq f(x_k) + c_1 \alpha_k g_k^T d_k, \\ \nabla f(x_k + \alpha_k d_k)^T d_k &\geq c_2 \nabla f(x_k)^T d_k, \end{aligned} \tag{8}$$

with $0 < c_1 < c_2 < 1$. Ensuring that the line search satisfies the Wolfe conditions guarantees that $s_k^T y_k > 0$ for all k [16].

3 A combined CG and DQN approach

The search direction d_k in our hybrid algorithm is determined through an adaptive two-case strategy:

Case 1: DQN phase

Under the condition that $s_{k-1}^T y_{k-1} - \|s_{k-1}\|^2 > 0$, the diagonal positive definite matrix $B_k = Diag(b_k^1, \ldots, b_k^n)$ is built such that it satisfies the weak secant equation

$$s_{k-1}^T B_k s_{k-1} = s_{k-1}^T y_{k-1}, \tag{9}$$

while guaranteeing that the diagonal entries of the matrix avoid becoming excessively small, thus maintaining numerical stability. The matrix B_k is determined by solving the optimization problem

$$\begin{aligned} &\min \ trace(B_k) = b_k^1 + \cdots + b_k^n \\ &s.t. \ s_{k-1}^T B_k s_{k-1} = s_{k-1}^T y_{k-1}, \\ &\quad b_k^i \geq 1, \quad i = 1, \ldots, n. \end{aligned} \tag{10}$$

The problem (10) is equivalent to

$$\begin{aligned} &\min \tilde{b_k}^1 + \cdots + \tilde{b_k}^n \\ &s.t. \sum_{i=1}^{n} (s_{k-1}^i)^2 \tilde{b_k}^i = s_{k-1}^T y_{k-1} - \|s_{k-1}\|^2, \end{aligned}$$

$$\tilde{b_k}^i \geq 0, \quad i = 1, \ldots, n, \tag{11}$$

where $\tilde{b_k}^i = b_k^i - 1$. This optimization problem bears structural similarities to the knapsack problem and admits an optimal solution given by

$$\tilde{b_k}^i = \begin{cases} \frac{s_{k-1}^T y_{k-1} - \|s_{k-1}\|^2}{(s_{k-1}^j)^2}, & \text{if } j = \arg\max\{(s_{k-1}^i)^2\}, \\ 0, & o.w. \end{cases} \tag{12}$$

Therefore, the optimal solution to problem (10) is characterized by

$$b_k^i = \begin{cases} \frac{s_{k-1}^T y_{k-1} - \|s_{k-1}\|^2}{(s_{k-1}^j)^2} + 1, & \text{if } j = \arg\max\{(s_{k-1}^i)^2\}, \\ 1, & o.w. \end{cases} \tag{13}$$

Case 2: CG phase

When $s_{k-1}^T y_{k-1} - \|s_{k-1}\|^2 \leq 0$, we use the CG-RMIL method; that is, the search direction is calculated by (6), where β_{k-1} is computed by (7).

Consequently, the new hybrid algorithm is as follows.

Algorithm 1 A combined CG and DQN algorithm (CG-DQN)

Require: a starting point x_0, a stopping parameter $\epsilon > 0$, and constants c_1 and c_2 satisfying $0 < c_1 < c_2 < 1$.

1:
2: Set $k = 0$. Calculate $g_0 = \nabla f(x_0)$. If $\|g_0\| < \epsilon$, stop. Otherwise set $d_0 = -g_0$ and go to step 4.
3: If $s_{k-1}^T y_{k-1} - \|s_{k-1}\|^2 \geq 0$, calculate b_k^i, $i = 1, \ldots, n$, using relation (13), calculate the search direction d_k using $d_k^i = -\frac{g_k^i}{b_k^i}$ and go to step 5. Otherwise, proceed to the next step.
4: Calculate β_{k-1} and the search direction d_k using (7) and (6).
5: Calculate α_k satisfying (8). Set $x_{k+1} = x_k + \alpha_k d_k$, and $g_{k+1} = \nabla f(x_{k+1})$.
6: If $\|g_{k+1}\| < \epsilon$, stop and accept x_{k+1} as an approximate solution to the problem; otherwise calculate the values of $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$. Set $x_k = x_{k+1}$, $g_k = g_{k+1}$ and $k = k + 1$, go to step 2.

4 Convergence analysis

We now analyze the convergence properties of the proposed algorithm. First, we recall an important convergence theorem for iterative methods known as the Zoutendijk theorem.

Theorem 1. [24] Let $f : \mathbb{R}^n \to \mathbb{R}$ be a function bounded from below, and twice continuously differentiable on an open set Ω that contains the sublevel set $\{x : f(x) \leq f(x_0)\}$, where x_0 is the algorithm's starting point. Suppose further that $g = \nabla f$ is Lipschitz continuous on Ω. If the search direction d_k is a descent direction and the step length α_k satisfies the Wolfe conditions, then

$$\sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} < \infty. \tag{14}$$

For the convergence proof of Algorithm 1, we adopt the following standard assumptions.

Assumption (A1): The level set $\mathcal{L}_0 = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$ is bounded. That is, there exists a constant $M > 0$ such that

$$\|x\| \leq M, \quad \text{for all } x \in \mathcal{L}_0. \tag{15}$$

Assumption (A2): In some neighborhood Ω of $\mathcal{L}_0$, the function f is continuously differentiable, and its gradient is Lipschitz continuous. Namely, there exists a constant $L > 0$ such that

$$\|g(x) - g(y)\| \leq L\|x - y\|, \quad \text{for all } x, y \in \Omega. \tag{16}$$

A direct consequence of assumptions (A1) and (A2) is that

$$\|x - y\| \leq 2M, \quad \text{for all } x, y \in \mathcal{L}_0, \tag{17}$$

and there exists a constant $\gamma > 0$ such that

$$\|g(x)\| \leq \gamma, \quad \text{for all } x \in \mathcal{L}_0. \tag{18}$$

In the following result, we show that the search direction of Algorithm 1 is a descent direction.

Lemma 1. Under assumption (A2), the direction d_k in Algorithm 1 satisfies the sufficient descent condition, meaning there exists a constant $c \in (0, 1)$ such that

$$g_k^T d_k \leq -c\|g_k\|^2. \tag{19}$$

Proof. In case 1, we have

$$g_k^T d_k = -g_k^T B_k^{-1} g_k \leq -\|g_k\|^2 \sigma_{\min}(B_k^{-1}) = -\frac{\|g_k\|^2}{\|B_k\|},$$

where $\sigma_{\min}$ denotes the smallest singular value of the matrix. Since,

$$\|B_k\| = \max_i b_k^i = \frac{s_{k-1}^T y_{k-1} - \|s_{k-1}\|^2}{(s_{k-1}^j)^2} + 1, \quad j = \arg\max\{(s_{k-1}^i)^2\},$$

and

$$\frac{s_{k-1}^T y_{k-1} - \|s_{k-1}\|^2}{(s_{k-1}^j)^2} \leq \frac{s_{k-1}^T y_{k-1}}{(s_{k-1}^j)^2} \leq \frac{\|s_{k-1}\|\|y_{k-1}\|}{(s_{k-1}^j)^2} \leq \frac{L\|s_{k-1}\|^2}{(s_{k-1}^j)^2} \leq nL,$$

we conclude that $\|B_k\| \leq nL + 1$, we obtain

$$g_k^T d_k \leq -\frac{\|g_k\|^2}{nL+1}.$$

In case 2, we have

$$\begin{aligned} g_k^T d_k &= -g_k^T g_k + \beta_{k-1} d_{k-1}^T g_k = -\|g_k\|^2 + \frac{g_k^T y_{k-1}}{d_{k-1}^T d_{k-1}} d_{k-1}^T g_k \\ &\leq -\|g_k\|^2 + \|g_k\|^2 \frac{\|y_{k-1}\|\|d_{k-1}\|}{\|d_{k-1}\|^2} \\ &\leq -\|g_k\|^2 + \|g_k\|^2 \frac{L\alpha_{k-1}\|d_{k-1}\|\|d_{k-1}\|}{\|d_{k-1}\|^2} \\ &\leq -\|g_k\|^2 (1 - L\alpha_{\max}). \end{aligned}$$

By choosing $\alpha_{\max} < \frac{1}{L}$, we have $1 - L\alpha_{\max} \in (0, 1)$.
Therefore

$$g_k^T d_k \leq -c\|g_k\|^2,$$

with $c = \min(1 - L\alpha_{\max}, \frac{1}{nL+1})$. □

The convergence result is presented in the following theorem.

Theorem 2. Under assumptions (A1) and (A2), the sequence $\{g_k\}$ generated by the CG-DQN algorithm satisfies

$$\liminf_{k\to\infty} \|g_k\| = 0. \tag{20}$$

Proof. The proof follows by contradiction. Assuming $\liminf_{k\to\infty} \|g_k\| \neq 0$, implies $\|g_k\| \geq \xi > 0$ for all k. We first show that $\|d_k\|$ is bounded.

In case 1, we have $b_k^i \geq 1$, for all i which yields to $\|B_k^{-1}\| \leq 1$. Thus,

$$\|d_k\| = \|B_k^{-1} g_k\| \leq \|g_k\| \leq \gamma.$$

In case 2, we have

$$\begin{aligned}
\|d_k\| &= \| - g_k + \beta_{k-1} d_{k-1}\| \\
&\leq \|g_k\| + \frac{\|g_k\|\|y_{k-1}\|\|d_{k-1}\|}{\|d_{k-1}\|^2} \\
&\leq \|g_k\| + \frac{\|g_k\| L\|s_{k-1}\|\|d_{k-1}\|}{\|d_{k-1}\|^2} \\
&= \|g_k\| + \frac{\|g_k\| L\alpha_{k-1}\|d_{k-1}\|\|d_{k-1}\|}{\|d_{k-1}\|^2} \\
&\leq \|g_k\|(1 + L\alpha_{\max}) \\
&\leq \gamma(1 + L\alpha_{\max}).
\end{aligned}$$

Since $\|d_0\| = \| - g_0\| \leq \gamma$, we conclude that

$$\|d_k\| \leq \Gamma, \qquad k = 0, 1, \ldots,$$

where

$$\Gamma = \gamma(1 + L\alpha_{\max}).$$

This implies

$$\frac{1}{\|d_k\|} \geq \frac{1}{\Gamma} > 0, \qquad k = 0, 1, \ldots,$$

and

$$\sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2} = +\infty. \tag{21}$$

On the other hand, using relation (19), we have

$$(g_k^T d_k)^2 \geq c^2 \|g_k\|^4 \geq c^2 \xi^4,$$

which together with relation (14) implies that

$$\infty > \sum_{k=0}^{\infty} \frac{(g_k^T d_k)^2}{\|d_k\|^2} \geq \sum_{k=0}^{\infty} \frac{c^2 \xi^4}{\|d_k\|^2} = c^2 \xi^4 \sum_{k=0}^{\infty} \frac{1}{\|d_k\|^2},$$

which directly contradicts condition (21). □

5 Numerical results

A numerical comparative analysis was conducted to evaluate the performance of the proposed CG-DQN algorithm against the CG-RMIL and BFGS methods. In the BFGS method, the search direction d_k is calculated by (3), where B_k is updated by (5) and the initial matrix was taken as the identity matrix. Also, we solve system (3) using a modified Cholesky decomposition and therefore the resulting search direction will always be a descent direction. All algorithms were implemented in MATLAB R2020b and executed on a system with a 3.3 GHz Intel Core i3 processor and 16 GB of RAM. For a fair comparison, we used a Wolfe line search with parameters $c_1 = 0.2$ and $c_2 = 0.8$. The maximum number of iterations was set to 200. The numerical results are presented and analyzed in the following three subsections.

5.1 Standard benchmarks

We evaluated the algorithms on 102 test problems (see [2]) across multiple dimensions (n). Each run was stopped upon satisfying any of the following conditions:

- $\|\nabla f(x_k)\| < 10^{-4}$.
- $\|x_{k+1} - x_k\|_\infty < 10^{-6}$.
- $\|x_{k+1} - x_k\|_2 < 10^{-6}$.

When $d_k^T g_k > -10^{-1}\|d_k\|\|g_k\|$, we restarted the search directions of the CG-DQN and CG-RMIL algorithms to the steepest-descent direction. For our comparisons, we used the notion of a performance profile [7] to evaluate and compare the performance of the set of solvers S on a test set P. Assume that we have n_s solvers and n_p problems. We are interested in using the computational time as a performance measure. Suppose $T_{p,s}$ is the computational time required to solve problem p by algorithm s. We compare the performance on problem p by solver s with the best performance by any solver on this problem; that is, we use the performance ratio

$$r_{p,s} := \frac{T_{p,s}}{\min_s T_{p,s}}.$$

We assume that a parameter R_m is chosen so that $R_m \geq r_{p,s}$, for all p and s, and we put $r_{p,s} = R_m$ if and only if algorithm s does not solve problem p. If we define

$$\rho_s(\tau) := \frac{size\{p \in P \ : \ r_{p,s} \leq \tau\}}{n_p},$$

then $\rho_s(\tau)$ is the probability that solver $s \in S$ that a performance ratio $r_{p,s}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio. The function ρ_s is the distribution function for the performance ratio. The value of $\rho_s(1)$ is the probability that the solver will win over the rest of the solvers. The Dolan–Moré [7] performance profiles in Figures 1 and 2 compare algorithms based on computational time. They confirm that the proposed method is more efficient than both CG-RMIL and BFGS approaches.

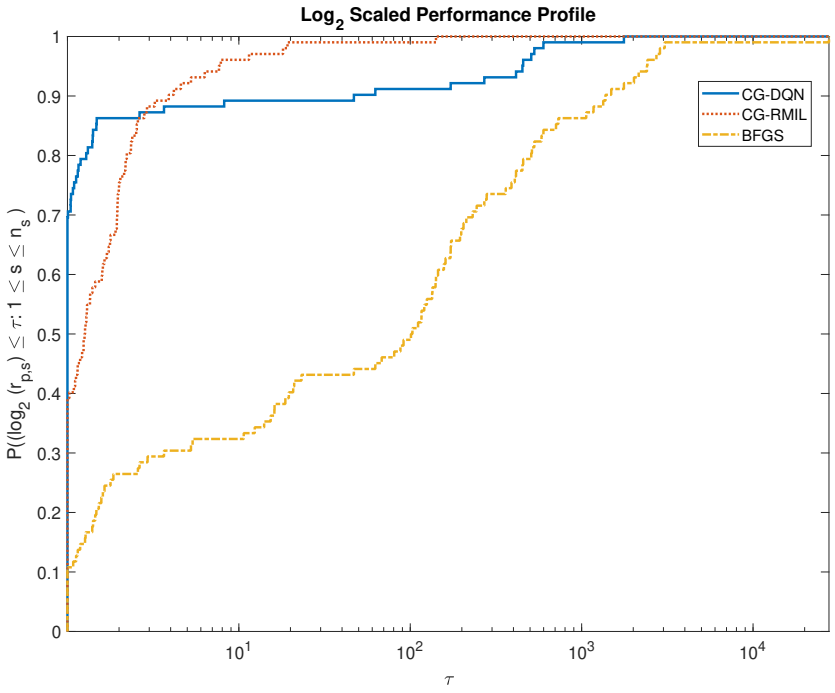

Figure 1: Performance profile based on central processing unit (CPU) time for problems with $n = 500$

5.2 Application to adaptive filter design problem

In this subsection, we apply the proposed algorithm to the fundamental signal processing problem of adaptive filter design. The objective is to perform system identification by finding the optimal coefficients of a finite impulse response (FIR) filter that minimize the prediction error between a desired signal and the filter's output [15, 12].

Problem formulation: The output of an N-tap FIR filter at time t is given by

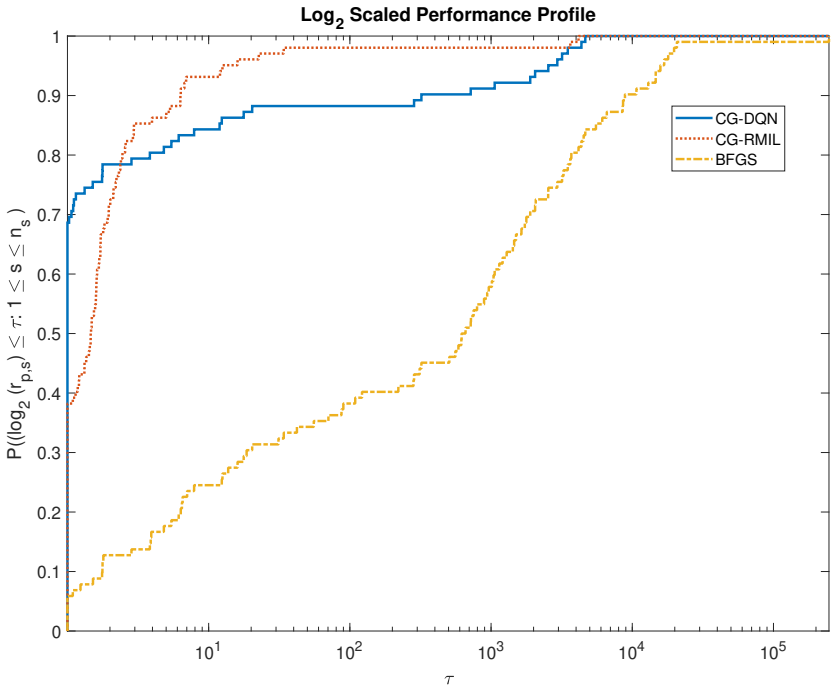

Figure 2: Performance profile based on CPU time for problems with $n = 1000$

$$y[t] = \sum_{k=0}^{N-1} w[k] \cdot x[t-k] = \mathbf{w}^T \mathbf{x}[t],$$

where $\mathbf{w} = [w[0], w[1], \ldots, w[N-1]]^T$ is the vector of filter coefficients (the optimization variables), and $\mathbf{x}[t] = [x[t], x[t-1], \ldots, x[t-N+1]]^T$ is the input vector. The instantaneous error is

$$e[t] = d[t] - y[t],$$

with $d[t]$ being the desired signal. The goal is to minimize the mean square error,

$$\min_{\mathbf{w}} J(\mathbf{w}) = \mathbb{E}\left[|e[t]|^2\right] = \mathbb{E}\left[|d[t] - \mathbf{w}^T \mathbf{x}[t]|^2\right].$$

For practical optimization, we use the sample average approximation over M data points

$$\min_{\mathbf{w} \in \mathbb{R}^N} J(\mathbf{w}) = \frac{1}{M} \sum_{t=N}^{M} \left(d[t] - \mathbf{w}^T \mathbf{x}[t]\right)^2.$$

To simulate a realistic system identification scenario, synthetic data were generated as follows. The input signal $x[t]$ was produced as colored noise using a second-order autoregressive (AR(2)) process:

$$x[t] = 0.5x[t-1] - 0.1x[t-2] + v[t], \quad v[t] \sim \mathcal{N}(0,1), \tag{22}$$

where $\mathcal{N}(0,1)$ is zero-mean, unit-variance Gaussian white noise. The generated signal was then normalized to unit variance:

$$x[t] \leftarrow \frac{x[t]}{\sigma_x}. \tag{23}$$

The desired signal was generated according to the system identification model:

$$d[t] = \mathbf{w}_{\text{Opt}}^T \mathbf{x}[t] + \eta[t], \quad \eta[t] \sim \mathcal{N}(0, \sigma_\eta^2), \tag{24}$$

with noise variance $\sigma_\eta^2 = 0.0004$, corresponding to a signal-to-noise ratio of approximately 34 dB. The true (unknown) system was modeled by a coefficient vector $\mathbf{w}_{\text{Opt}}$ that follows an exponential decay pattern with minor random perturbations:

$$\mathbf{w}_{\text{Opt}} = [0.9, 0.7, 0.5, 0.3, 0.1, \epsilon_6, \epsilon_7, \ldots, \epsilon_N]^T, \tag{25}$$

where $\epsilon_i \sim \mathcal{N}(0, 0.0025)$ for $i = 6, 7, \ldots, N$ represent small random perturbations.
The problem dimensions were set to a filter length of $N = 32$ coefficients, $M = 500000$ data samples, leading to a 32-dimensional optimization over $\mathbf{w} \in \mathbb{R}^{32}$.

We executed the BFGS, CG-RMIL, and CG-DQN algorithms to solve this problem. A common stopping criterion of $\|\nabla J(\mathbf{w})\| < 10^{-4}$ was used for all methods. Figures 3–5 present comparative analyses of the BFGS, CG-RMIL, and proposed CG-DQN algorithms, respectively, through the following subplots:

- Convergence Analysis: shows the exponential decay of both the objective function value and the gradient norm, indicating rapid convergence toward the optimum.

- Filter Coefficients Comparison: compares the estimated filter coefficients against the true optimal values.

- Signal Prediction: demonstrates the filtering performance by plotting the desired signal $d[t]$ alongside the filter output $y[t]$ over a representative sample range. The close alignment and minimal residual error $e[t] = d[t] - y[t]$ confirm the filter's effectiveness in tracking the system behavior.

- Error Distribution: presents a histogram of the prediction errors $e[t]$. The distribution approximates a zero-mean Gaussian profile, indicating successful modeling with unbiased estimation and appropriate model complexity.

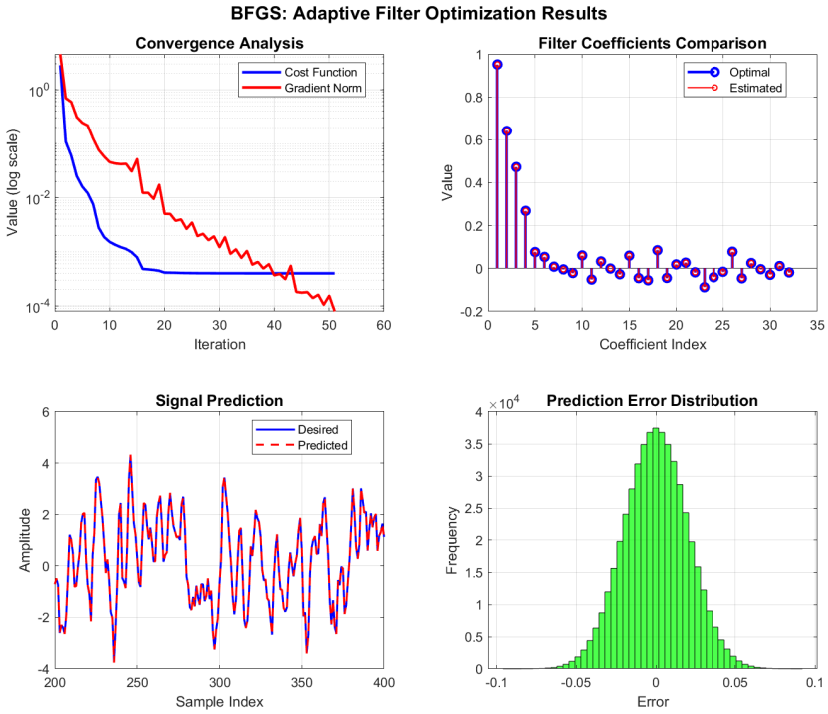

Figure 3: The results obtained from implementing the BFGS algorithm on the adaptive filter design problem.

Collectively, these analyses substantiate the efficacy of the proposed method and show robust convergence, accurate parameter estimation, and excellent predictive performance.

5.3 Application to power allocation problem

This subsection presents the results of applying the proposed algorithm to the power minimization problem in the uplink of a single-input single-output system with K users and one base station. Efficient power allocation is critical in wireless networks to balance energy efficiency with quality-of-service requirements [23, 14].

The problem is formulated as a resource allocation task. Let

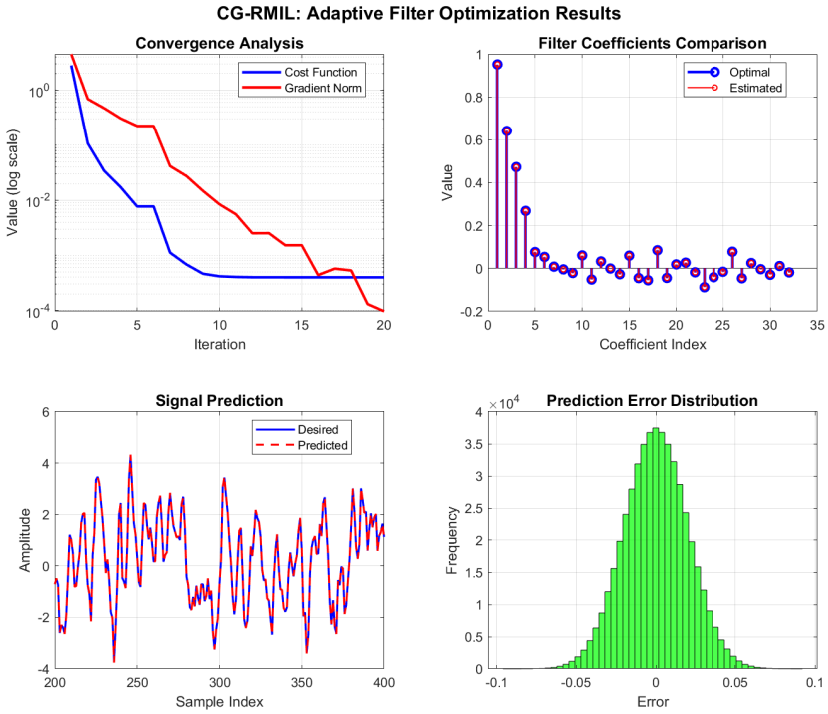

Figure 4: The results obtained from implementing the CG-RMIL algorithm on the adaptive filter design problem.

$$\mathbf{p} = [p_1, p_2, \ldots, p_K], \tag{26}$$

denote the vector of transmit powers for all users. The achievable uplink rate for user i is given by

$$R_i(\mathbf{p}) = \log_2 \left(1 + \frac{p_i |h_{ii}|^2}{\sum_{j \neq i} p_j |h_{ij}|^2 + \sigma^2} \right), \tag{27}$$

where h_{ii} is the direct channel gain between user i and the base station, h_{ij} is the interference channel gain from user j and σ^2 is the noise variance. If $R_{\min}$ is the minimum required rate for user i, and $p_i^{\min}$ and $p_i^{\max}$ denote the minimum and maximum allowable transmit powers for user i, respectively, then the power minimization problem is formulated as follows:

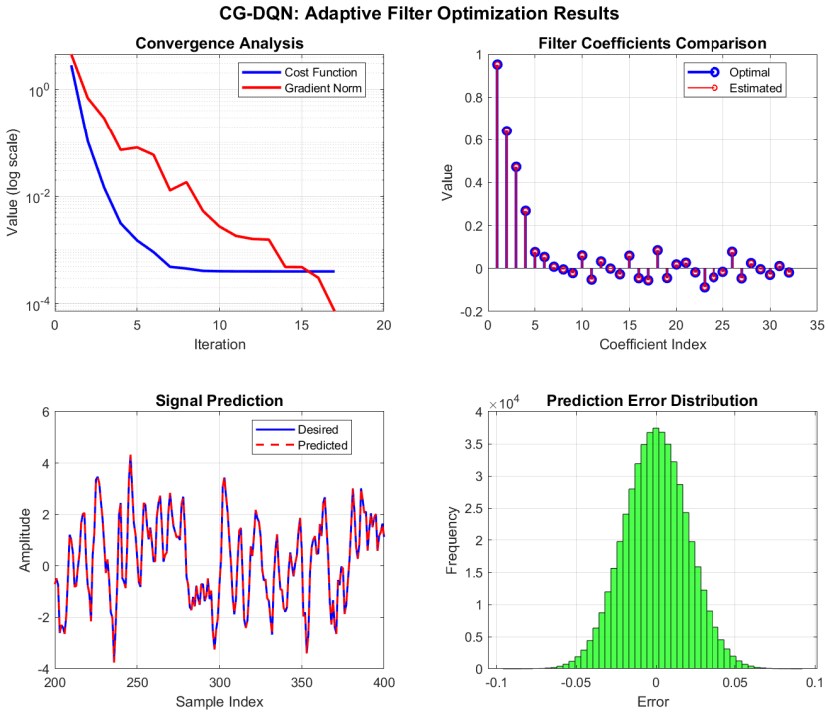

Figure 5: The results obtained from implementing the CG-DQN algorithm on the adaptive filter design problem.

$$\begin{aligned} \min \quad & \sum_{i=1}^{K} p_i \\ s.t. \quad & R_i(\mathbf{p}) \geq R_{\min}, \qquad i = 1, \ldots, K, \\ & p_i^{\min} \leq p_i \leq p_i^{\max}, \qquad i = 1, \ldots, K. \end{aligned} \tag{28}$$

By incorporating a penalty term for constraint violations, we reformulate the constrained problem into an unconstrained one.

$$\min_{\mathbf{p}} J(\mathbf{p}) = \sum_{i=1}^{K} p_i + \lambda \sum_{i=1}^{K} [\max(0, R_{\min} - R_i(\mathbf{p}))]^2. \tag{29}$$

For the bound constraints, we also employ a projection operator that maps the value of p_i to the closest point within the interval $[p_i^{\min}, p_i^{\max}]$.

This penalized problem was solved using the BFGS, CG-RMIL, and the proposed CG-DQN algorithms. The simulation parameters are summarized in Table 1.

Table 1: Simulation parameter settings

Parameter	**Value**
Number of users	$K = 5$
Minimum rate requirement	$R_{\min} = 1.0$ bps/Hz
Noise power	$\sigma^2 = 0.01$
Transmit power constraints	$0.001 \text{ W} \leq p_i \leq 2.0 \text{ W}$
Penalty parameter	$\lambda = 10$
Direct channel gains	$h_{ii} \sim \mathcal{U}(0.5, 1.0)$
Interference channel gains	$h_{ij} \sim \mathcal{U}(0.1, 0.3),\ i \neq j$

The results obtained from the BFGS, CG-RMIL, and CG-DQN algorithms are shown in Figure 6. The algorithms were programmed to stop running if $\|\nabla J(\mathbf{p})\| < 10^{-4}$. The approximate value of the ith component of the gradient is obtained from the following relation:

$$\nabla^i J(\mathbf{p}) = \frac{J(\mathbf{p} + 10^{-6} I_i) - J(\mathbf{p})}{10^{-6}},$$

where I_i is the ith column of the identity matrix.

In Figure 6, the first subplot illustrates the convergence behavior of the algorithms. As shown, the proposed CG-DQN method converges to a stationary point faster than CG-RMIL. The BFGS method, however, fails to converge. As one can see in this figure, the BFGS method showed good performance in the first 4 iterations and even achieved greater reduction compared to the other two methods. However, from the fifth iteration onward, the decreasing trend of the algorithm became very slow and almost stopped. The step length from the fourth iteration onward is very small, and the reason for this appears to be the approximate calculation of both the gradient and the Hessian. Although the direction calculated using the modified Cholesky decomposition is a descent direction, but from iteration 4 onward, the step length that can achieve sufficient reduction in the algorithm has become very small (this means that the approximations are not suitable), and practically no improvement is obtained in the algorithm.

Despite their different convergence speeds, CG-DQN and CG-RMIL both arrive at essentially the same optimal solution, which is verified in the remaining subplots.

The second subplot shows the optimized transmit power allocation. Both algorithms produce similar nonuniform distributions, reflecting the underlying channel and interference conditions: Users with weaker channels are assigned higher power to compensate, while those with stronger channels transmit at much lower levels.

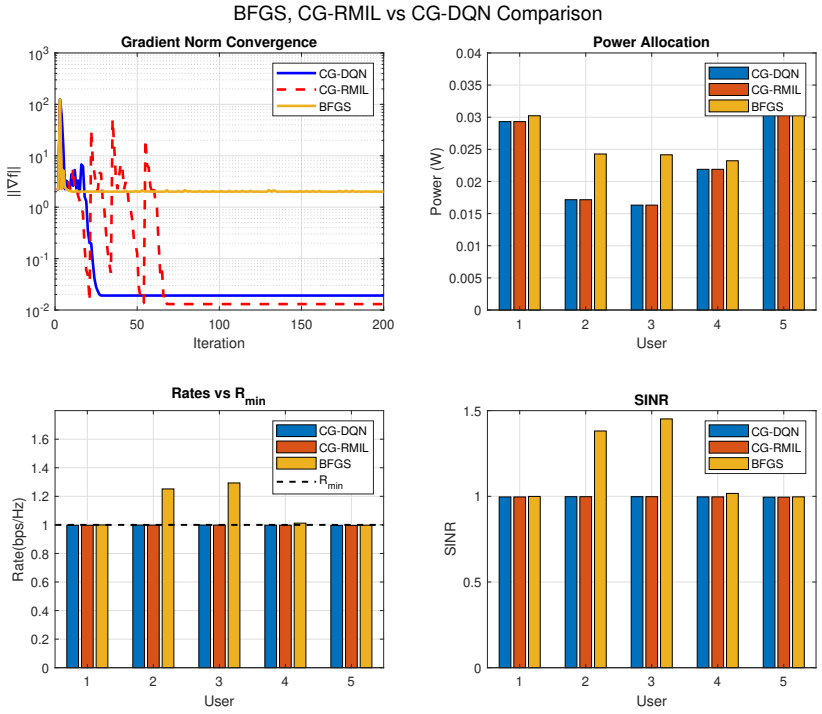

Figure 6: The results obtained from implementing the BFGS, CG-RMIL and CG-DQN algorithms on the power allocation problem.

The third subplot compares the achieved data rates with the minimum requirement. Both algorithms allocate just enough power to meet this threshold, confirming that the optimization avoids unnecessary power expenditure.

Finally, the fourth subplot shows the resulting signal-to-interference-plus-noise ratio (SINR) for each user. The SINR levels align with the allocated powers and achieved rates, showing that each method precisely adjusts the transmit powers to satisfy the performance requirement while minimizing total power consumption.

These results confirm that the proposed algorithm provides an efficient solution to the power allocation problem in wireless communication systems.

6 Conclusion

This paper presents a novel algorithmic framework for unconstrained optimization, introducing a hybrid method that innovatively combines the RMIL conjugate gradient approach with a diagonal quasi-Newton strategy. The algorithm is built on rigorous mathematical foundations, ensuring formally guaranteed global convergence properties. Extensive numerical experiments validate the computational efficiency of the proposed method. The evaluations, conducted on a diverse set of standard test problems, as well as two real-world engineering applications an adaptive filter design problem and a power allocation problem, demonstrate its practical superiority. The results indicate that the hybrid method not only outperforms the standard RMIL conjugate gradient method but also shows significant advantages over the classical BFGS algorithm in terms of computational time.

Funding

This research received no external funding.

Conflicts of Interest

The author declares no conflicts of interest.

References

[1] Aini, N., Hajar, N., Mamat, M., Zull, N. and Rivaie, M., *Hybrid quasi-Newton and conjugate gradient method for solving unconstrained optimization problems*, J. Eng. Appl. Sci. 12(18) (2017), 4627–4631.

[2] Andrei, N., *An unconstrained optimization test functions collection*, Adv. Model. Optim. 10 (2008), 147–161.

[3] Andrei, N., *Accelerated conjugate gradient algorithm with finite difference Hessian/vector product approximation for unconstrained optimization*, J. Comput. Appl. Math. 230(2) (2009), 570–582.

[4] Broyden, C. G., *The convergence of a class of double-rank minimization algorithms. Part I. The new algorithm*, J. Inst. Math. Appl. 6 (1970), 76–90.

[5] Cao, H. and Li, W., *A diagonal quasi-Newton method based on automatic differentiation for nonlinear equations*, Calcolo 62(2) (2025), 1–17.

[6] Davidon, W. C., *Variable metric method for minimization*, Technical Report ANLC5990 (Revised), Argonne National Laboratory, Argonne, 1959.

[7] Dolan, E. D. and Moré, J. J., *Benchmarking optimization software with performance profiles*, Math. Program. 91 (2002), 201–213.

[8] Enshaei, S., Leong, W. H. and Farid, M., *Diagonal quasi-Newton method via variational principle under generalized Frobenius norm*, Optim. Methods Softw. 31(6) (2016), 1258–1271.

[9] Fletcher, R., *A new approach to variable metric algorithms*, Comput. J. 13(3) (1970), 317–322.

[10] Fletcher, R. and Powell, M. J. D., *A rapidly convergent descent method for minimization*, Comput. J. 6 (1963), 163–168.

[11] Goldfarb, A., *A family of variable metric methods derived by variational means*, Math. Comput. 24(109) (1970), 23–26.

[12] Haykin, S., *Adaptive filter theory*, Prentice-Hall, Inc., Upper Saddle River, NJ, 1996.

[13] Huynh, D. Q. and Hwang, F. N., *An accelerated structured quasi-Newton method with a diagonal second-order Hessian approximation for nonlinear least squares problems*, J. Comput. Appl. Math. 442 (2024), 115718.

[14] Liu, M., Zhang, J., Xiong, K., Zhang, M., Fan, P. and Letaief, K. B., *Effective user clustering and power control for multiantenna uplink NOMA transmission*, IEEE Trans. Wireless Commun. 21(11) (2022), 8995–9009.

[15] Monson, H., *Statistical digital signal processing and modeling*, John Wiley & Sons, Hoboken, NJ, 1996.

[16] Nocedal, J. and Wright, S. J., *Numerical optimization*, Springer Series in Operations Research and Financial Engineering, Springer, New York, 2006.

[17] Osinuga, I. and Stanimirovic, P., *Another hybrid BFGS-conjugate gradient algorithm for unconstrained optimization*, AIP Conf. Proc. 2852 (2023), 130006.

[18] Polyak, B. T., *The conjugate gradient method in extremal problems*, USSR Comput. Math. Math. Phys. 9(4) (1969), 94–112.

[19] Rivaie, M., Mamat, M., Leong, W. and Ismail, M., *A new conjugate gradient coefficient for large scale nonlinear unconstrained optimization*, Int. J. Math. Anal. 6(23) (2012), 1131–1146.

[20] Shanno, D.F.d, *Conditions of quasi-Newton methods for function minimization*, Math. Comput. 24(111) (1970), 647–650.

[21] Sim, H. S., Leong, W. J., Chen, C. Y. and Ibrahim, S. N. I., *Diagonal quasi-Newton updating formula using log-determinant norm*, in: *Advances in Industrial and Applied Mathematics: Proceedings of the 23rd Malaysian National Symposium of Mathematical Sciences (SKSM23)*, (2016), 1–7.

[22] Zhu, Z., Zhu, X. and Tan, Z., *An accelerated conjugate gradient method with adaptive two-parameter with applications in image restoration*, Comput. Appl. Math. 43 (2024), Article 116.

[23] Zola, E. and Martin-Escalona, I., *A robust user association, backhaul routing, and switching off model for a 5G network with variable traffic demands*, IEEE Access 8 (2020), 96714–96726.

[24] Zoutendijk, G., *Nonlinear programming, computational methods*, in: J. Abadie (ed.), *Integer and Nonlinear Programming*, North-Holland Publishing Company, Amsterdam and London, (1970), 37–86.