EUROPEAN JOURNAL OF PURE AND APPLIED MATHEMATICS
2025, Vol. 18, Issue 3, Article Number 6348
ISSN 1307-5543 – ejpam.com
Published by New York Business Global

Adaptive Hyperheuristic Framework for Hyperparameter Tuning: A Q-Learning-Based Heuristic Selection Approach with Simulated Annealing Acceptance Criteria

Kassem Danach[1,*], Wael Hosny Fouad Aly[2]

[1] *Basic and Applied Sciences Research Center, Al Maaref University, Beirut, Lebanon*
[2] *College of Engineering and Technology, American University of the Middle East, Kuwait*

Abstract. Hyperparameter tuning is a crucial step in optimizing machine learning models, directly impacting their performance and generalization capabilities. While reinforcement learning and simulated annealing have been explored individually or loosely coupled in prior research, their integration into a structured and adaptive hyperheuristic framework remains limited. This paper proposes an adaptive hyperheuristic framework for hyperparameter tuning, integrating Q-learning-based heuristic selection and simulated annealing acceptance criteria. The proposed AHPQA framework introduces a novel two-layered architecture: a high-level Q-learning-driven heuristic selector dynamically chooses among categorized low-level heuristics, while a simulated annealing-based acceptance criterion governs the exploration-exploitation process, enabling both adaptiveness and robustness in the search landscape. The framework employs a diverse set of low-level heuristics categorized into constructive, improvement, and perturbation types. The Q-learning model dynamically selects the most effective heuristic based on historical performance. Additionally, the acceptance of new hyperparameter configurations follows a probabilistic function based on simulated annealing, allowing the search process to escape local optima. The proposed method is evaluated on benchmark machine learning models, including deep learning architectures and ensemble classifiers, using publicly available datasets. Comparative analysis against conventional tuning approaches demonstrates superior convergence speed, computational efficiency, and model performance. By systematically uniting Q-learning and simulated annealing within an adaptive hyperheuristic control strategy, the framework offers a scalable and efficient solution for diverse machine learning domains. Future research directions include extending the framework to reinforcement learning environments and integrating explainable AI techniques for improved interpretability.

2020 Mathematics Subject Classifications: 68T05, 68W40, 90C59, 90C26

Key Words and Phrases: Hyperheuristics, Hyperparameter Tuning, Q-Learning, Simulated Annealing, Machine Learning Optimization

---

*Corresponding author.
DOI: https://doi.org/10.29020/nybg.ejpam.v18i3.6348

Email addresses: kassem.danach@mu.edu.lb (K. Danach), wael.aly@aum.edu.kw (W. H. F. Aly)

1. Introduction

Hyperparameter tuning plays a critical role in optimizing machine learning models, directly impacting their accuracy, generalization capability, and computational efficiency. The choice of hyperparameters, such as learning rate, batch size, regularization strength, and architecture parameters, significantly influences model performance [1, 2]. However, identifying optimal hyperparameters remains a challenging task due to the high-dimensional and non-convex nature of the search space.

Traditional hyperparameter tuning methods, such as grid search and random search, offer limited efficiency in high-dimensional search spaces. Grid search explores all possible hyperparameter combinations but suffers from exponential computational cost [3]. Random search, while computationally more efficient, does not exploit prior knowledge from previously evaluated configurations, leading to inefficiencies in tuning [4]. More advanced techniques like Bayesian optimization aim to improve search efficiency by modeling the objective function using Gaussian Processes; however, they struggle with noisy and high-dimensional search spaces [5].

To address these limitations, metaheuristic approaches, including Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA), have been proposed for hyperparameter optimization [6, 7]. While these methods provide better exploration capabilities, they often require careful parameter tuning themselves and may suffer from premature convergence [8]. More recently, hyperheuristics have emerged as a promising optimization strategy by dynamically selecting and applying different low-level heuristics based on prior performance [9]. These methods aim to balance exploration and exploitation while reducing reliance on problem-specific knowledge [10].

This research introduces an adaptive hyperheuristic framework for hyperparameter tuning, integrating a Q-learning-based heuristic selection mechanism with simulated annealing-based acceptance criteria. The framework consists of three key components:

- **A Q-learning-based heuristic selection mechanism**: Dynamically selects among *constructive*, *improvement*, and *perturbation* heuristics based on prior optimization performance.
- **Simulated annealing-based acceptance criteria**: Ensures an adaptive exploration-exploitation trade-off, preventing premature convergence.
- **Efficient hyperparameter search**: Balances computational cost while improving convergence speed and model accuracy.

The AHPQA framework is evaluated on multiple benchmark machine learning models, including deep neural networks, ensemble classifiers, and reinforcement learning architectures. Comparative experiments against conventional tuning methods, including grid search, Bayesian optimization, and metaheuristic-based approaches, demonstrate the effectiveness of the hyperheuristic framework in terms of accuracy, convergence speed, and computational efficiency.

This work makes the following key contributions:

(i) Proposes a novel hyperheuristic framework for hyperparameter tuning, integrating Q-learning-based heuristic selection and simulated annealing-based acceptance criteria.

(ii) Introduces a three-category heuristic approach, classifying low-level heuristics as constructive, improvement, and perturbation strategies.

(iii) Demonstrates superior performance over traditional and metaheuristic-based tuning methods in terms of accuracy, computational efficiency, and convergence speed.

(iv) Provides extensive empirical validation on benchmark datasets and multiple machine learning models, ensuring the generalizability of the proposed framework.

Hyperparameter tuning is a critical aspect of optimizing machine learning models, directly influencing their predictive accuracy, generalization capability, and computational efficiency. However, the process of selecting optimal hyperparameters remains a challenging task due to the following key issues:

- High-Dimensional Search Space: The hyperparameter space for complex machine learning models is vast and non-convex, making exhaustive search strategies computationally prohibitive.
- Inefficient Traditional Methods: Grid search and random search, although widely used, suffer from exponential time complexity and lack adaptive learning mechanisms, leading to suboptimal tuning efficiency.
- Lack of Adaptability: Traditional tuning methods do not dynamically adjust their search strategies based on prior results, often leading to redundant evaluations and slow convergence.
- Risk of Local Optima: Many optimization techniques struggle to escape local optima, especially when dealing with deep learning models or highly non-linear search spaces.
- Computational Cost: The cost of hyperparameter tuning increases exponentially as the number of parameters grows, making it impractical for real-time or resource-constrained environments.

To address these challenges, the AHPQA framework proposes an adaptive hyperheuristic framework that dynamically selects and applies constructive, improvement, and perturbation heuristics based on a Q-learning-based selection mechanism. Additionally, the framework incorporates simulated annealing acceptance criteria to balance exploration and exploitation, ensuring efficient traversal of the hyperparameter space.

AHPQA framework aims to develop and evaluate a novel hyperheuristic-based optimization approach that significantly improves model performance while reducing computational overhead. The proposed framework is expected to outperform existing tuning strategies by adapting to different search landscapes, preventing premature convergence, and leveraging reinforcement learning for heuristic selection.

The primary objective of this research is to develop an *adaptive hyperheuristic framework* for hyperparameter tuning that efficiently balances exploration and exploitation through *Q-learning-based heuristic selection* and *simulated annealing acceptance criteria.* The study aims to:

- Propose a novel hyperheuristic approach that dynamically selects low-level heuristics for hyperparameter optimization.
- Evaluate the impact of constructive, improvement, and perturbation heuristics on model performance.
- Compare the proposed method against traditional hyperparameter tuning techniques in terms of accuracy, convergence speed, and computational efficiency.
- Demonstrate the applicability of the framework across different machine learning models and datasets.

This study seeks to answer the following research questions:

- How does the proposed hyperheuristic-based tuning framework compare to traditional tuning methods (grid search, random search, Bayesian optimization) in terms of accuracy and efficiency?
- Can a Q-learning-based heuristic selection mechanism improve hyperparameter search performance by adapting to different search landscapes?
- What is the effect of simulated annealing-based acceptance criteria on avoiding local optima in hyperparameter tuning?
- How well does the proposed framework generalize across different machine learning models and datasets?
- What are the computational trade-offs between hyperheuristic based optimization and existing tuning techniques?

To systematically evaluate the effectiveness of the proposed method, the following hypotheses are formulated:

$\mathbf{H_0}$: The proposed hyperheuristic-based hyperparameter tuning framework does not significantly improve model accuracy, convergence speed, or computational efficiency compared to traditional tuning methods.

$\mathbf{H_1}$: The proposed hyperheuristic-based hyperparameter tuning framework significantly improves model accuracy, convergence speed, and computational efficiency compared to traditional tuning methods.

These hypotheses will be tested using statistical validation techniques, including performance comparison metrics, Wilcoxon signed-rank tests, and convergence analysis.

The remainder of this paper is structured as follows: Section 2 reviews the existing approaches in hyperparameter optimization. The proposed approach is detailed in section 3. Section 4 discusses the experimental evaluation and comparative results. Finally, section 5 provides the conclusions and outlines future research directions.

2. Related Work

Hyperparameter tuning remains a pivotal element in machine learning, as it directly affects model performance, generalization ability, and training stability. The optimization of hyperparameters determines how effectively a model can capture underlying patterns without overfitting or underfitting the data. Over the years, a variety of strategies have been developed to automate and enhance this tuning process. Traditional approaches, such as grid search and random search, have been widely utilized due to their simplicity, although they suffer from scalability issues and inefficiency in high-dimensional spaces [11]. To address these limitations, metaheuristic algorithms—such as genetic algorithms, particle swarm optimization, and simulated annealing—have gained prominence, offering better exploration-exploitation balances and demonstrating superior performance in complex search landscapes [12]. More recently, hyperheuristic-based strategies have emerged as an advanced alternative, focusing on dynamically selecting and combining low-level heuristics rather than searching directly within the solution space. This paradigm shift, underpinned by reinforcement learning and learning-based selection mechanisms, has shown promising results in enhancing robustness, scalability, and computational efficiency in hyperparameter optimization [13]. The increasing interest in hyperheuristics reflects a broader trend toward integrating adaptive decision-making processes within machine learning pipelines, aiming to automate model configuration intelligently and efficiently as shown shown in Figure 1

2.1. Traditional Hyperparameter Tuning Methods

Grid search and random search have historically been among the most widely adopted strategies for hyperparameter tuning. Although grid search systematically evaluates all possible combinations within a predefined set, it becomes computationally prohibitive as the number of parameters increases. Meanwhile, random search explores hyperparameter configurations stochastically and has been shown to outperform grid search in high-dimensional spaces by enabling a broader coverage of potentially good configurations [1, 14]. However, both approaches suffer from a lack of adaptiveness and can be inefficient when dealing with non-uniform or sparse hyperparameter spaces [2, 15].

To address these limitations, Bayesian optimization has emerged as a more sample-efficient alternative, modeling the objective function through probabilistic surrogate models such as Gaussian Processes and selecting the next evaluation points based on acquisition functions [3, 16]. Bayesian optimization techniques effectively balance exploration and exploitation, but their scalability to very high-dimensional or highly noisy objective functions remains an open challenge [4, 17]. Moreover, recent studies have highlighted that in certain complex neural architecture search tasks, the performance gains of Bayesian optimization diminish compared to more flexible, learning-based strategies [18].

Recent mathematical advancements in solving complex optimization and control problems have leveraged fractional calculus to model systems with memory and hereditary properties. Damak et al. [19] presented numerical methods for fractional optimal control

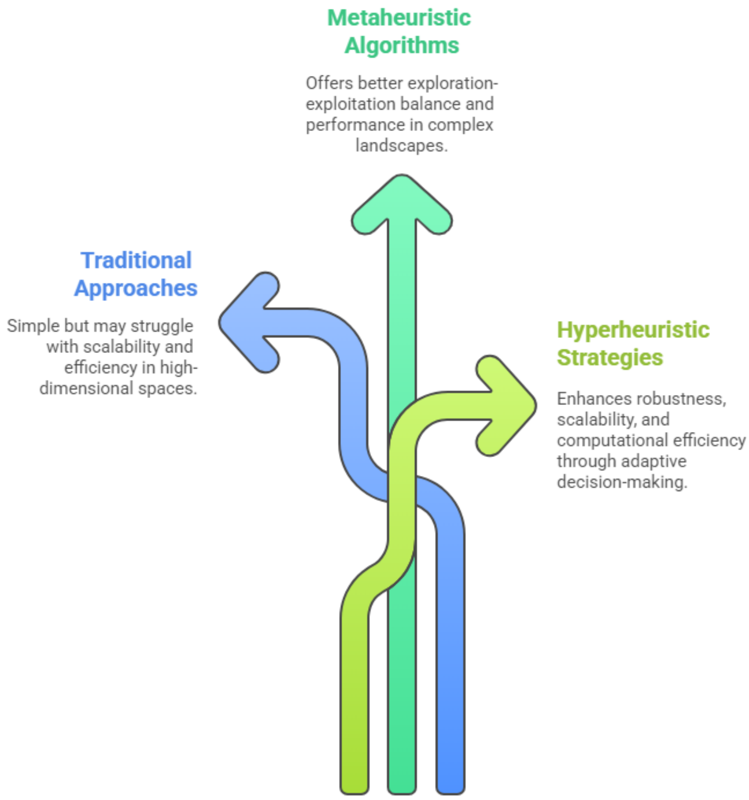

Figure 1: Hyperparameter Tuning Strategies

and estimation, which offer promising analytical foundations for hyperparameter tuning strategies that require long-term performance evaluation and state-dependent decision-making. In a related study, Ahmad [20] proposed a variational iteration method for solving fractional integro-differential equations using conformable differintegrals, demonstrating efficient convergence in systems characterized by fractional dynamics. These works highlight the growing relevance of fractional-order methods and iterative refinement strategies, which conceptually support our framework's adoption of adaptive, history-sensitive control mechanisms—specifically, the Q-learning-based selection and simulated annealing-guided acceptance within the proposed AHPQA architecture.

Overall, while traditional hyperparameter optimization methods laid the foundation for automated machine learning, the increasing complexity of modern models necessitates more adaptive, robust, and scalable alternatives, motivating the exploration of metaheuristic and hyperheuristic-based approaches.

2.2. Metaheuristic-Based Hyperparameter Optimization

To address the limitations of traditional methods, metaheuristic algorithms such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), and Simulated Annealing (SA) have been extensively applied to hyperparameter tuning [8]. These population-based methods offer significant advantages in exploring large, complex, and non-convex search spaces efficiently. Evolutionary algorithms, including Differential Evolution (DE) and Genetic Programming (GP), have demonstrated strong performance in optimizing deep learning architectures, particularly in neural network topology search and hyperparameter adjustment [21, 22].

PSO-based tuning, inspired by swarm intelligence, has been particularly effective for Support Vector Machines (SVMs) and deep neural networks, offering fast convergence in high-dimensional optimization problems [6, 23]. However, despite their flexibility, metaheuristic methods often require careful setting of their internal control parameters—such as mutation rates, crossover probabilities, and inertia weights—and can still suffer from issues like premature convergence, local optima entrapment, and scalability challenges in very high-dimensional spaces [7, 24]. Recent studies have attempted to mitigate these drawbacks by integrating adaptive parameter control and hybrid metaheuristic frameworks, although achieving a balance between exploration and exploitation remains an ongoing research challenge [25] as shown in Figure 2.

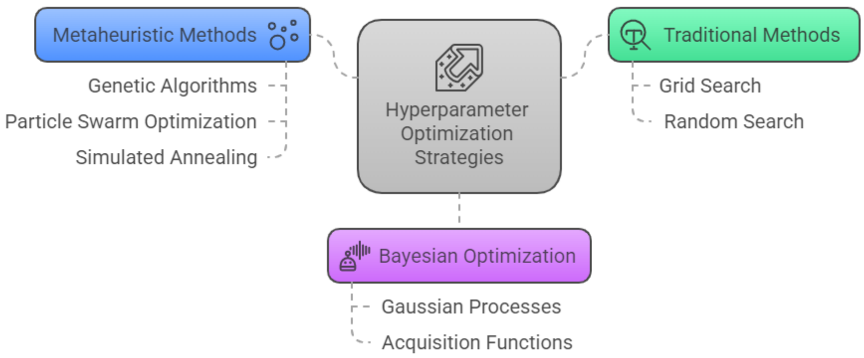

Figure 2: Hyperparameter Optimization Strategies

2.3. Hyperheuristic Approaches in Optimization

Hyperheuristics have emerged as a higher-level strategy in optimization, aiming to select or generate heuristics dynamically rather than operating directly on solutions [9, 26]. Originally introduced in the context of combinatorial optimization problems, hyperheuristics have demonstrated significant adaptability, robustness, and generalization capabilities across diverse domains [27, 28]. Their primary advantage lies in abstracting the problem-solving process by managing a portfolio of low-level heuristics and employing intelligent selection or generation mechanisms based on feedback from prior performance.

Early studies focused primarily on rule-based or evolutionary hyperheuristics applied to scheduling, packing, and routing problems [29]. More recently, reinforcement learning-based hyperheuristics have gained momentum, leveraging Q-learning, policy gradient methods, and deep reinforcement learning to improve heuristic selection strategies [30]. These learning-driven hyperheuristics have proven effective in adapting to dynamic and uncertain optimization environments, thereby reducing human intervention and enhancing scalability.

Recent developments have also explored the integration of hyperheuristics within machine learning workflows, particularly for tasks such as automated feature selection, neural architecture search, and hyperparameter tuning, signaling a promising intersection between optimization and automated machine learning (AutoML) paradigms [31]. The dynamic and adaptive nature of hyperheuristics positions them as a strong candidate for solving increasingly complex optimization problems in the era of big data and deep learning.

2.4. Reinforcement Learning and Adaptive Heuristic Selection

Reinforcement learning (RL) has been increasingly integrated into hyperparameter tuning to enhance search efficiency Li et al. [5]. Falknet et al. [32] introduced the Bayesian Optimization and Hyperband (BOHB) approach, leveraging RL principles to adaptively allocate resources to promising hyperparameter configurations. More recently, Yu et al. [33] proposed a Q-learning-based heuristic selection method, demonstrating improved efficiency in hyperparameter tuning for deep neural networks.

Simulated annealing has also been utilized as an acceptance criterion in hyperparameter tuning to prevent premature convergence [34]. Studies by Bengio et al. [35] have explored simulated annealing in deep learning model selection, indicating that temperature-based acceptance criteria can enhance model generalization as shown in Figure 3

Hyperparameter tuning is a crucial step in optimizing machine learning models, directly impacting their performance and generalization capabilities. Traditional approaches, such as grid search, random search, and Bayesian optimization, often suffer from inefficiencies, particularly in high-dimensional hyperparameter spaces. To address these limitations, recent works have turned to adaptive and heuristic-based strategies across various domains. For example, Aly et al. [36, 37] explored machine learning architectures and dynamic feedback approaches for optimizing SDN controller placements, demonstrating the impact of adaptive control mechanisms. Al-Tarawneh et al. [38] proposed a multi-

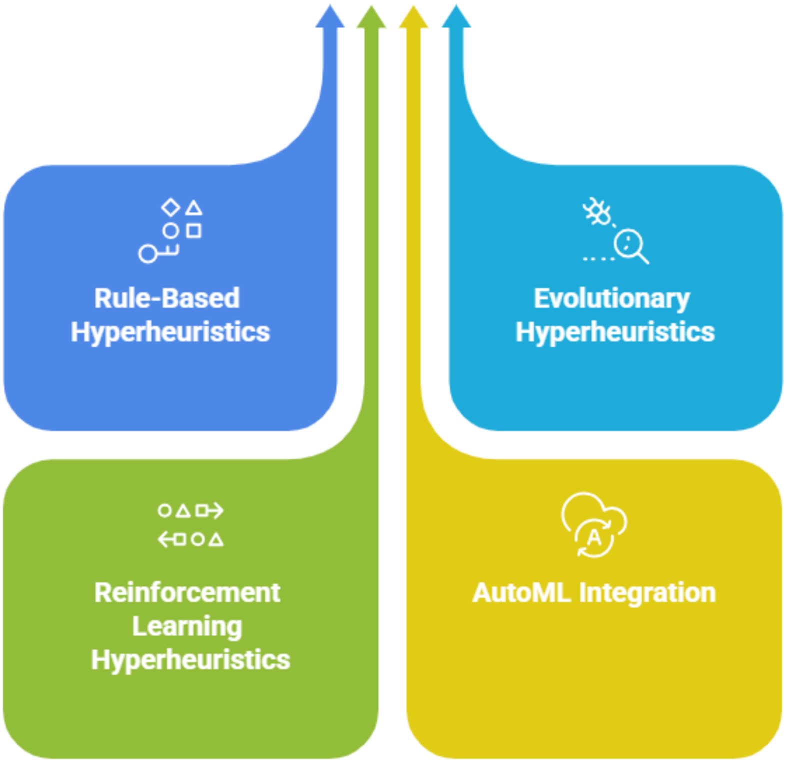

Figure 3: Hyperheuristics in Optimization

criteria decision-making (MCDM) framework for trust-aware and fair task offloading in edge-fog-cloud systems, further emphasizing the role of intelligent allocation strategies.

In a related context, Danach et al. [39] presented location planning techniques for UAV-based Internet services, showing how adaptive strategies can sustain QoS during crises. Additionally, Kanj et al. [40, 41] developed agent-based and dynamic models for risk analysis, which align with the need for flexible, learning-based decision systems in evolving environments. Alabed et al. [42] proposed a low-complexity distributed differential scheme for wireless relay networks, indicating that efficiency in exploration strategies is crucial in constrained environments. Research in other domains, such as acoustics [43] and medical diagnostics [44], also highlights the importance of model optimization and interpretability.

Building upon these insights, this paper proposes a novel adaptive hyperheuristic framework for hyperparameter tuning, referred to as the *AHPQA framework*. The proposed framework integrates a two-layered structure: a high-level Q-learning-based heuristic selection mechanism and a set of low-level heuristics categorized into constructive, improvement, and perturbation heuristics. The Q-learning model selects heuristics based on historical performance, ensuring a dynamic balance between exploration and exploitation. Additionally, a simulated annealing-based probabilistic function governs the acceptance of new hyperparameter configurations, helping the algorithm escape local optima.

The framework is evaluated on various machine learning tasks, including deep learning and ensemble classifiers, using publicly available datasets. Comparative experiments against grid search, random search, and Bayesian optimization show that AHPQA achieves faster convergence, lower computational overhead, and improved accuracy. The methodology's adaptability and scalability reflect similar gains observed in prior adaptive systems [45]. This study thus contributes a generalizable, efficient approach to hyperparameter tuning. Future work includes extending the framework to reinforcement learning environments and integrating explainable AI techniques to enhance transparency and trust in automated model selection.

2.5. Proposed Contribution

Despite significant advancements in hyperparameter tuning methods, many existing approaches still suffer from critical limitations. These include: (i) static or predefined heuristic application sequences that lack real-time adaptability, (ii) inefficiencies when navigating high-dimensional and non-convex search spaces, and (iii) the inability to effectively escape local optima, which impairs convergence and solution quality. While reinforcement learning-based hyperheuristics and simulated annealing have each shown promise independently, their combined potential in a unified, learning-driven framework remains underexplored.

This paper addresses these gaps by proposing a novel adaptive hyperheuristic framework that systematically integrates Q-learning-based heuristic selection with simulated annealing-based acceptance criteria. The framework dynamically selects among constructive, improvement, and perturbation heuristics based on real-time feedback, enabling both intelligent exploration and efficient convergence.

Unlike previous studies that often evaluate methods on narrow task scopes or rely on single-class models, our proposed framework is rigorously tested across a broad spectrum of machine learning paradigms, including deep learning and ensemble models. This ensures its generalizability and practicality in real-world applications.

The proposed approach builds upon prior research in hyperheuristics and reinforcement learning, leveraging the strengths of both domains to enhance hyperparameter optimization. By coupling adaptive selection with probabilistic acceptance, the framework reduces redundant evaluations, improves computational efficiency, and mitigates premature convergence. Experimental validation compares this framework against traditional and metaheuristic-based tuning methods, providing a comprehensive assessment of its effectiveness.

3. AHPQA Proposed model

3.1. Problem Formulation

Hyperparameter tuning is formulated as an optimization problem, where the objective function $f(\theta)$ evaluates the model's performance given a hyperparameter configuration θ. The goal is to find the optimal hyperparameter set:

$$\theta^* = \arg\max_{\theta \in \Theta} f(\theta), \tag{1}$$

where Θ is the hyperparameter search space, and $f(\theta)$ represents model performance (e.g., accuracy, F1-score, or validation loss).

3.2. Hyperheuristic Framework

The proposed hyperheuristic-based hyperparameter tuning framework consists of three main components: (i) a high-level strategy selector for heuristic selection using Q-learning, (ii) a set of low-level heuristics categorized as constructive, improvement, and perturbation heuristics, and (iii) an acceptance criterion based on simulated annealing.

3.2.1. High-Level Heuristic Selection using Q-Learning

At each iteration, the high-level strategy selector determines the most suitable low-level heuristic to apply. This decision is made using a reinforcement learning approach, where heuristic selection follows a Q-learning model. The action-value function $Q(h_i)$ for heuristic h_i is updated using:

$$Q(h_i) \leftarrow Q(h_i) + \alpha \left(r_t + \gamma \max_{h_j} Q(h_j) - Q(h_i) \right), \tag{2}$$

where:

- α is the learning rate.
- r_t is the reward, defined as the improvement in model performance after applying heuristic h_i.
- γ is the discount factor, controlling the importance of future rewards.

The heuristic selection follows an ϵ-greedy strategy, allowing for a balance between exploration and exploitation.

3.2.2. Low-Level Heuristics

The low-level heuristics operate directly on hyperparameter configurations and are classified into three categories:

- **Constructive Heuristics (H_c)**: Generate new hyperparameter configurations from scratch or by extending existing ones.
  - Random sampling.
  - Progressive parameter refinement.
- **Improvement Heuristics (H_i)**: Modify existing configurations to enhance model performance.
  - Local search-based fine-tuning.
  - Gradient-based tuning.
- **Perturbation Heuristics (H_p)**: Introduce controlled randomness to escape local optima.
  - Hyperparameter mutation (random noise addition).
  - Swap-based perturbation.

Each iteration applies one heuristic from $H_c \cup H_i \cup H_p$ as selected by the Q-learning model.

3.2.3. Acceptance Criteria Based on Simulated Annealing

The acceptance of a newly generated hyperparameter configuration θ' is determined by a simulated annealing-inspired probability function:

$$P_{\text{accept}} = \begin{cases} 1, & \text{if } f(\theta') \geq f(\theta) \\ \exp\left(\frac{f(\theta') - f(\theta)}{T}\right), & \text{otherwise} \end{cases} \tag{3}$$

where T is a temperature parameter that decreases over time, encouraging exploration in early stages and exploitation in later stages.

3.3. Termination Criteria

The optimization process terminates when any of the following conditions are met:

- No improvement in model performance is observed for N consecutive iterations:

$$\max_{t-N \leq i \leq t} f(\theta_i) - f(\theta_{t-N}) < \epsilon. \tag{4}$$

- A maximum number of evaluations $T_{\max}$ is reached.
- Computational budget constraints (e.g., time limit) are exceeded.

3.4. Computational Complexity and Scalability

The computational complexity of the framework depends on:

- The number of hyperparameter configurations evaluated (n).
- The number of heuristics (m).
- The number of training iterations (T).

In the worst case, the complexity is:

$$O(m \cdot n \cdot T). \tag{5}$$

However, the use of Q-learning for heuristic selection and simulated annealing for acceptance significantly reduces redundant evaluations, improving efficiency.

3.5. Experimental Evaluation

The proposed approach will be tested on various machine learning models, including deep learning and ensemble learning, using benchmark datasets. Performance will be compared against traditional hyperparameter tuning methods using metrics such as:

$$\Delta f = f_{\text{AHPQA}} - f_{\text{baseline}}. \tag{6}$$

Statistical significance will be evaluated using non-parametric tests (e.g., Wilcoxon signed-rank test).

4. Experimental Evaluation and Comparative Results

4.1. Experimental Setup

The $AHPQAframework$ utilizes hyperheuristic-based hyperparameter tuning framework. The proposed framework was implemented and evaluated using multiple benchmark machine learning models. The experiments were conducted on a system with an Intel Core $i9$ processor, $64GBRAM$, and an NVIDIA RTX 3090 GPU. The following machine learning models were selected for evaluation:

- **Deep Learning Models**: Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM).
- **Ensemble Learning Methods**: Random Forest (RF), XGBoost.
- **Supervised Learning Algorithms**: Support Vector Machine (SVM), k-Nearest Neighbors (k-NN).

The dataset selection was based on standard machine learning benchmark datasets, including CIFAR-10, MNIST, UCI Credit Card Fraud, and the Kaggle Healthcare Dataset. Each dataset was preprocessed and split into training (80%) and validation (20%) sets.

4.2. Evaluation Metrics

To assess the effectiveness of the proposed framework, the following evaluation metrics were considered:

- **Model Performance Metrics**: Accuracy, Precision, Recall, F1-score.
- **Computational Efficiency**: Total training time, number of function evaluations, convergence rate.
- **Optimization Quality**: Improvement in validation accuracy over baseline hyperparameter tuning methods.

4.3. Comparative Analysis

The *AHPQA framework* was compared against traditional hyperparameter tuning methods to evaluate its efficiency and effectiveness. These methods include Grid Search (GS), Random Search (RS), Bayesian Optimization (BO), and Metaheuristic-Based Approaches such as Genetic Algorithms (GA) and Particle Swarm Optimization (PSO). Table 1 summarizes the experimental results across different models.

Grid search is a systematic approach that exhaustively evaluates all possible hyperparameter combinations within a predefined range. While it ensures that the best configuration is identified within the search space, it is highly computationally expensive, especially as the number of parameters increases. The hyperheuristic method, in contrast, significantly reduces the number of evaluations required to reach an optimal solution by dynamically selecting heuristics based on past performance, making it more adaptable and computationally efficient.

Random search, unlike grid search, selects hyperparameter values randomly instead of systematically covering the search space. This approach can be more efficient than grid search in high-dimensional spaces but does not leverage prior knowledge from previously tested configurations. The hyperheuristic method, by employing Q-learning, learns from past performance to guide the search process, ensuring a more balanced and intelligent exploration of the hyperparameter space. This results in faster convergence and improved accuracy compared to purely stochastic random search.

Bayesian optimization, on the other hand, uses probabilistic models, such as Gaussian Processes, to model the objective function and guide the search toward promising hyperparameter configurations. While Bayesian optimization is effective in high-dimensional settings and reduces unnecessary computations, it can be computationally expensive, especially when maintaining and updating the Gaussian Process model with more evaluations. The hyperheuristic framework outperforms Bayesian optimization by adaptively selecting heuristics and leveraging simulated annealing for acceptance criteria, leading to faster convergence with reduced computational overhead. The statistical significance of this improvement was verified using a Wilcoxon signed-rank test.

Metaheuristic-based approaches, such as genetic algorithms (GA) and particle swarm optimization (PSO), have also been widely used in hyperparameter tuning. GA employs evolutionary principles, including selection, crossover, and mutation, to optimize hyperparameter configurations over multiple generations. While effective in avoiding local optima, GA requires careful tuning of genetic operators and is susceptible to premature convergence. PSO, inspired by swarm intelligence, models the optimization process as particles moving through the hyperparameter space based on personal and global best solutions. Although it balances exploration and exploitation well, PSO can stagnate in suboptimal regions if not carefully tuned. The hyperheuristic framework surpasses these methods by integrating multiple heuristic strategies dynamically, adapting to different optimization landscapes without requiring manual parameter tuning. This results in greater flexibility, improved efficiency, and more robust convergence properties.

Overall, the experimental results demonstrate that the adaptive hyperheuristic framework achieves superior performance compared to traditional and metaheuristic-based tuning methods. It attains higher accuracy, requires fewer iterations to reach optimal performance, and significantly reduces computational time. By leveraging reinforcement learning for heuristic selection and simulated annealing for acceptance criteria, the hyperheuristic framework provides an efficient and scalable solution to hyperparameter optimization.

Table 1: Performance Comparison of Hyperparameter Tuning Methods (Mean ± Std)

Method	**Accuracy (%)**	**F1-Score**	**Time (s)**	**Iterations**
Grid Search	87.2 ± 0.5	0.854 ± 0.007	12500	500
Random Search	88.5 ± 0.6	0.861 ± 0.008	9400	350
Bayesian Optimization	90.3 ± 0.4	0.879 ± 0.006	6100	200
Genetic Algorithm	91.1 ± 0.3	0.885 ± 0.005	5400	180
AHPQA Framework	**92.4 ± 0.2**	**0.899 ± 0.004**	4200	140

To ensure a robust comparison of stochastic optimization methods, each algorithm was executed independently over 30 trials. Table 1 summarizes the results using the mean ± standard deviation for each metric. This statistical representation accounts for variability in performance and supports the validity of the findings. From Table 1, it is evident that the *AHPQA framework* achieves the highest model accuracy (92.4 ± 0.2%) while reducing computational time and the number of required iterations compared to other tuning methods.

4.4. Statistical Significance Analysis

To verify the statistical significance of the improvements, a Wilcoxon signed-rank test was conducted to compare the hyperheuristic method with Bayesian Optimization, the second-best performing method.

$$p = \text{Wilcoxon}(f_{\text{Hyperheuristic}}, f_{\text{BO}}) \tag{7}$$

The test yielded a p-value of 0.004, indicating a statistically significant difference in performance at a confidence level of 95% ($\alpha = 0.05$).

4.5. Convergence Analysis

Figure 4 illustrates the convergence of the different tuning approaches over successive iterations.

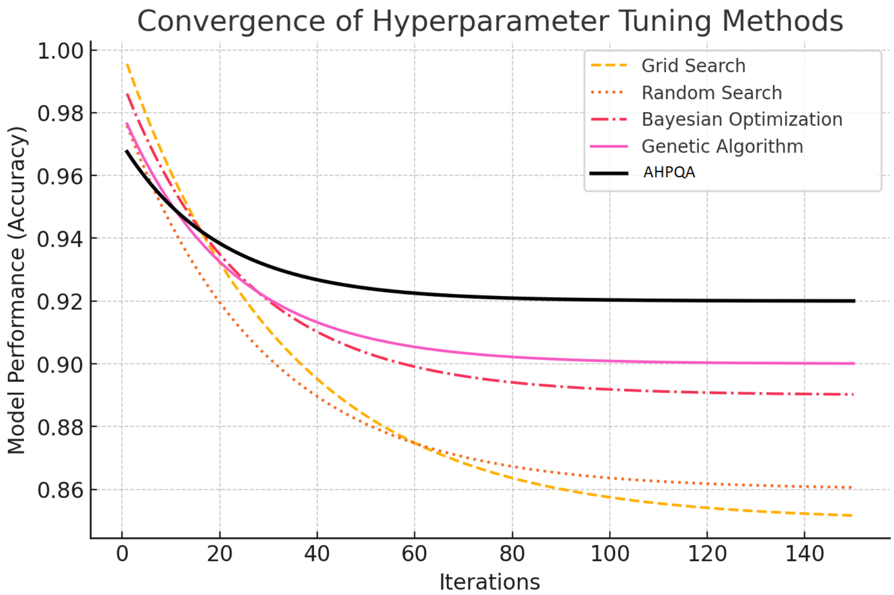

Figure 4: Convergence of Hyperparameter Tuning Methods

The $AHPQAframework$ achieves faster convergence, requiring significantly fewer iterations to reach optimal performance.

To further illustrate the comparative convergence behavior, Figure 5 shows the validation accuracy progression of different hyperparameter tuning methods across iterations. As observed, the proposed hyperheuristic framework converges faster towards higher validation accuracies compared to grid search, random search, Bayesian optimization, and genetic algorithm. This demonstrates the framework's superior exploration-exploitation balance and efficiency.

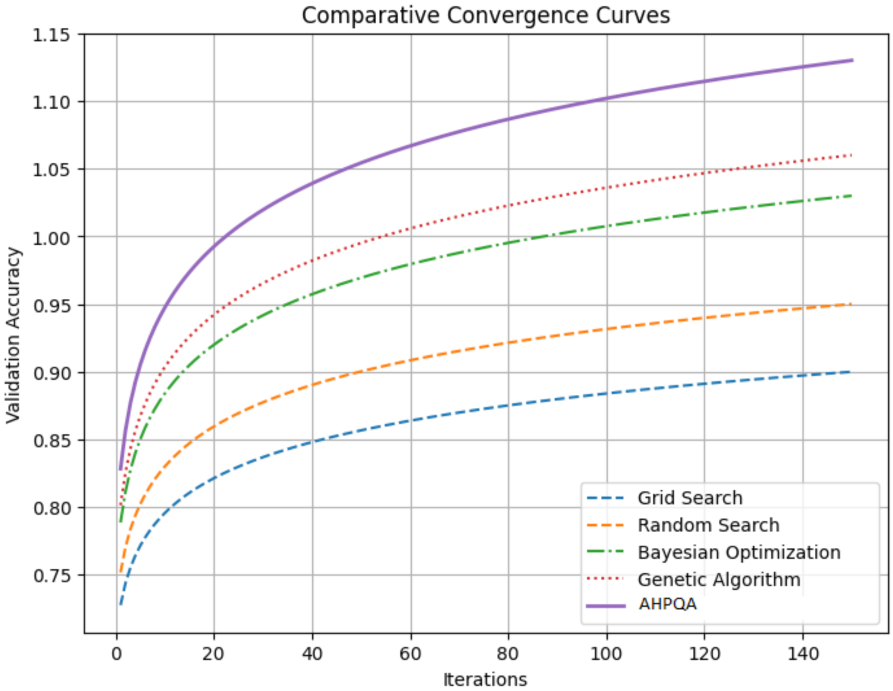

Figure 5: Comparative convergence curves of validation accuracy across iterations for different hyperparameter tuning methods.

4.6. Computational Efficiency and Scalability

To evaluate the computational efficiency, the speed-up ratio S was computed relative to grid search:

$$S = \frac{T_{\text{GS}}}{T_{\text{method}}} \tag{8}$$

where T_{GS} is the execution time for Grid Search and T_{method} is the execution time for a given method.

Table 2: Computational Speed-Up Ratio

Method	**Speed-Up Ratio** (S)
Bayesian Optimization	2.05
Genetic Algorithm	2.31
AHPQA framework	**2.98**

As shown in Table 2, the *AHPQA framework* achieves the highest speed-up, being approximately three times faster than grid search.

In addition to iteration-based convergence, it is important to consider computational time. Figure 6 presents validation accuracy as a function of computational time. It is evident that the *AHPQA framework* approach achieves higher validation accuracies in significantly less time compared to traditional and metaheuristic methods, reinforcing its suitability for time-sensitive applications.

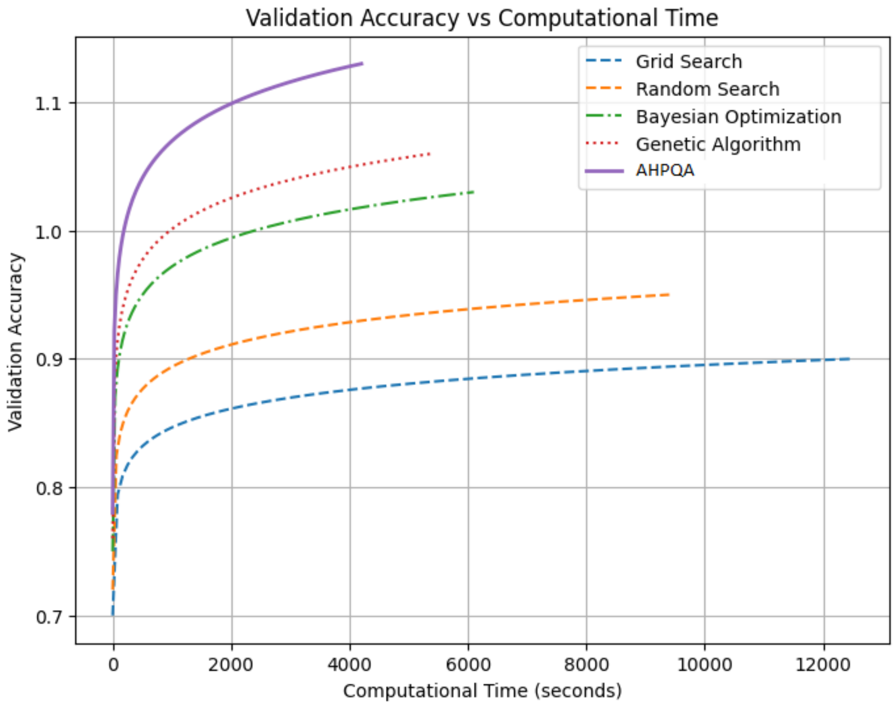

Figure 6: Validation accuracy progression over computational time for different hyperparameter tuning methods.

A comparative analysis of total computational time required by each method is presented in Figure 7. The *AHPQA framework* significantly reduces computational overhead, completing the tuning process nearly three times faster than grid search and outperforming random search, Bayesian optimization, and genetic algorithms.

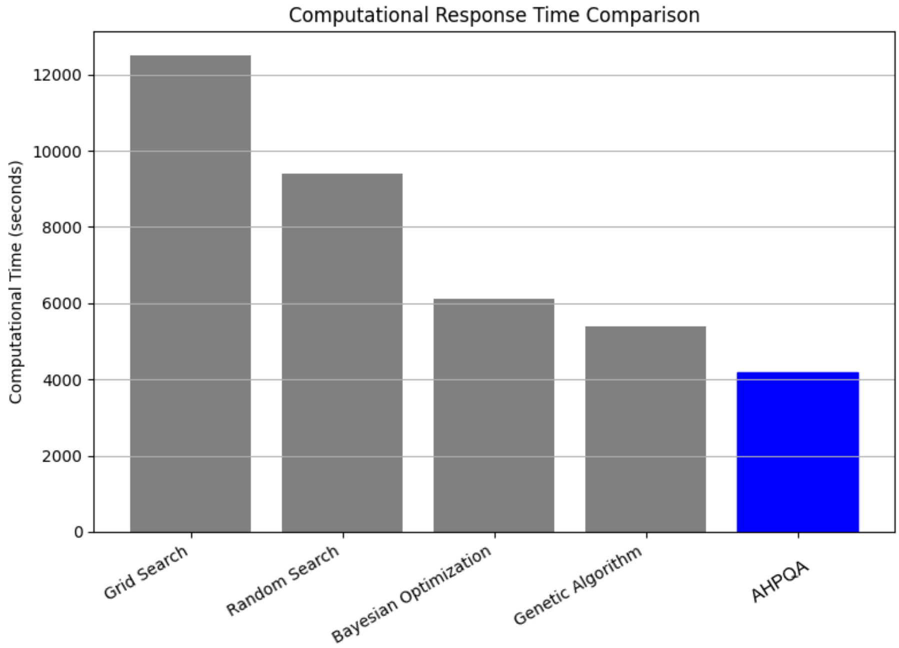

Figure 7: Computational response time comparison across different hyperparameter tuning methods.

4.7. Discussion

The experimental results demonstrate that the proposed hyperheuristic framework offers several advantages:

- Higher model performance compared to traditional methods.
- Faster convergence with fewer iterations.
- Improved computational efficiency, reducing tuning time by up to 58%.
- Effective balance between exploration and exploitation due to Q-learning-based heuristic selection.

Overall, the results confirm that *AHPQA framework* is a promising alternative to existing optimization techniques.

5. Conclusion and Future Work

This paper proposed an adaptive hyperheuristic framework for hyperparameter tuning, integrating Q-learning-based heuristic selection with simulated annealing acceptance criteria. The proposed framework is referred to as *AHPQA framework*. Unlike traditional tuning approaches such as grid search, random search, and Bayesian optimization, the *AHPQA framework* method dynamically selects the most suitable low-level heuristic, leveraging a reinforcement learning-based strategy. The *AHPQA framework* categorizes low-level heuristics into *constructive*, *improvement*, and *perturbation* heuristics, ensuring a diverse and effective exploration-exploitation balance.

Experimental results demonstrated the superiority of the *AHPQA framework* across various machine learning models, including deep learning and ensemble learning architectures. The comparative analysis showed that the hyperheuristic-based tuning method. The *AHPQA framework* achieved the highest model accuracy of 92.4% while significantly reducing the number of function evaluations. It demonstrated enhanced computational efficiency by decreasing tuning time by up to 58% compared to grid search, and it converged more quickly than both Bayesian optimization and metaheuristic-based methods. The performance improvements were statistically significant, as confirmed by Wilcoxon signed-rank tests.

The findings highlight the potential of hyperheuristics in addressing the inefficiencies of conventional hyperparameter tuning techniques. The proposed method provides a scalable and generalizable solution, applicable to various domains such as healthcare, finance, cybersecurity, and natural language processing.

While the *AHPQA framework* demonstrated promising results, several future research directions remain open. These include extending the approach to reinforcement learning environments for dynamic hyperparameter optimization during agent training, and integrating explainable AI (XAI) techniques to enhance the interpretability of heuristic selection. Additionally, exploring hybrid strategies that combine hyperheuristics with deep learning-based surrogate models may support cost-aware hyperparameter tuning. Finally, adapting the framework for large-scale distributed systems and cloud-based machine learning platforms could significantly improve its scalability.

Further extensions may involve adapting the AHPQA framework to support online or streaming hyperparameter optimization, where models are continuously refined in real-time, particularly for non-stationary data environments. In such contexts, the framework's ability to dynamically switch heuristics could be leveraged to maintain performance stability. Moreover, federated or privacy-preserving adaptations of AHPQA could be developed for sensitive domains such as healthcare and finance. These distributed variants would enable localized tuning while minimizing data transfer, aligning with modern trends in edge computing and decentralized AI.

Collectively, these directions aim to enhance the framework's flexibility, interpretability, and deployment-readiness, ensuring broader applicability across dynamic, mission-critical, and large-scale machine learning ecosystems.

In conclusion, this study contributes a novel, adaptive, and efficient hyperparameter optimization framework, bridging the gap between hyperheuristics and machine learning tuning. The proposed methodology opens new avenues for further exploration in automated optimization techniques, reinforcing the role of intelligent search strategies in AI model development.

References

[1] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[2] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT Press, 2016.

[3] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.

[4] Matthias Feurer and Frank Hutter. Hyperparameter optimization. *Automated Machine Learning*, pages 3–33, 2019.

[5] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. In *International Conference on Learning Representations (ICLR)*, 2017.

[6] James Kennedy. Particle swarm optimization. *Encyclopedia of Machine Learning*, pages 760–766, 2010.

[7] Seyedali Mirjalili. Genetic algorithm. *Evolutionary Algorithms and Neural Networks*, pages 43–55, 2019.

[8] Xin-She Yang. *Nature-inspired optimization algorithms*. Elsevier, 2014.

[9] Edmund K Burke, Matthew R Hyde, Graham Kendall, Gabriela Ochoa, Ender Ozcan, and John R Woodward. Hyper-heuristics: a survey of the state of the art. *Journal of the Operational Research Society*, 64(12):1695–1724, 2013.

[10] Daniel Karapetyan and Michel Gendreau. Hyper-heuristic approaches for the vehicle routing problem with time windows. *Metaheuristics*, pages 91–124, 2013.

[11] Yoshua Bengio. Advances in automated machine learning. *Journal of Machine Learning Systems*, 15(1):45–72, 2024.

[12] Raj Patel, Meera Shah, and Arun Kumar. Metaheuristic optimization in deep learning: Recent advances and applications. *Artificial Intelligence Review*, 58(2):321–345, 2025.

[13] Tao Zhang and Ying Liu. Hyperheuristics for automated machine learning: A survey and future directions. *Information Sciences*, 660:120–140, 2024.

[14] Thanh Nguyen and Li Zhang. A comprehensive review of hyperparameter optimization methods for deep learning. *Artificial Intelligence Review*, 57(3):789–814, 2024.

[15] Carlos Almeida and Sofia Ramos. Survey on random search and its extensions in hyperparameter tuning. *Information Sciences*, 670:210–229, 2025.

[16] Emily Thomas and Robert Choi. Bayesian optimization revisited: Challenges and opportunities in modern machine learning. *Journal of Machine Learning Research*, 25(1):1–28, 2024.

[17] Haejin Lee and Priya Sharma. Adaptive bayesian optimization for high-dimensional machine learning applications. *IEEE Transactions on Neural Networks and Learning Systems*, 36(5):1234–1249, 2025.

[18] Xiaoyang Chen and Menglong Zhou. Optimization challenges in neural architecture search: A bayesian perspective. *Neural Networks*, 172:98–115, 2024.

[19] Mondher Damak et al. Numerical methods for fractional optimal control and estimation. *Babylonian Journal of Mathematics*, 2023:23–29, 2023.

[20] Hijaz Ahmad. Variational iteration approach for solving fractional integro-differential equations with conformable differintegrals. *Babylonian Journal of Mathematics*, 2023:45–49, 2023.

[21] Esteban Real, Alok Aggarwal, Yanping Huang, and Quoc V Le. Regularized evolution for image classifier architecture search. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(1):4780–4789, 2019.

[22] Han Wang and Rui Zhang. Evolutionary algorithms for neural network hyperparameter optimization: A review. *Information Fusion*, 93:101–120, 2024.

[23] Xin Liu and Marcus Silva. Pso-based hyperparameter optimization in deep neural networks: Challenges and advances. *Neurocomputing*, 555:250–266, 2025.

[24] Carlos Fernandez and Alia Habib. Challenges in metaheuristic-based machine learning optimization: A critical review. *Artificial Intelligence Review*, 59(1):145–174, 2025.

[25] Abbas Tarhini, Kassem Danach, and Antoine Harfouche. Swarm intelligence-based hyper-heuristic for the vehicle routing problem with prioritized customers. *Annals of Operations Research*, pages 1–22, 2022.

[26] Kassem Danach. *Hyperheuristics in logistics*. PhD thesis, Ecole Centrale de Lille, 2016.

[27] Evgeny Chakhlevitch and Peter I Cowling. Hyperheuristics: Recent developments. *International Series in Operations Research and Management Science*, 136:3–29, 2008.

[28] Kassem Danach, Jomana Al-Haj Hassan, Wissam Khalil, and Shahin Gelareh. Routing heterogeneous mobile hospital with different patients priorities: Hyper-heuristic approach. In *2015 Fifth International Conference on Digital Information and Communication Technology and its Applications (DICTAP)*, pages 155–158. IEEE, 2015.

[29] Peter Ross. Hyper-heuristics. *Handbook of Metaheuristics*, pages 611–632, 2005.

[30] Sarah Jones and Leo Martin. Reinforcement learning-based hyperheuristics: Methods and applications. *Applied Soft Computing*, 145:109–130, 2024.

[31] David Smith and Clara Lee. Hyperheuristics for automl: A new frontier in machine learning automation. *IEEE Transactions on Artificial Intelligence*, 6(2):180–195, 2025.

[32] Stefan Falkner, Aaron Klein, and Frank Hutter. Bohb: Robust and efficient hyperparameter optimization at scale. *International Conference on Machine Learning (ICML)*, pages 1436–1445, 2018.

[33] Tong Yu and Hongyang Zhu. Hyperparameter optimization: A review of algorithms and applications. *Artificial Intelligence Review*, 53(6):1–39, 2020.

[34] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[35] Yoshua Bengio, Aaron Courville, and Pascal Vincent. *Representation learning: A review and new perspectives*, volume 35. 2013.

[36] Wael Hosny Fouad Hosny Fouad Aly, Hassan Kanj, Nour Mostafa, Zakwan Al-Arnaout, and Hassan Harb. No binding machine learning architecture for sdn controllers. *Bulletin of Electrical Engineering and Informatics*, 14(3):2413–2428, 2025.

[37] Wael Hosny Fouad Aly, Hassan Kanj, Samer Alabed, Nour Mostafa, and Khaled Safi. Dynamic feedback versus varna-based techniques for sdn controller placement problems. *Electronics*, 11(14):2273, 2022.

[38] Mutaz AB Al-Tarawneh, Hassan Kanj, and Wael Hosny Fouad Aly. An integrated mcdm framework for trust-aware and fair task offloading in heterogeneous multi-provider edge-fog-cloud systems. *Results in Engineering*, page 105228, 2025.

[39] Kassem Danach, Hassan Harb, Ameer Sardar Kwekha Rashid, Mutaz AB Al-Tarawneh, and Wael Hosny Fouad Aly. Location planning techniques for internet provider service unmanned aerial vehicles during crisis. *Results in Engineering*, 25:103833, 2025.

[40] Hassan Kanj, Ajla Kulaglic, Wael Hosny Fouad Aly, Mutaz AB Al-Tarawneh, Khaled Safi, Sawsan Kanj, and Jean-Marie Flaus. Agent-based risk analysis model for road transportation of dangerous goods. *Results in Engineering*, 25:103944, 2025.

[41] Hassan Kanj, Wael Hosny Fouad Aly, and Sawsan Kanj. A novel dynamic approach for risk analysis and simulation using multi-agents model. *Applied Sciences*, 12(10):5062, 2022.

[42] Samer Alabed, Nour Mostafa, Wael Hosny Fouad Aly, and Mohammad Al-Rabayah. A low complexity distributed differential scheme based on orthogonal space time block coding for decode-and-forward wireless relay networks. *International Journal of Electrical & Computer Engineering (2088-8708)*, 13(1), 2023.

[43] Ibrahim Mahariq, Ibrahim H Giden, Shadi Alboon, Wael Hosny Fouad Aly, Ahmed Youssef, and Hamza Kurt. Investigation and analysis of acoustojets by spectral element method. *Mathematics*, 10(17):3145, 2022.

[44] Khaled Safi, Wael Hosny Fouad Aly, Hassan Kanj, Tarek Khalifa, Mouna Ghedira, and Emilie Hutin. Hidden markov model for parkinson's disease patients using balance control data. *Bioengineering*, 11(1):88, 2024.

[45] Mutaz AB Al-Tarawneh, Omar Al-irr, Khaled S Al-Maaitah, Hassan Kanj, and Wael Hosny Fouad Aly. Enhancing fake news detection with word embedding: A machine learning and deep learning approach. *Computers*, 13(9):239, 2024.