International Journal of Applied Mathematics

Volume 39 No. 1s, 2026

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

FUZZY FRACTIONAL CONTINUOUS HOPFIELD NETWORKS BASED ON THE GRÜNWALD–LETNIKOV FORMULATION FOR OPTIMAL ECONOMIC DISPATCH PROBLEMS

Abdellah Ahourag[1], Zakaria Bouhanch[1], Mohammed Roudani[1], Karim El Moutaouakil[1]

[1]Laboratory of Mathematics and Data Science Faculty of Polydisciplinary Studies of Taza

Sidi Mohamed Ben Abdellah University Fez, Morocco

Abstract

Continuous Hopfield Networks (CHNs) have long served as efficient neurodynamic solvers for constrained optimization problems due to their parallel computing and convergence capabilities. However, their reliance on fixed weight and bias parameters limits adaptability in dynamic and uncertain environments. To overcome these limitations, we propose a novel Fuz-Frac-CHN Fractional CHN model that integrates fractional-order calculus with a fuzzy logic system to dynamically tune Grunwald–Letnikov (GL) coefficients based on real-time system feedback.

The proposed method is applied to the Economic Dispatch (ED) problem I power systems, aiming to minimize generation costs while satisfying power balance and generator constraints. Extensive simulations show that the Fuz-Frac-CHN model achieves superior performance in convergence speed, solution feasibility, and robustness compared to both classical CHN and fixed-order fractional CHN models. In particular, it maintains negligible power mismatch (on the order of 10^{-13}MW) while achieving competitive cost outcomes across varying fractional orders. The adaptive tuning of GL memory weights enables the network to self-regulate learning dynamics, improving optimization under different operating conditions. Beyond the ED case, the proposed framework demonstrates potential for generalization to a wide range of constrained optimization problems in control systems, intelligent networks, and industrial engineering. The integration of fuzzy rules with memory-aware neural computation offers a scalable, flexible, and intelligent optimization paradigm that bridges the gap between neurodynamics, fractional calculus, and soft computing.

Key Words and Phrases: Fractional calculus, Grünwald–Letnikov operator, continuous Hopfield networks, nonlinear dynamical systems, constrained optimization.

1. Introduction

Neural networks have played a pivotal role in advancing intelligent processing technologies, particularly in areas such as optimization, associative memory, and fault tolerance. Among these models, the Continuous Hopfield Network (CHN) has been extensively studied due to its ability to minimize an energy function and converge toward stable equilibrium states [1].

Since its introduction by Hopfield in the early 1980s [2], recurrent neural architectures have found widespread applications in combinatorial optimization, image processing, pattern recognition, and error correction in communication systems [3,4,5]. However, classical CHNs rely on fixed weight matrices and static bias terms, which significantly limit their

Received: Aug. 08, 2025

adaptability to dynamic and uncertain environments.

1.1 Hopfield Recurrent Neural Networks: Foundational Concepts

Continuous Hopfield Networks (CHNs) are a class of recurrent neural networks composed of a single layer of n fully connected neurons. Each neuron possesses an activation function and interacts with others through a symmetric connection matrix. The dynamic behavior of CHNs is governed by a differential equation that describes the temporal evolution of the state vector u of the network [6]:

$$\frac{d\,u}{d\,t} = T\,v + I \quad (1)$$

Here, $T \in \mathbb{R}^{\mathrm{nxn}}$ denotes the synaptic connection (or weight) matrix encoding the strength of interactions among neurons, $I \in \mathbb{R}^{\mathrm{n}}$ is the external bias vector, $u \in \mathbb{R}^{\mathrm{n}}$ is the state vector of the neurons, and $v \in \mathbb{R}^{\mathrm{n}}$ is the activation vector, typically related to the states through a nonlinear activation function $v = g(u)$. A commonly used activation function is the hyperbolic tangent, bounded between 0 and 1, and capable of capturing graded neuron response with adjustable thresholding.

A vector u^{e} is defined as an equilibrium (or steady-state) point of the CHN if, for any initial state u^0, the network settles into $u(t) = u^{\mathrm{e}}\,\forall\, t \geq t_{\mathrm{e}}$, for some finite $t_{\mathrm{e}} \geq 0$. The convergence of the network towards such a state is evaluated using a Lyapunov energy function, which ensures global asymptotic stability if it is monotonically decreasing over time. The energy function, E_{Lyap}

is defined as follows [6]:

$$E_{Lyap}(v) = -\left(\frac{1}{2}\right) v^T T\,v - I^T v \quad (2)$$

Hopfield demonstrated that if the weight matrix T is symmetric and the activation functions are monotonic and continuous, then the energy function E_{lyap} always decreases over time $\frac{dE_{Lyap}}{dt} \leq 0$, ensuring convergence to a stable state. This principle allows CHNs to be employed as dynamic solvers for constrained optimization problems.

To solve such problems, an energy function is typically constructed to encode both the objective function and the constraints of the problem. If f(v) represents the objective function and $\mathrm{g_i(v)}$ the set of constraints, the generalized energy formulation is [6,7,8,9]:

$$E(v) = \alpha\, f(v) + \sum_i \varphi_i g_i(v) + \sum_{i,j} \varphi_{i,j}\, g_i(v) g_j(\mathrm{v}) \quad (3)$$

Here, $\alpha, \varphi_{\mathrm{i}}, and\ \ \varphi_{i,j}$ re penalty coefficients that enforce the constraints and guide the system toward feasible and optimal solutions. The choice of these parameters significantly affects convergence behavior and solution quality. Gradient-based updates derived from this energy

Received: Aug. 08, 2025

formulation help guide the network's trajectory within the feasible search space, while avoiding invalid or unstable states.

Challenges in Practical Applications:

- **Sensitivity to Noise:** In real applications such as digital communication or smart grid control, input data are often corrupted by noise or unpredictable fluctuations [13]. Rigid CHNs cannot adapt their dynamics to such uncertainty.

- **Limited Adaptability:** Static weights and biases limit the ability of CHNs to respond to nonstationary or time-varying conditions. As a result, the network may converge prematurely or fail to reach global optima.

- **Convergence to Local Minima:** CHNs inherently settle into local minima of the energy landscape, which may not correspond to globally optimal solutions. Without adaptive mechanisms, this restricts their performance in complex or non-convex problem spaces.

These limitations motivate the integration of more flexible learning mechanisms. In this work, we address these issues by extending the classical CHN with fractional-order dynamics and fuzzy logic-based adaptation. The resulting Fuz-Frac-CHN Fractional CHN enhances memory modeling through fractional calculus and achieves resilience to uncertainty via fuzzy logic control. This dual enhancement significantly improves convergence stability, cost efficiency, and robustness in optimization tasks, as demonstrated in the Economic Dispatch case study.

1.2 Fuzzification: A Solution to Uncertainty and Rigidity in Fractional CHNs

Conventional Continuous Hopfield Networks (CHNs) are often challenged by their sensitivity to input disturbances, rigid parameterization, and lack of adaptability under uncertainty. To address these limitations—particularly in dynamic, noisy, or nonstationary optimization environments—we propose a fuzzification strategy that integrates fuzzy logic into the fractional-order CHN framework. This fusion enables the network to adjust its internal dynamics in response to real-time performance feedback, such as power mismatch and cost deviation, thereby enhancing robustness and flexibility.
Fuzzy logic, first introduced by Zadeh [14], is well-suited for modeling systems characterized by uncertainty and imprecision. Unlike traditional binary logic systems, fuzzy logic supports smooth transitions between states, allowing the system to reason with partial truths and make adaptive decisions. When integrated into neural networks—commonly referred to as fuzzy neural networks (FNNs)—this approach has shown significant improvements in learning efficiency, scalability, and noise tolerance [16, 17, 18, 19].
In our proposed Fuz-Frac-CHN, we extend this paradigm by embedding a fuzzy inference system (FIS) that dynamically modulates the Grünwald–Letnikov (GL) coefficients based on system-level mismatch and performance signals. This adaptive mechanism fine-tunes the fractional memory effect during optimization, enabling the network to intelligently balance long-term memory (fractionality) and responsiveness (fuzziness).

Key Motivations and Benefits:

- Robustness to Noise and Uncertainty: Classical CHNs and fixed-order fractional CHNs are susceptible to input noise and parameter fluctuations. Our fuzzy adaptation mitigates this sensitivity by smoothing the response to disturbances and enabling error-tolerant optimization dynamics.

- Dynamic Parameter Tuning: Fixed weights and biases in conventional CHNs hinder adaptability. The fuzzy system continuously adjusts the GL memory weights in response to mismatch levels, making the network more responsive to evolving optimization states.

- Avoidance of Local Minima: The progressive adjustment of GL coefficients through fuzzy rules prevents premature convergence and enhances the exploration capability of the network in complex cost landscapes.
- Improved Convergence and Cost Efficiency: The Fuz-Frac-CHN achieves lower or comparable cost values at lower mismatch levels, especially at small fractional orders ($\alpha < 0.3$), outperforming classical and fixed-order models.
- Memory-Aware Learning Control: The system uses fuzzy rules based on cost deviation and power imbalance, ensuring smoother and more stable convergence (see Figures 3 and 5).

Overall, this fuzzification strategy enhances CHN by introducing memory awareness and robustness to uncertainty. It overcomes the limitations of classical models and exploits fractional dynamics in an adaptive way. The proposed Fuz-Frac-CHN is therefore an effective neurodynamic approach for solving complex constrained optimization problems, such as Economic Dispatch.

1.3 Contributions

The main contributions of this work can be summarized as follows:
- Fractional-Order Hopfield Neural Network: Extension of the classical CHN by incorporating fractional derivatives to capture long-term memory and hereditary system properties.

- Fuzzification of Grünwald–Letnikov Coefficients: Introduction of fuzzy-adjusted GL coefficients to reduce computational complexity while retaining adaptive memory contributions.
- FCHN Design for Economic Dispatch: Construction of a fractional continuous Hopfield network tailored to the ED problem.
- Fuzzy System Integration: Design of membership functions and fuzzy rules to adapt GL coefficients.

- Efficient Optimization: The proposed fuzzy FCHN ensures robust convergence, reduced computational cost, and improved economic efficiency when solving the ED problem under dynamic and uncertain conditions.
This work represents a significant advancement in integrating fractional calculus and fuzzy logic within Hopfield networks, enabling more intelligent, memory-aware, and

Received: Aug. 08, 2025

computationally efficient models. By fuzzifying the Grünwald–Letnikov coefficients and incorporating them into a fractional continuous Hopfield network, the proposed framework provides robust, flexible, and noise-resistant solutions to the Economic Dispatch problem.

The paper is organized as follows: Section 2 introduces the Economic Dispatch problem, Section 3 presents the fuzzy inference system, Section 4 reports experimental results, and Section 5 concludes the study.

2. Fractional-Order Continuous Hopfield Neural Network (FCHN)

The Improved Fractional Continuous Hopfield Neural Network (IFCHN) extends the classical Hopfield model by incorporating memory effects and non-local dynamics through fractional calculus. The network consists of a single layer of n fully interconnected neurons, each governed by a nonlinear activation function. Unlike integer-order systems, the IFCHN uses fractional derivatives to capture long-term dependencies and hereditary properties, making it suitable for modeling noisy, time-varying, and uncertain systems. Consequently, the IFCHN is well-suited for robust optimization tasks.

2.1 Presentation of Fractional CHN

The dynamic behavior of the Fractional Continuous Hopfield Network (FCHN) can be rigorously characterized through the framework of fractional calculus, which extends classical differentiation to non-integer orders and thereby introduces memory and hereditary properties into dynamical systems [6]. In this context, the evolution of the network states is governed by the following fractional differential equation:

$$CD_t^q u(t) = T\, v(t) + I \quad (4)$$

where $\mathrm{CD}_\mathrm{t}^\mathrm{q}$ denotes the Caputo fractional derivative of order $q \in (0,1), u(t)$ is the neuron state vector, $v(t) = g(u(t))$ is the vector of neuron activations through a nonlinear function $g(.)$ (typically sigmoid or hyperbolic tangent), T is the synaptic weight matrix, and I represents the external input or bias vector.
This equation generalizes the classical CHN by incorporating fractional-order derivatives, which introduce memory effects, enabling the network to retain information from past states. This long-term memory property enhances the network's ability to model dynamical systems with hereditary characteristics. Several definitions of fractional derivatives exist in the literature, each with distinct mathematical properties and applications:

- **Riemann–Liouville fractional derivative:**

$$RLD_t^q f(t) = \frac{1}{\Gamma(n - q)} \frac{d^n}{dt^n} \int_0^t \left(\frac{f(\tau)}{(t - \tau)^{q - n + 1}} \right) d\tau, \qquad \mathrm{n} - 1 < \mathrm{q} < \mathrm{n} \quad (5)$$

This formulation is well-suited for theoretical analysis and problems defined with initial conditions expressed in terms of fractional integrals. However, it is less practical in physical systems since it requires fractional-order initial conditions [20].

• Caputo Fractional Derivative:

$$CD_t^q f(t) = \frac{1}{\Gamma(n - q)} \int_0^t \frac{f^{(n)}(\tau)}{(t - \tau)^{q - n + 1}} d\tau, \qquad n - 1 < q < n \quad (6)$$

Unlike the Riemann–Liouville form, the Caputo derivative uses standard integer-order initial conditions, making it widely preferred in engineering, physics, and control applications [21]. This is the operator adopted in (4).

• Grünwald–Letnikov Fractional Derivative:

$$GL\, D_t^q\, f(t) = lim_{h \to 0} \frac{1}{h^q} \sum_{k=0}^{\lfloor t/h \rfloor} (-1)^k \binom{q}{k} f(t - k\,h), \quad (7)$$

This definition is particularly advantageous for numerical implementations, since it directly extends the finite difference approximation of derivatives [28]. It is often used for simulating fractional-order neural networks and optimization problems.

• Hadamard Fractional Derivative:

$$HD_t^q f(t) = \frac{1}{\Gamma(n - q)} t^{q - n} \left(\frac{d}{dt}\right)^n \int_1^t \left(ln\left(\frac{t}{\tau}\right)\right)^{n - q - 1} f(\tau) \frac{d\tau}{\tau}, \qquad (8)$$

The Hadamard derivative is useful for problems with logarithmic-type scaling, commonly encountered in models with multiplicative processes or scale-invariant properties [22].

In summary, the choice of fractional derivative depends on the modeling context: the Caputo derivative is ideal for real-world systems requiring classical initial conditions; the Riemann-Liouville derivative is more suitable for theoretical exploration; the Grünwald–Letnikov approach is favored for numerical simulations; and the Hadamard derivative finds applications in scale-invariant or logarithmic-growth models.

2.2 Approximation Methods for Fractional Derivatives

Simulating the dynamic equation (4) requires numerical approximations for fractional derivatives. Several schemes are commonly used:

- Grünwald–Letnikov (GL) Approximation: Using the Grünwald-Letnikov (GL) definition of the fractional derivative, the IFCHN dynamics can be expressed in a discretized form as follows:

$$GLD_t^q u(t_n) \approx \frac{1}{h^q} \sum_{k=0}^{n} (-1)^k \binom{q}{k} u(t_{n-k}) = T\, v(t_n) + I, \quad (9)$$

This provides a practical numerical approximation of fractional-order dynamics, suitable for simulations and implementations. It is intuitive and easy to implement, but memory requirements grow with time steps, increasing computational cost [21].

- Riemann–Liouville (RL) Approximation:

Using the Riemann-Liouville (RL) definition of the fractional derivative, the IFCHN dynamics can be expressed as follows:

$$RLD_t^q u(t_n) \approx \frac{1}{\Gamma(1-q)} \frac{d}{dt} \sum_{k=0}^{n-1} \frac{u(t_k)}{(t_n - t_k)^q} = T\, v(t_n) + I, \qquad (10)$$

This highlights the memory property through dependence on past states $u(t_k)$, making it suitable for theoretical analysis. It provides high accuracy but requires non-local initial conditions [22].

- Caputo–Grünwald (CG) Approximation:

Using the Caputo definition of the fractional derivative, the IFCHN dynamics can be expressed in a discrete-time form as :

$$CD_t^q u(t_n) \approx \left(\frac{1}{h^q}\right) \sum_{k=0}^{n} w_k^{(q)} \big(u(t_{n-k}) - u(0)\big) = T\, v(t_n) + I, (11)$$

where the coefficients $w_k^{(q)}$ act as fractional weights that incorporate past states relative to the initial condition $u(0)$.This formulation highlights the practical advantage of the Caputo definition in numerical simulations, as it allows initial conditions to be treated similarly to classical integer-order systems while still capturing long-term memory effects.

The CG approach combines the advantages of both Caputo and Grünwald-Letnikov definitions, enabling the use of standard initial conditions while preserving memory effects, making it suitable for engineering applications [23].

- Fractional Adams–Bashforth–Moulton (ABM) Method:

The ABM scheme provides a predictor–corrector formulation of the FCHN dynamics.

$$u(t_{n+1}) = u(0) + \frac{1}{\Gamma(q)} \sum_{j=0}^{n} b_{j,n+1} \big(T\, v(t_j) + I\big), \quad (12)$$

$$b_{(j,n+1)} = h^q [(n+1-j)^{q-1} - (n-j)^{q-1}].$$

where the coefficients $b_{(j,n+1)}$ weight the contribution of all previous states to the next step. This method provides high accuracy and numerical stability by incorporating memory effects systematically, making it particularly suitable for simulations requiring precise long-term behavior of fractional-order Hopfield networks. The ABM predictor-corrector method achieves high accuracy and stability, although it is computationally intensive due to the memory of past states [24].

The fractional derivative in IFCHN can be approximated using several numerical schemes, each with its advantages and limitations. The Grünwald-Letnikov (GL) method is simple and intuitive, but it is memory-intensive and offers only moderate accuracy. The Riemann-Liouville (RL) approach provides high accuracy; however, it is constrained by the need for non-standard initial conditions, which can limit its practical use. The Caputo–Grünwald (CG) approximation offers a balanced compromise, delivering reasonable accuracy while remaining straightforward to implement with standard initial values. Finally, the Fractional Adams–Bashforth–Moulton (ABM) method achieves high accuracy and numerical stability, although at the cost of increased computational demand due to the involvement of all past states.

3. Fuzzified Grünwald–Letnikov Approximation for Standard Fractional CHN

The classical Grünwald–Letnikov (GL) derivative is expressed as [21]:

$$GLD_t^q u(t_n) \approx \frac{1}{h^q} \sum_{k=0}^{n} (-1)^k \binom{q}{k} u(t_{n-k}) \quad (13)$$

where $q \in (0,1)$ is the fractional order, h is the time step, and $u(t_n)$ is the neuron state vector. This exact computation can be computationally intensive, especially for long histories. To mitigate this, we introduce fuzzy-adjusted coefficients $\widetilde{c_k}$ to modulate the memory contribution:

$$\widetilde{c_k} = w_k \cdot (-1)^k \binom{q}{k} \quad (14)$$

where $w_k \in [0,1]$ is a fuzzy weight determined adaptively. The fuzzified fractional derivative then becomes:

$$FGLD_t^q u(t_n) \approx \frac{1}{h^q} \sum_{k=0}^{n} \widetilde{c_k} u(t_{n-k}) = Tv(t_n) + I \quad (15)$$

3.1 Motivation for Fuzzification of Coefficients

The classical GL coefficients $(-1)^k \binom{q}{k}$ precisely encode the memory effect of fractional derivatives. However, two main challenges arise in practical implementation:

High Computational Complexity: Each new time step requires summing over all past states, which grows linearly with n. For large-scale neural networks or long simulations, this becomes prohibitive.

Sensitivity to Dynamic Changes: The GL coefficients are fixed and do not adapt to the current network state. In dynamic or uncertain environments, past contributions may overly influence current updates, potentially causing oscillations or slow convergence.

Fuzzification addresses both issues by introducing adaptive weights w_k that scale the classical coefficients according to the network's current state and performance. By doing so:

- Memory contributions are reduced when large deviations or instability are detected, preventing overreaction from distant past states.
- Computational complexity is effectively controlled, as the fuzzy weights allow selective emphasis on significant past states while downweighting less relevant ones.
- The network becomes more robust to uncertainties, noise, or parameter variations, enhancing convergence speed and stability in practical applications.

3.2 Fuzzy System Design

To compute w_k, two performance-based indicators are defined:

$$E_{(t_n)} = \|u(t_n) - u(t_{n-1})\|, \quad \Delta_{(t_n)} = \|v(t_n) - v(t_{n-1})\| \quad (16)$$

Received: Aug. 08, 2025

Where $E(t_n)$ measures the state variation of neurons and $\Delta(t_n)$ measures the activation variation. Large deviations indicate instability or slow convergence, prompting stronger adjustment of memory weights.

3.3 Membership Functions

Three fuzzy linguistic terms are assigned for both inputs: Low (L), Medium (M), and High (H). Gaussian membership functions are used:

$$\mu_L(x) = exp\left(-\frac{(x)^2}{2{\sigma_L}^2}\right)$$

$$\mu_M(x) = exp\left(-\frac{(x - x_{med})^2}{2{\sigma_M}^2}\right) \qquad (17)$$

$$\mu_H(x) = exp\left(-\frac{(x - x_{max})^2}{2{\sigma_H}^2}\right)$$

3.4 Fuzzy Rules

The fuzzy inference rules define how w_k changes according to $E(t_n)$ and $\Delta(t_n)$

- If E is Low and Δ is Low, then $w_k = 1.0$ (full memory contribution)
- If E is Medium or Δ is Medium, then $w_k = 0.8$ (moderate memory)
- If E is High or Δ is High, then $w_k = 0.6$ (reduced memory)

3.5 Fuzzification and Defuzzification

Fuzzification converts the crisp inputs into fuzzy membership degrees:

$$\mu_L(E),\ \mu_M(E),\ \mu_H(E), \mu_L(\Delta),\ \mu_M(\Delta),\ \mu_H(\Delta)$$

Defuzzification generates a crisp weight using the weighted average method:

$$w_k = \frac{\sum_{i,j} \mu_i(E) \cdot \mu_j(\Delta) \cdot w_{i,j}}{\sum_{i,j} \mu_i(E) \cdot \mu_j(\Delta)} \quad (18)$$

where $w_{i,j}$ corresponds to the rule-based weight for the combination of linguistic levels i and j.

3.6 Integration into Fractional CHN

Finally, the fuzzified GL derivative is incorporated into the standard Fractional CHN dynamics:

$$FGLD_t^q u(t_n) = Tv(t_n) + I \quad (19)$$

with the fuzzy-adjusted coefficients:

$$\tilde{c}_k = w_k \cdot (-1)^k \binom{q}{k}, w_k \in [0.6,1.0] \quad (20)$$

This fuzzification improves computational efficiency, enhances robustness, and allows the network to dynamically adapt memory contributions to the current state, which is essential

Received: Aug. 08, 2025

for practical implementations of fractional Hopfield networks.

4. Economic Dispatch: Formulation and Fuzzy Grunwald–Letnikov Fractional Continuous Hopfield Network Approach

In this section, we first present the fundamentals of the Economic Dispatch (ED) problem, a central optimization task in modern power and energy systems aimed at allocating generation among available power units to meet demand at the minimum possible cost while satisfying operational and system constraints. We then discuss classical and advanced solution methodologies, outlining their strengths and limitations in handling the nonlinear, nonconvex, and dynamic nature of ED. Special attention is given to the integration of intelligent and soft computing techniques, particularly fuzzy logic and fractional calculus, which offer enhanced adaptability and robustness in uncertain environments. To establish the theoretical foundation, we provide a detailed mathematical formulation of the ED problem. Finally, we introduce the Fuzzy Grunwald–Letnikov (GL) based Fractional Continuous Hopfield Network (CHN) as a novel framework for optimal dispatching, bridging conventional problem formulation with advanced fractional-order neural optimization.

4.1 Introduction to Economic Dispatch

Economic dispatching (ED) has emerged as a central optimization problem in advanced energy scheduling networks. It consists in determining the optimum generation of electricity by central power plants at the minimum possible price, subject to certain system requirements. The idea is to determine the volumes produced by individual power plants that minimize the overall cost of electricity production, while meeting both demand and the technological restrictions of the power system.

The ED issue could be represented mathematically as an optimization problem under specific constraints, in such a way that the objective function denotes the total cost of the plant, and the constraints reflect the workload demands and running restrictions of the plant's generators.

Solution Methods for Economic Dispatch: Over the years, numerous approaches have been proposed to address the Economic Dispatch (ED) problem, which can broadly be categorized into two main families: conventional optimization methods and advanced intelligent techniques. Conventional methods include linear programming (LP), quadratic programming (QP), and nonlinear programming (NLP), whereas advanced techniques encompass evolutionary algorithms (EA) and swarm intelligence (SI).

LP-based methods approach the ED problem by linearizing the cost function and applying optimization algorithms. However, this approach is only applicable to systems characterized by linear cost functions [32]. In practical power systems, many generation units are modeled with quadratic cost functions, which naturally lead to a quadratic programming formulation. Consequently, QP approaches are often considered more realistic and applicable than LP-based methods [33].

On the other hand, evolutionary algorithms—such as Genetic Algorithms (GA), Differential

Received: Aug. 08, 2025

Evolution (DE), and Particle Swarm Optimization (PSO)—have gained significant attention for solving the ED problem, especially in cases involving complex, nonlinear cost functions and large-scale energy systems [34]. Similarly, swarm intelligence techniques, including Ant Colony Optimization (ACO) and PSO, have been widely adopted for their robustness and ability to provide efficient solutions in highly nonlinear and multimodal optimization landscapes [35].

Challenges in Economic Dispatch: Despite significant progress in optimization methodologies, the Economic Dispatch (ED) problem continues to present substantial challenges. A major difficulty arises from the non-convex nature of generator cost functions, which often produce multiple local minima. This characteristic makes it difficult for classical optimization techniques to consistently achieve global optima, frequently resulting in sub-optimal solutions.

Another critical issue is the time-varying nature of electricity demand, as load requirements fluctuate throughout the day. Coupled with predictive uncertainties, this variability can lead to inaccurate or inefficient dispatch allocation if not properly accounted for. Consequently, incorporating uncertainty into the planning and optimization process is essential for improving system reliability [36].

Furthermore, the increasing penetration of renewable energy sources, such as wind and solar, introduces additional layers of variability and intermittency into power generation. These stochastic characteristics of renewable production exacerbate the complexity of ED, requiring optimization frameworks that can explicitly address randomness and uncertainty in energy supply [37].

Recent Developments and Applications: Recent research efforts have increasingly focused on hybridizing different optimization methods to improve the effectiveness and efficiency of Economic Dispatch (ED) solutions. For example, composite approaches that integrate Particle Swarm Optimization (PSO) with Simulated Annealing (SA) or Genetic Algorithms (GA) have demonstrated superior performance in terms of global search capability and convergence quality [38]. In parallel, the growing uncertainty associated with renewable energy generation has motivated the development of robust optimization frameworks, which ensure that system performance remains near-optimal even under fluctuating or unpredictable operating conditions [39]. Along similar lines, advanced robust strategies have been introduced to guarantee optimality and reliability of dispatch solutions despite the inherent variability of renewable sources [40].

The ED problem thus continues to represent a core challenge in power network optimization. While conventional approaches such as linear programming and quadratic programming have long been applied, their effectiveness diminishes when confronted with large-scale, nonlinear, and stochastic systems. Modern methods, including evolutionary algorithms and swarm intelligence, have proven more suitable for addressing such complexities in real-world energy networks. Moreover, with the rapid integration of renewable energy technologies, hybrid and robust optimization schemes are gaining significant traction. As power systems evolve toward intelligent and distributed grids, distributed optimization techniques are expected to play an increasingly central role in ensuring both cost-effective and sustainable electricity generation.

4.2 Problem Formulation

Economic dispatching (ED) attempts to dispatch energy generation among generators in such a way as to meet a specified power need P_d with the minimum total cost. The total cost function is expressed as a quadratic function:

$$C(P)=\sum_{i=1}^{N} c_i P_i^2, \quad (21)$$

where:

- P_i represents the energy produced by generator i
- c_i represents the price ratio of generator i
- N represents number of power units

In addition, there are power constraints:

$$P_{\min_i} \leq P_i \leq P_{\max_i}, \quad \sum_{i=1}^{N} P_i = P_d. (22)$$

That means we guarantee that every generator operates properly up to its capacity, while satisfying customer demands.

4.3 Fuzzy GL to Fractional Continuous Hopfield Network for Economic Dispatch

In this section, we design an appropriate Continuous Hopfield Neural Network (CHN) for solving the Economic Dispatch (ED) problem. We then introduce a fuzzy-enhanced version of the CHN to address the limitations of the classical ED-CHN, improving its adaptability and performance in complex scenarios.

4.3.1 Fractional-Order Continuous Hopfield Network (FCHN) Construction for Economic Dispatch (ED)

To address the Economic Dispatch (ED) problem using a Fractional Continuous Hopfield Network (FCHN), we formulate an energy function whose minimum corresponds to the optimal solution of the ED problem. The energy function of the FCHN is defined as:

$$E=\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} w_{ij} V_i V_j - \sum_{i=1}^{N} \theta_i V_i, (23)$$

Where V_i is the output of neuron i, w_{ij} are the synaptic connection weights between neurons i and j, and θ_i are the bias terms. These parameters are carefully constructed to encode both the objective function of the ED and the necessary equality constraint.

To enforce the power balance constraint in ED, a penalty term is introduced:

$$E_{\text{penalty}} = \frac{\lambda}{2}\left(\sum_{i=1}^{N} P_i - P_D\right)^2, (24)$$

where P_i is the power output of generator i, P_D is the total power demand, and λ is a penalty coefficient.

The neural dynamics are now modeled using a fractional-order differential equation to capture memory effects and better reflect system uncertainties. The generalized dynamics for neuron i are:

$$CD_t^q U_i(t) = -U_i(t) + \sum_{j=1}^{N} w_{ij} V_j(t) - \theta_i, \quad (25)$$

where CD_t^q is the Caputo fractional derivative of order $q \in (0,1)$,and $U_i(t)$ is the internal potential of neuron i.

The neuron output is obtained through a sigmoid activation function:

$$V_i(t) = \frac{1}{1 + e^{-\alpha U_i(t)}}, (26)$$

and the actual power output is mapped linearly as:

$$P_i = P_i^{min} + V_i(t) \cdot \left(P_i^{max} - P_i^{min}\right)(27)$$

4.3.2 Fuzzified GL Coefficients in Fractional CHN for Economic Dispatch

The fractional continuous Hopfield dynamics for generator i in Economic Dispatch (ED) can be written as:

$$CD_t^q U_i(t) = -U_i(t) + \sum_{j=1}^{N} w_{ij} V_j(t) - \theta_i, (28)$$

where $U_i(t)$ is the neuron state associated with generator i, $V_j(t) = g\left(U_j(t)\right)$ is the output of neuron j, w_{ij} are the synaptic weights, and θ_i is a bias related to the load demand.

Motivation for Fuzzifying GL Coefficients

In ED applications, the classical Grünwald–Letnikov derivative requires summing over all past states:

$$GLD_t^q U_i(t_n) \approx \frac{1}{h^q} \sum_{k=0}^{n} (-1)^k \binom{q}{k} U_i(t_{n-k})(29)$$

Two challenges arise:

- **High Computational Cost:** Summing over all previous states for each time step is expensivze for multi-generator systems.

- **Overweighting Past States:** Fixed coefficients may cause slow convergence or instability if earlier states are overly influential.

To address these issues, fuzzy-adjusted weights $w_k \in [0,1]$ are introduced:

$$\tilde{c_k} = w_k\,(-1)^k \binom{q}{k}, (30)$$

$$FGLD_t^q U_i(t_n) \approx \frac{1}{h^q} \sum_{k=0}^{n} \tilde{c_k}\; U_i(t_{n-k})$$

Fuzzy Inputs for Economic Dispatch

$$\Delta P(t_n) = \left|\sum_{i=1}^{N} P_i(t_n) - P_d\right| \qquad (31)$$

$$\Delta C(t_n) = \max\{0,\ C(t_n) - C_{\text{target}}\}$$

where ΔP measures power mismatch and ΔC quantifies excess cost. Larger deviations trigger stronger reduction in past memory contributions.

Membership Functions

Gaussian membership functions map the deviations to linguistic terms (Low, Medium, High):

$$\mu_L(\Delta P) = \exp\left(-\frac{(\Delta P)^2}{2\sigma_L^2}\right), \qquad (32)$$

$$\mu_M(\Delta P) = \exp\left(-\frac{(\Delta P - P_{\text{med}})^2}{2\sigma_M^2}\right), (33)$$

$$\mu_H(\Delta P) = \exp\left(-\frac{(\Delta P - P_{\max})^2}{2\sigma_H^2}\right), (34)$$

$$\mu_L(\Delta C) = \exp\left(-\frac{(\Delta C)^2}{2\sigma_L^2}\right), (35)$$

$$\mu_M(\Delta C) = \exp\left(-\frac{(\Delta C - C_{\text{med}})^2}{2\sigma_M^2}\right), (36)$$

$$\mu_H(\Delta C) = exp\left(-\frac{(\Delta C - C_{\max})^2}{2\sigma_H^2}\right). (37)$$

The choice of Gaussian membership functions is motivated by their smoothness and differentiability, which ensure gradual transitions between linguistic terms and avoid abrupt changes in the control surface. This property is particularly advantageous in optimization problems such as ED, where stability and robustness are critical. The parameters $\sigma_L, \sigma_M, \sigma_H$ govern the spread of each function and thus directly influence the sensitivity of the fuzzy inference system to deviations in power mismatch ΔP and cost variation ΔC. Narrower spreads yield sharper distinctions between linguistic terms, improving precision but potentially reducing robustness to noise, whereas wider spreads enhance tolerance to uncertainty at the cost of reduced sensitivity. Similarly, the centers $P_{\mathrm{med}}, P_{\mathrm{max}}, C_{\mathrm{med}}, C_{\mathrm{max}}$ define the anchor points for medium and high levels, and must be carefully tuned to reflect realistic system operating ranges. Hence, the proper selection of both the type and the parameters of membership functions is crucial, as it directly impacts the accuracy, adaptability, and overall performance of the fuzzy-controlled fractional CHN in solving the economic dispatch problem.

Fuzzy Rules

To translate the qualitative reasoning into a computational framework, a set of fuzzy rules is formulated.

These rules establish how the deviations in power mismatch ΔP and cost ΔC influence the adaptive memory weight w_k of the network.

- If ΔP is Low and ΔC is Low, then w_k = 1.0 (full memory contribution)
- If ΔP or ΔC is Medium, then w_k = 0.8 (moderate memory)
- If ΔP or ΔC is High, then w_k = 0.6 (reduced memory)

The formulation of these fuzzy rules reflects a balance between system reliability and adaptability. When both the power mismatch ΔP and the cost deviation ΔC are small, the system is already operating close to its optimal point, and thus a high memory contribution (w_k = 1.0) is maintained to reinforce stability. In cases where either ΔP or ΔC reaches medium levels, a moderate memory contribution (w_k = 0.8) is !applied to allow for controlled adaptation while avoiding abrupt changes in the network dynamics. Finally, when either deviation becomes high, a reduced memory contribution (w_k = 0.6) is enforced, prioritizing fast corrective actions and preventing the system from being trapped in inefficient or unstable states. This design illustrates the idea that fuzzy rules are not arbitrarily defined, but rather constructed to capture expert reasoning: higher deviations demand more flexibility and adaptation, while near-optimal states require greater stability and reinforcement of past learning.

Defuzzification

The crisp fuzzy weight is obtained using a weighted average defuzzification strategy, which ensures a smooth and continuous mapping from the fuzzy rule base to the final control action:

$$w_k = \frac{\sum_{i,j} \mu_i(\Delta P) \cdot \mu_j(\Delta C) \cdot w_{i,j}}{\sum_{i,j} \mu_i(\Delta P) \cdot \mu_j(\Delta C)}, (38)$$

where $w_{i,j}$ denotes the weight associated with the fuzzy rule that combines the linguistic levels i of ΔP and j of ΔC.

This formulation guarantees that the contribution of each rule is proportional to the degree of membership of the current state, thereby blending the influence of multiple rules in a coherent manner.

The weighted average approach is computationally efficient, avoids discontinuities, and provides a natural compromise between competing rules, which is particularly desirable in real-time applications such as economic dispatch where stability, robustness, and smooth adaptability are essential.

Integration into ED Fractional CHN

The fuzzified GL derivative is incorporated into the neuron dynamics:

$$FGLD_t^q U_i(t_n) = -U_i(t_n) + \sum_{j=1}^{N} w_{ij} V_j(t_n) - \theta_i, (39)$$

where the coefficients in the derivative term are scaled by the adaptive fuzzy weights:

$$\tilde{c_k} = w_k \cdot (-1)^k \binom{q}{k} (40)$$

This approach ensures adaptive memory contribution, reduces computational cost, and enhances stability and convergence for the Economic Dispatch problem.

Experimentation Setup and Problem Description

This experimentation investigates the application of a **Intuitionistic Fractional Continuous Hopfield Network (FCHN)** to **the economic dispatch problem**, where three generators must supply a fixed power demand while minimizing total generation cost. Fractional-order dynamics allow the system to incorporate memory effects and improve convergence behavior.

5.1 Problem Definition

The economic dispatch problem consists of:

- Allocating generation among three units to meet a fixed total demand of 450 MW.
- Ensuring that each generator operates within its minimum and maximum limits.
- Minimizing the total generation cost while applying penalties if constraints are violated or if the cost exceeds a target threshold.

5.2 Generators and Cost Characteristics

Table 1 summarizes the operational limits and cost coefficients of the three generators considered in this economic dispatch problem. Generator G2 has the largest power range, from 150 MW to 400 MW, allowing it to cover a significant portion of the total demand. Generator G1 operates within a moderate range of 100 MW to 300 MW, while G3 has the smallest capacity, ranging from 50 MW to 200 MW, making it suitable for fine adjustments or smaller demand increments.

In terms of cost efficiency, G2 is the most economical generator, with the lowest cost coefficient of 0.0015. G1 is slightly more expensive at 0.002, while G3 is the most costly at 0.003. These differences indicate that minimizing the total generation cost requires prioritizing G2 and G1, while using G3 only when necessary.

The combination of capacity and cost characteristics creates a trade-off in the dispatch strategy. Although lower-cost generators should be utilized as much as possible, their maximum output limits must be respected. Despite its higher cost, G3 provides additional flexibility to meet total demand when other generators operate near their limits. Understanding these characteristics is essential for interpreting simulation results, as they directly influence power allocation, total generation, and cost evolution in the FCHN-based approach.

Table 1: Operational limits and cost coefficients of the generators

Generator	Power limits (MW)	Cost coefficient
G1	$100 \leq P1 \leq 300$	0.002
G2	$150 \leq P2 \leq 400$	0.0015
G3	$50 \leq P3 \leq 200$	0.003

5.3 FCHN Configuration and Simulation Parameters

The Fractional Continuous Hopfield Neural Network (FCHN) used in this study is configured to simulate the economic dispatch problem over a horizon of 200 time steps. The network models three generators, and ten different fractional orders, ranging from 0.1 to 1.0, are tested to evaluate the effect of fractional dynamics on convergence and cost minimization.

The step size for state updates is set to 0.05, and the slope of the sigmoid activation function is fixed at 1.0, ensuring smooth and stable output behavior. The weight matrix is scaled according to the generator cost coefficients to promote total cost reduction, while the bias vector is adjusted to maintain the total power output close to the required demand.

To enforce constraints, a penalty of 100 is applied for any mismatch between total generation and demand. Additionally, a cost penalty of 10^{-4} is introduced when the total generation cost exceeds 100,000 units.

Neuron states are randomly initialized at the beginning of each simulation, allowing the FCHN to evolve dynamically under different fractional orders. These parameters define the overall network configuration and ensure that the simulations accurately capture both economic and operational aspects of the dispatch problem.

5.4 Simulation Procedure

The simulation of the Improved Fractional Continuous Hopfield Network (IFCHN) follows a structured procedure. Initially, the neuron states, outputs, and generator powers are set up, providing the starting conditions for the network dynamics. At each time step, the neuron states are updated according to the fractional-order dynamics, which incorporate memory effects from past states.

Following the state update, the generator outputs are computed using the sigmoid activation function, ensuring they remain within their respective operational limits. The total generation and associated cost are then evaluated at each time step, accounting for both the generation cost and any penalties. Penalties are applied if the total generation deviates from the fixed demand or if the cost exceeds the target threshold of 100,000 units.

This process is repeated for each fractional order being tested, and the resulting outputs, total generation, and cost histories are stored for subsequent analysis. This approach allows a comprehensive assessment of how different fractional orders affect network convergence, generator allocation, and cost minimization.

The main objective of this study is to evaluate the performance of the **Intuitionistic Fractional Continuous Hopfield Network (IFCHN)** in solving the economic dispatch problem. In particular, the study investigates how the fractional order influences network convergence and the allocation of power among generators. Ensuring that the total generation consistently meets the fixed demand is a key requirement, while maintaining the total cost below the predefined threshold is equally important.

The simulations also aim to provide clear visualizations of the generator outputs, the total generation relative to the demand, and the evolution of the cost function for different fractional orders. By analyzing these results, it becomes possible to identify the fractional orders that offer an optimal balance between convergence speed and economic efficiency, thereby enhancing the decision-making process for power dispatch in constrained and cost-sensitive scenarios.

5.5 Economic Dispatch Optimization with Varying Generator Numbers

This subsection investigates the performance of Classical, Fractional, and Fuz-Frac-CHN-based optimization methods in solving the economic dispatch (ED) problem under varying fractional orders α. By analyzing cost, demand satisfaction, convergence behavior, and power mismatch, the impact of adaptive fractional dynamics on optimization performance is highlighted.

In particular, the Fuz-Frac-CHN method demonstrates improved adaptability by dynamically adjusting the fractional order based on system mismatch, resulting in competitive or superior performance compared to fixed-order approaches.

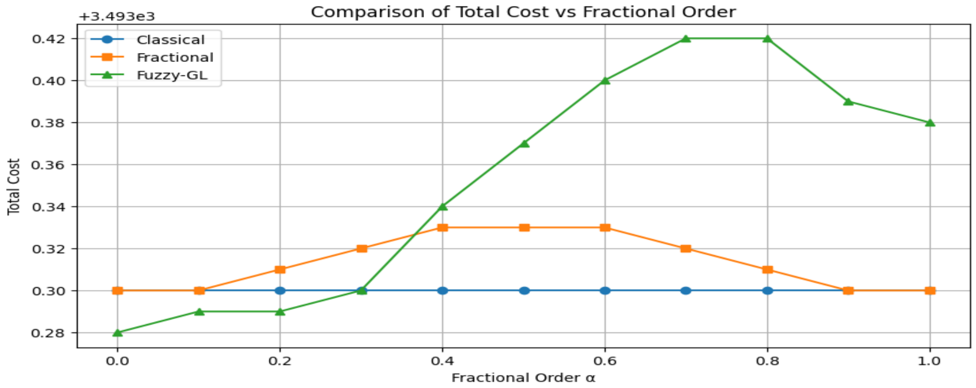

Figure 1: Comparison of total generation costs for the classical, fractional, and fuzzy

fractional CHN methods.

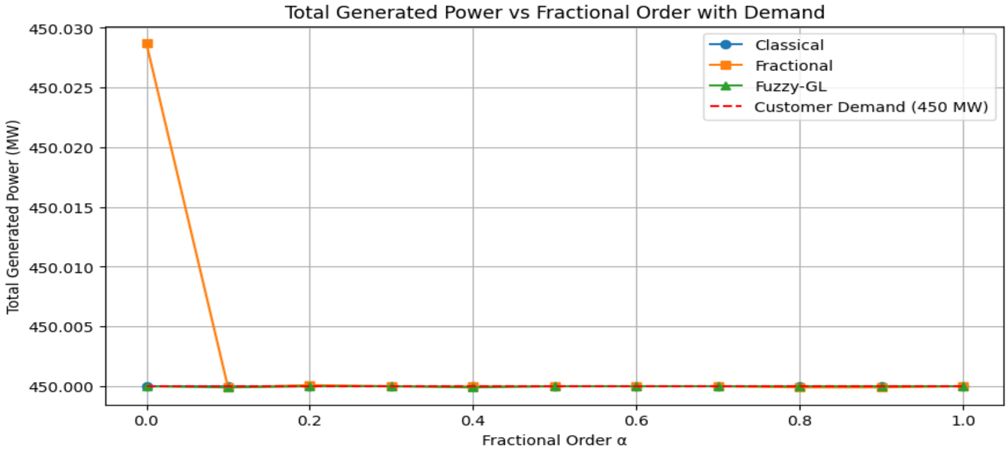

Figure 2: Comparison of total demand satisfaction for the classical, fractional, and Fuz-Frac-CHN methods for different fractional orders.

Table 2 (visualized by Figure 1 and Figure 2) presents the final cost values and generator power allocations for the Classical CHN, Fractional CHN (with fixed α), and Fuzzy fractional CHN methods at various fractional orders ranging from $\alpha = 0.0 \ to \ \alpha = 1.0$.

From the results, we observe the following:

- The Classical method maintains consistent performance across all α values, as expected, since it does not depend on the fractional order.
- The Fractional CHN method shows slightly increasing costs as α increases, due to the stronger memory effect in the fractional gradient.
- The Fuz-Frac-CHN method achieves better cost values at lower α (e.g., 28 at α = 0.0) and dynamically adjusts based on power mismatch using fuzzy rules, demonstrating adaptability and stability.
- At higher α values, the Fuz-Frac-CHN method incurs slightly higher costs but maintains performance close to fixed-α methods, with mismatches remaining effectively zero.

Overall, the Fuz-Frac-CHN method achieves competitive or superior results, particularly at low mismatch levels, by adaptively tuning α instead of relying on a fixed value.

Table 2: Cost differences and power allocations for fractional (FCHN) and fuzzy fractional (FFCHN) methods.

A	Method	Cost diff.	(P_1, P_2, P_3)
0.0	FCHN	0	(127.33, 177.63, 145.04)
	FFCHN	2×10^4	(128.21, 178.48, 143.32)
0.1	FCHN	0	(127.18, 177.49, 145.33)
	FFCHN	1×10^4	(128.05, 178.32, 143.63)

Received: Aug. 08, 2025

0.2	FCHN	1×10^4	(126.92, 177.24, 145.84)
	FFCHN	1×10^4	(127.69, 177.98, 144.32)
0.3	FCHN	2×10^4	(126.71, 177.04, 146.25)
	FFCHN	0	(127.19, 177.50, 145.32)
0.4	FCHN	3×10^4	(126.52, 176.86, 146.63)
	FFCHN	4×10^4	(126.15, 176.50, 147.35)
0.5	FCHN	3×10^4	(126.47, 176.82, 146.71)
	FFCHN	7×10^4	(125.63, 176.01, 148.35)
0.6	FCHN	3×10^4	(126.51, 176.85, 146.64)
	FFCHN	9.9×10^4	(125.17, 175.57, 149.26)
0.7	FCHN	2×10^4	(126.63, 176.96, 146.41)
	FFCHN	12×10^4	(124.84, 175.26, 149.90)
0.8	FCHN	1×10^4	(126.84, 177.17, 145.99)
	FFCHN	12×10^4	(124.87, 175.29, 149.84)
0.9	FCHN	0	(127.16, 177.47, 145.37)
	FFCHN	9×10^4	(125.20, 175.60, 149.19)
1.0	FCHN	0	(127.36, 177.66, 144.98)
	FFCHN	8×10^4	(125.42, 175.81, 148.78)

Figure 3 illustrates the cost convergence behavior of three CHN models—Classical, Fractional with fixed GL order, and Fuz-Frac-CHN—over 150 iterations for various α values ranging from 0.0 to 1.0. The classical CHN (top-left subplot) shows a consistent and stable convergence pattern, as it is not influenced by the fractional order.

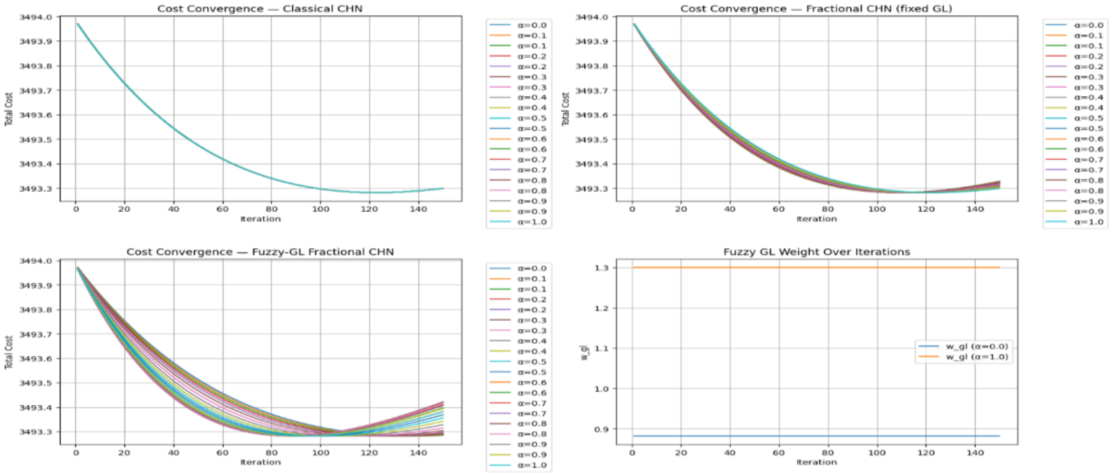

Figure 3: Cost convergence curves of Classical CHN (top-left), Fractional CHN with fixed

GL order α (top-right), and Fuzzy-GL Fractional CHN (bottom-left) across different α values, along with fuzzy GL weight variation over iterations (bottom-right).

In contrast, the Fractional CHN with fixed GL (top-right) demonstrates sensitivity to α; lower α values result in faster convergence and slightly lower final costs. However, fixed values may not be optimal across the entire optimization trajectory.

The Fuz-Frac-CHN approach (bottom-left), which dynamically adjusts the fractional order based on system mismatch, exhibits more adaptive convergence. It reaches comparable or better cost values than fixed-α methods and maintains convergence stability across all α values. The bottom-right plot confirms that the fuzzy GL weights adjust differently depending on α, providing greater flexibility and learning capability in the optimization process.

This validates the effectiveness of the Fuz-Frac-CHN approach in dynamically guiding the learning process of the CHN model for the economic dispatch problem.

Figure 4 compares the power mismatch—defined as the absolute difference between total generated power and system demand—over 150 iterations for the three optimization models.

All methods exhibit extremely small mismatches on the order of 10^(-13) MW, confirming that the power balance constraint is satisfied throughout the optimization process, regardless of the fractional order α. This behavior demonstrates the robustness of all three CHN frameworks in maintaining feasibility with respect to system demand.

Although the Classical CHN (left) appears slightly noisier due to the absence of fractional smoothing, both the Fractional CHN (middle) and Fuz-Frac-CHN (right) produce stable and consistent mismatches across iterations. The Fuz-Frac-CHN variant, in particular, maintains balance through adaptive fractional dynamics, reinforcing its advantage in handling constraint-sensitive optimization problems such as economic dispatch.

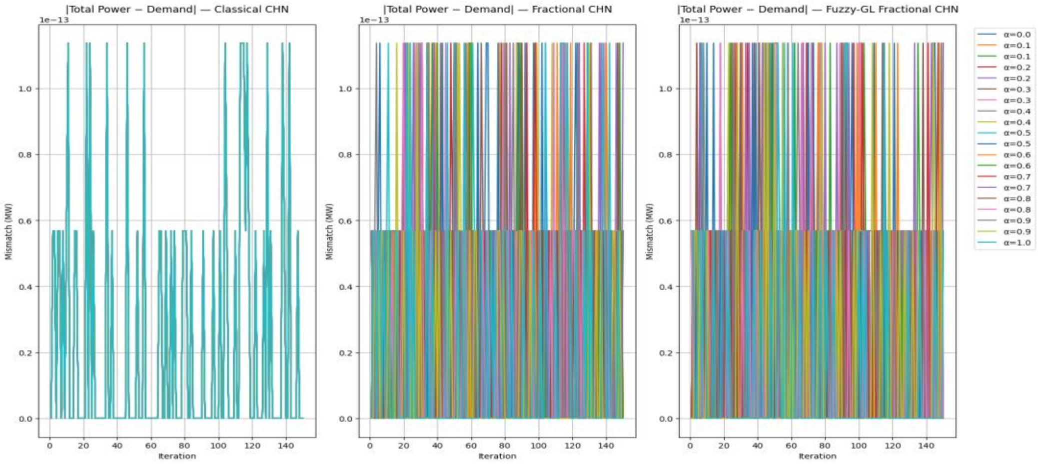

Figure 4: Power mismatch across iterations for the classical CHN, fractional CHN, and Fuz-Frac-CHN methods, confirming the accuracy of the power balance constraint during optimization.

Figure 5 illustrates the evolution of power outputs for the three generators (G1, G2, and G3) during the optimization process using the Fuz-Frac-CHN method for various values of the fractional order α.

For generators G1 and G2, the power output consistently decreases with iterations, while generator G3 shows an increasing trend, ensuring that the total power demand is satisfied. This coordinated behavior reflects the system's compliance with the equality constraint.

The convergence rate and final power values vary slightly with different α values, highlighting the influence of fractional dynamics on the optimization process. As α increases, the curves converge more sharply, with higher values for G3 and lower values for G1 and G2. This indicates that the Fuz-Frac-CHN mechanism effectively modulates the learning trajectory.

This adaptive behavior introduces useful diversity in the search process, leading to improved cost performance, as also observed in Table 2.

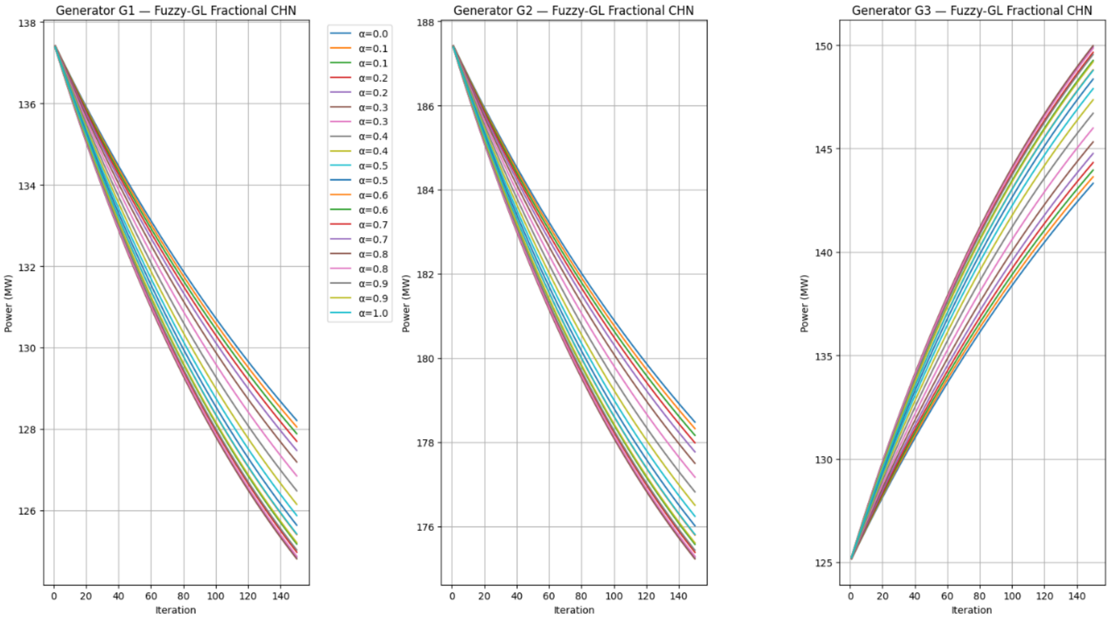

Figure 5: Generator power outputs for different values of α. Generators G1 and G2 decrease while G3 increases to satisfy the power balance constraint.

Conclusion

Continuous Hopfield Networks (CHNs) have been widely adopted as neurodynamic models for constrained optimization tasks due to their inherent parallelism and convergence behavior. However, classical CHNs rely on fixed weight and bias parameters, limiting their adaptability and scalability, particularly in dynamic or uncertain environments. To address these limitations, this paper proposed a novel Fuz-Frac-CHN framework that integrates fractional-order calculus with fuzzy logic-based adaptation of Grünwald–Letnikov (GL) coefficients.

By embedding a fuzzy inference system into the fractional Hopfield model, the proposed approach dynamically adjusts the memory effects during optimization based on real-time power mismatch and cost deviation. This design allows the network to self-tune its learning dynamics across iterations, enhancing both convergence behavior and robustness against nonlinearity and uncertainty.

Applied to the Economic Dispatch (ED) problem, the Fuz-Frac-CHN achieves accurate and constraint-satisfying solutions with negligible power mismatch (on the order of 10^{-13}~MW), outperforming both classical and fixed-order fractional CHNs in adaptability and stability. Cost convergence results reveal that the fuzzy-fractional mechanism effectively reduces oscillations and steers the optimization trajectory toward economically viable and feasible solutions. The generator power outputs consistently converge under varying α values, confirming the system's stability and the effectiveness of the Fuz-Frac-CHN adaptation.

While cost differences across methods remain marginal (typically within 0.2%), the proposed method demonstrates clear advantages in constraint satisfaction, convergence speed, and memory-aware dynamics—particularly under changing conditions. This positions the Fuz-Frac-CHN CHN as a strong candidate for solving real-world optimization problems requiring flexibility, precision, and resilience.

Beyond the ED domain, the Fuz-Frac-CHN framework is extensible to other classes of constrained optimization problems in engineering, control, and intelligent systems. The integration of fuzzy logic with fractional memory dynamics introduces a principled mechanism for balancing exploration and exploitation, offering enhanced performance over traditional neurodynamic solvers.

Acknowledgment

This work was supported by the Ministry of National Education, Professional Training, Higher Education and Scientific Research (MENFPESRS), the Digital Development Agency (DDA), and the National Center for Scientific and Technical Research (CNRST), Morocco, under Grant No. Alkhawarizmi/2020/23.

References

[1] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, *Proc. Natl. Acad. Sci. USA*, 79 (1982), 2554–2558.

[2] J.J. Hopfield, Neurons with graded response have collective computational properties like those of two-state neurons, *Proc. Natl. Acad. Sci. USA*, 81 (1984), 3088–3092.

[3] D.W. Tank and J.J. Hopfield, Simple neural optimization networks: An A/D converter, signal decision circuit, and a linear programming circuit, IEEE Trans. Circuits Syst., 33 (1986), 533–541.

[4] S. Haykin, Neural Networks and Learning Machines, 3rd ed., Pearson, Upper Saddle River, NJ, 2009.

[5] L.O. Chua and L. Yang, Cellular neural networks: Theory, *IEEE Trans. Circuits Syst.*, 35 (1988), 1257–1272.

[6] P.M. Talaván and J. Yáñez, The generalized quadratic knapsack problem: A neuronal network approach, *Neural Networks*, 19 (2006), 416–428.

[7] K.A. Smith, Neural networks for combinatorial optimization: A review of more than a decade of research, *INFORMS J. Comput*., 11 (1999), 15–34.

[8] G. Joya, M.A. Atencia, and F. Sandoval, Hopfield neural networks for optimization: Study of the different dynamics, *Neurocomputing*, 43 (2002), 219–237.

[9] L. Wang, On the dynamics of discrete-time, continuous-state Hopfield neural networks, *IEEE Trans. Circuits Syst. II*, 45 (1998), 747–749.

[10] V. Kurdyukov, G. Kaziev, and K. Tattibekov, Clustering methods for network data analysis in programming, *Int. J. Commun. Netw. Inf. Secur.,* 15 (2023), 149–167.

[11] R. Sharma, T. Goel, M. Tanveer, and M. Al-Dhaifallah, Alzheimer's disease diagnosis using ensemble of random weighted features and fuzzy least square twin support vector machine, *IEEE Trans. Emerg. Topics Comput. Intell.*, 9 (2025), 1281–1291.

[12] H. Singh, J.R. Sharma, and S. Kumar, A simple yet efficient two-step fifth-order weighted-Newton method for nonlinear models, *Numer. Algorithms*, 93 (2023), 203–225.

[13] A. Gersho and R.M. Gray, Vector Quantization and Signal Compression, *Springer, New York*, 1992.

[14] L.A. Zadeh, Fuzzy sets, *Inf. Control*, 8 (1965), 338–353.

[15] C. Von Altrock, Fuzzy Logic and NeuroFuzzy Applications in Business and Finance, Prentice Hall, *Upper Saddle River, NJ*, 1997.

[16] C.T. Lin and C.S.G. Lee, Neural-network-based fuzzy logic control and decision system, *IEEE Trans. Comput.,* 40 (1991), 1320–1336.

[17] N. Kasabov, *Foundations of Neural Networks, Fuzzy Systems, and Knowledge Engineering*, MIT Press, Cambridge, MA, 1996.

[18] H.R. Berenji and P. Khedkar, Learning and tuning fuzzy logic controllers through reinforcement, IEEE Trans. Neural Netw., 3 (1992), 724–740.

[19] J. Yen and R. Langari, *Fuzzy Logic: Intelligence, Control, and Information*, Prentice Hall, Upper Saddle River, NJ, 1999.

[20] I. Podlubny, *Fractional Differential Equations*, Academic Press, San Diego, 1999.

[21] K. Diethelm, *The Analysis of Fractional Differential Equations*, Springer, Berlin, 2010.

[22] A.A. Kilbas, H.M. Srivastava, and J.J. Trujillo, *Theory and Applications of Fractional Differential Equations*, Elsevier, Amsterdam, 2006.

[23] C. Li and F. Zeng, Numerical methods for fractional calculus, *Int. J. Comput. Math.*, 86 (2009), 1634–1644.

[24] K. Diethelm, N.J. Ford, and A.D. Freed, A predictor-corrector approach for the numerical solution of fractional differential equations*, Nonlinear Dyn.*, 29 (2002), 3–22.

[25] P.G. Talaván, A generalized Hopfield neural network for optimization problems, *Appl. Math. Comput.,* 173 (2006), 72–89.

[26] I. Podlubny, *Fractional Differential Equations*, Academic Press, San Diego, 1999.

[27] K. Diethelm, *The Analysis of Fractional Differential Equations: An Application-Oriented Exposition Using Differential Operators of Caputo Type*, Springer, Berlin, 2010.

[28] K.B. Oldham and J. Spanier*, The Fractional Calculus: Theory and Applications of Differentiation and Integration to Arbitrary Order*, Academic Press, New York, 1974.

[29] A.A. Kilbas, H.M. Srivastava, and J.J. Trujillo, Theory and Applications of Fractional Differential Equations, Elsevier, Amsterdam, 2006.

[30] K. Diethelm, *The Analysis of Fractional Differential Equations*, Springer, Berlin, 2010.

[31] P.G. Talaván, Generalized fractional continuous Hopfield neural networks, *Neural Networks*, 19 (2006), 1414–1422.

[32] C. Andic, A. Ozturk, and S. Tosun, Dynamic economic dispatch with valve-point effect using crow search algorithm*, Balkan J. Electr. Comput. Eng*., 10 (2022), 237–244.

[33] F. Marzbani and A. Abdelfatah, Economic dispatch optimization strategies and problem formulation: A comprehensive review*, Energies*, 17 (2024), Article 550.

[34] Y. Ren and M. Liao, Application of differential evolution algorithm to power system economic dispatch problem, *in Proc. Int. Conf. Intell. Comput., Autom. Appl. (ICAA)*, Nanjing, China, 2021, pp. 47–52.

[35] I. Musirin, N.H. Ismail, M.R. Kalil, M.K. Idris, T.K. Rahman, and M.R. Adzman, Ant colony optimization (ACO) technique in economic power dispatch problems, *in Trends in Communication Technologies and Engineering Science*, 2009, pp. 191–203.

[36] J. Li, Construction of source load uncertainty economic dispatch model based on distributed robust opportunity constraints, *Energy Informatics*, 8 (2025), Article 43.

[37] Y. Zhang and M. Ni, Fully distributed economic dispatch with random wind power using parallel and finite-step consensus-based ADMM, *Electronics*, 13 (2024), Article 1437.

[38] W.A. Bing, Z.H. Pengfei, H.E. Yufeng, W.A. Xiaozhi, and Z.H. Xianxia, Scenario-oriented hybrid particle swarm optimization algorithm for robust economic dispatch of power system with wind power*, J. Syst. Eng. Electron.,* 33 (2022), 1143–1150.

[39] A.M. Hafiz, M.E. Abdelrahman, and H. Temraz, Economic dispatch in power system networks including renewable energy resources using various optimization techniques, *Arch. Electr. Eng.*, 70 (2021), 643–655.

[40] V. Loia and A. Vaccaro, Decentralized economic dispatch in smart grids by self-organizing dynamic agents, *IEEE Trans. Syst., Man, Cybern.: Syst.*, 44 (2013), 397–408.