International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4205 - 4212
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0545

Multi-Objective Neural Architecture Search (MONAS) Framework using Genetic Algorithms

Mohini Avatade[1], Kasturi Apte[2], Sharvari Dhote[3], Shivali Waghmare[4], Ketki Takalkar[5]
[1]Department of computer engineering Dr. D. Y. Patil Institute of Engineering management and Research and DYPIU,Pune, India
[2,3,4,5]Department of computer engineeringDr. D. Y. Patil Institute of Engineering management and Research Pune, India
Emails: *mohini.avatade@dypiemr.ac.in[1], kasturiapte765@gmail.com[2], dhotesharvari6@gmail.com[3], shivaliwaghmare2004@gmail.com[4], ketkitakalkar2004@gmail.com[5]*

Abstract
The traditional approaches to Neural Architecture Search (NAS) are typically limited to the maximization of accuracy, which leads to large and computationally expensive models that are not always applicable in practice, e.g. real-time or edge deployment scenarios. This brings out the necessity of a framework that can optimize multiple goals including accuracy, inference latency, memory usage and model complexity simultaneously. This paper presents a design-oriented approach for Multi-Objective Neural Architecture Search framework, MONAS, which is a framework of evolutionary optimization. Unlike traditional NAS techniques that optimize a single objective, MONAS formulates architecture search as a multi-objective optimization problem and it employs genetic algorithms to evolve a diverse set of candidate architectures, out of which the most optimal architecture is chosen. Pareto optimality is used to guide the search process so that the architectures offering balanced trade-offs between various performance measures can be identified. The framework uses the NAS-Bench-301 surrogate model to predict the performance of architectures without training them, which minimizes the computational cost. A directed acyclic graph (DAG) is used to represent each candidate neural architecture in the search space. This allows effective execution of genetic operations such as crossover and mutation. The selection process is performed using NSGA-II and Pareto-optimal architectures are passed on to the next generation.

Keywords: *Neural Architecture Search (NAS), Multi-Objective Optimization, Genetic Algorithms, NAS-Bench-301, Surrogate Model, Pareto Optimization.*

1. Introduction

Deep learning has been incredibly successful in a very broad spectrum of fields, largely due to the creation of more and more complex neural network architectures. Nevertheless, manual design of optimal architectures is time consuming and needs extensive knowledge and experience in the domain. In order to overcome this drawback, Neural Architecture Search (NAS) has become an automated method of finding high-performing neural network architectures [1] [7][8]. The goal of NAS methods is to search a predefined search space and find architectures with the highest predictive performance. Although it has been successful, traditional NAS methods are largely aimed at maximizing one objective, which is usually accuracy. This frequently results in the creation of huge and computationally intensive models, which cannot be deployed in environments with limited computing power or with hard latency limits. As noted in the recent research, the absence of attention to various goals including model size, inference latency, and computational cost restricts the practical usefulness of most NAS-generated architectures [1] [6][9]. In practice, neural networks have to meet several, and often conflicting, requirements. As an example, higher accuracy can lead to a more complex model, whereas lowering the cost of computation can lead to worse performance. The resulting trade-off is that NAS is formulated as a multi-objective optimization problem. Rather than finding one optimal solution, the aim is to find a group of solutions that reflect various trade-offs between competing objectives. This notion is mathematically defined by Pareto optimality in which a solution is said to be optimal when no objective can be maximized without minimizing at least one other objective [2]. Evolutionary algorithms have become popular in order to solve multi-objective optimization problems because they are population-based search and can explore a wide range of solutions. Specifically, the NSGA-II has been demonstrated to be very effective in finding Pareto-optimal solutions and at the same time maintaining diversity in the population [2]. The combination of evolutionary strategies and neural architecture search has also been investigated recently, with promising results being obtained [4]. The second significant issue in NAS is the computational cost of the evaluation of candidate architectures. The cost of training every architecture is prohibitively high, particularly when searching large spaces. In order to address this shortcoming, surrogate-based methods have been suggested, in which predictive models are employed to determine the performance of candidate architectures. One such benchmark is the NAS-Bench-301 which offers a surrogate model that can predict the performance of architecture with high accuracy and the computational overhead is minimized [3]. This paper is inspired by these issues and suggests MONAS, a Multi-Objective Neural Architecture Search framework, which combines evolutionary optimization with surrogate-assisted evaluation. The suggested framework presents architecture search as a multi-objective problem and uses genetic algorithms to evolve architectures in the form of Directed Acyclic Graphs in a structured search space. Through the effective use of surrogate models to evaluate efficiently and the use of NSGA-II to select Pareto-optimal architectures, MONAS will produce a wide range of architectures with balanced trade-offs in terms of accuracy and computational efficiency.

The key contributions of this work are as follows:

- A neural architecture search framework that is multi-objective and pareto optimal, i.e. balanced trade-off between multiple objectives like accuracy, latency, computational cost, model complexity, etc.
- Combination of genetic algorithms and Pareto-based optimization with the help of NSGA-II to explore the search space effectively.
- Surrogate-assisted evaluation with NAS-Bench-301 adopted to greatly decrease computational needs.
- An architecture representation in the form of Directed Acyclic Graphs to support the efficient execution of evolutionary operations.

2. Literature Review

Neural Architecture Search (NAS) has become one of the leading research directions that focus on the automation of neural network architecture design. In the last ten years, many different NAS methods have been suggested, such as reinforcement learning-based, gradient-based optimization, and evolutionary algorithms [1][7][8][9]. In a detailed survey, White et al. point out that although NAS methods have demonstrated state-of-the-art performance on a wide range of tasks, they tend to be expensive to compute and have a strong emphasis on single-objective optimization, which is mainly accuracy [1]. Efficient evaluation of candidate architectures is one of the key problems of NAS research. Traditional approaches include full training of all candidate architectures, which is computationally infeasible. In order to address this, surrogate benchmarks have been suggested as an alternative evaluation plan. Zela et al. suggested surrogate-based NAS benchmarks that go beyond small tabular search spaces by training predictive models that can approximate architecture performance [3]. A significant step towards this direction is the NAS-Bench-301 which offers a scalable and efficient performance prediction framework. This greatly minimizes the computational load and allows exploration of the search space to a greater extent. Although the initial NAS techniques were aimed at optimizing a single objective, recent studies have moved towards multi-objective optimization to capture more accurately the deployment needs of the real world. Practically, models must compromise a number of conflicting factors such as accuracy, computational complexity, memory usage and inference latency. This has been popularly used with evolutionary algorithms because they have the inherent capability of dealing with multiple objectives and maintaining a diverse population of solutions. Specifically, the NSGA-II has become a standard method of solving multi-objective optimization problems to find Pareto-optimal solutions [2]. Cai and Luo investigated the combination of multi-task learning and multi-objective evolutionary NAS and showed that evolutionary approaches can be useful in balancing conflicting goals and enhancing generalization performance [4][10]. Their article emphasizes that it is possible to combine evolutionary computation with advanced learning techniques to enhance the efficiency and effectiveness of NAS systems. In spite of these developments, there are a number of drawbacks in current methods. Most NAS approaches continue to be characterized by high computational cost, poor extrapolation of search space, and inflexibility in responding to user-specified constraints. Also, the combination of multi-objective optimization and effective surrogate-based assessment is a field where there is a lot of room to develop. The suggested MONAS framework can solve these issues by integrating multi-objective evolutionary optimization with surrogate-assisted assessment. With the help of genetic algorithms to evolve architecture, NSGA-II to select Pareto-based, and NAS-Bench-301 to predict the performance of the neural architecture efficiently, MONAS is expected to offer a scalable and flexible solution to the problem of neural architecture search with a variety of constraints.

3. Methodology

This section outlines the proposed MONAS (Multi-Objective Neural Architecture Search) framework, which combines evolutionary optimization with surrogate-assisted evaluation to effectively find neural architectures that meet multiple performance goals.

3.1.Problem Formulation

Define The neural architecture search problem is defined as a multi-objective optimization problem in a search space of candidate architectures A. The objective is to find a combination of architectures that maximizes a number of conflicting objectives at once:

$$\max_{A_0 \in A} \{\boldsymbol{f_1(A), f_2(A), \ldots, f_k(A)}\}$$

where:

- f1 (A) is predictive performance (accuracy),
- f2 (A), f3(A), etc. are computational constraints, including FLOPs, number of parameters, and inference latency, etc.

There is no optimal solution because of the conflicting nature of these objectives. Rather, the goal is to achieve a Pareto-optimal collection of architectures, in which no solution can be optimized in a single objective at the expense of another [2].

3.2.Search Space Design

MONAS uses a directed acyclic graph (DAG) representation of the search space, which is a cell-based representation. The architectures consist of reusable computational cells, stacked to create a complete neural network.

Within each cell:

- Intermediate feature maps are denoted by nodes.
- Edges denote operations that are chosen out of a set of pre-defined operations such as convolution, pooling and identity mappings.

This representation is based on differentiable NAS methods and allows manipulating architectures flexibly during evolutionary operations. The encoding of the DAG guarantees that the architectures generated are valid and that crossover and mutation can be effectively applied.

3.3.Surrogate-Assisted Evaluation

The Full training of candidate architectures is computationally expensive and infeasible in many cases with large-scale search. To overcome this, MONAS uses the NAS-Bench-301, a surrogate benchmark that forecasts the performance of architectures without full training [3]. The surrogate model estimates validation accuracy and computational cost metrics. This greatly minimizes the search time and allows a greater number of candidate architectures to be explored, as also noted in earlier research on surrogate NAS benchmarks [3].

3.4.Genetic Algorithm-Based Search

The basic search process of MONAS is a Genetic Algorithm (GA), in which every candidate architecture is a member of a population.

- Initialization: The search space is sampled to produce an initial population P_0 of size N.
- Evolution Process: The steps to be carried out in each generation g = 1,2,….,G are as

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4205 - 4212
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0545

follows:

- Fitness Evaluation: The surrogate predictions of various objectives are used to evaluate each individual.
- Selection: The candidate architectures are chosen according to their Pareto rank and diversity.
- Crossover: The new offspring are created by fusing substructures of parent architectures, allowing new areas of the search space to be explored.
- Mutation: Random mutations are made to ensure diversity, including, replacing operations and changing node relationships.
- Elitism: Architectures that perform well are maintained over generations so as to guarantee convergence.

This evolutionary process enables the framework to refine architectures through an iterative process and to search a wide range of solutions.

3.5.Multi-Objective Optimization using NSGA-II

After MONAS uses the NSGA-II to rank and select to effectively manage multiple objectives [2].

NSGA-II performs:

- Non-dominated sorting to sort solutions into Pareto fronts.
- Computation of crowding distance to preserve diversity.

An architecture A_i is said to dominate another architecture A_j in case:

$$\forall k, f_k(A_i) \geq f_k(A_j) \text{ and } \exists k \text{ such that } f_k(A_i) > f_k(A_j) \quad (1)$$

This guarantees that the algorithm approaches a well-spread Pareto front that depicts optimal trade-offs between objectives.

3.6.Overall Workflow

- The MONAS framework works as follows pipeline:
- Define search space and objective functions.
- Initialize the set of candidate architectures.
- Evaluate architectures with surrogate models.
- Use genetic operations (selection, crossover, mutation)
- Rank solutions with NSGA-II.
- Do it again through several generations.
- Return a set of Pareto-optimal architectures.

This workflow allows to explore the architecture space efficiently and balance between performance and computational efficiency.

4. Proposed Algorithm

In this section, the algorithmic framework of MONAS, a combination of Genetic Algorithms and multi-objective optimization via NSGA-II, is introduced to effectively search the neural architecture space.

4.1. Algorithm Overview

MONAS works with a population of candidate neural architectures, and evolves them through a series of generations. Every member of the population is a neural architecture coded as a Directed Acyclic Graph (DAG). The algorithm will attempt to optimize this population to a series of Pareto-optimal solutions that will trade off various goals, including accuracy and computational efficiency. The general steps include initiation, screening, selection, replication (crossover and mutation), and substitution, and are directed by principles of multi-objective optimization.

4.2.Pseudocode of MONAS

- Input: Search space A, Population size N, Generations G, Mutation rate p_m, Crossover rate p_c
- Output: Pareto-optimal set P*

Pseudocode:

1: Initialize population P_0 with N random architectures
2: Evaluate each individual using surrogate model
3: for g = 1 to G do
4: Perform non-dominated sorting using NSGA-II
5:Compute crowding distance
6: Select parent population P_par
7: Initialize Q_g = Ø
8: while |Q_g| < N do
9: Select A_1, A_2 from P_par
10:Apply crossover (p_c)
11: Apply mutation (p_m)
12: Add offspring to Q_g
13: end while
14: Evaluate Q_g

e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4205 - 4212
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0545

```
R_g = P_g ∪ Q_g
Perform non-dominated sorting
Select best N → P_{g+1}
end for
Return Pareto-optimal set P*
```

4.3.Important Algorithm Components.

- **Population Initialization:** The first population is created by sampling valid architectures in the predefined search space. The architectures are represented as a DAG, which guarantees structural soundness and compatibility with evolutionary operations.
- **Fitness Evaluation:** The evaluation of each individual is based on a surrogate model offered by NAS-Bench-301, predicting performance metrics like accuracy and computational cost [3]. This avoids the need for expensive full training.
- **Selection Strategy:** The selection is done based on Pareto-based ranking and crowding distance as defined in NSGA-II [2]. This ensures that solutions of high quality are prioritized and variety in solutions is preserved.
- **Crossover Operation:** Crossover is a mixture of structural elements of two parent architectures. This entails the exchange of subgraphs or links in the DAG representation, which enables the algorithm to experiment with new architectural structures.
- **Mutation Operation:** Mutation brings about random changes to ensure that there is diversity in the population. Common mutation operations are:
    - Replacing an operation on an edge.
    - Changing node connectivity.
- These stochastic modifications assist in avoiding premature convergence and allow exploration of new areas in the search space.
- Elitism and Replacement
- The algorithm keeps the best individuals between generations to converge to the best solutions. The total population of parents and offspring is ranked with the help of NSGA-II, and the top N individuals are chosen to be included in the next generation.

4.4.Algorithm Characteristics

The algorithm proposed has the following properties:

Multi-objective optimization: Optimizes several conflicting objectives at once with Pareto dominance.

Sample efficiency: Minimizes the cost of computation by use of surrogate evaluation.

Diversity preservation: Maintains a variety of solutions with crowding distance.

Scalability: Can efficiently search large search spaces.

5. Evaluation And Comparative Analysis

Without empirical assessment, the anticipated performance of the suggested MONAS framework is examined in terms of known results in neural architecture search, multi-objective optimization, and evolutionary algorithms.

5.1.Pareto Front Formation

The main consequence of MONAS is expected to be the emergence of a clear Pareto front, which is the optimal trade-offs between conflicting goals like predictive performance and computational efficiency. Multi-objective methods are also developed to produce a collection of non-dominated architectures, in contrast to single-objective NAS methods, which produce a single optimal solution. The population is supposed to approach the Pareto-optimal region as the evolutionary process continues, and a smooth trade-off curve between objectives is formed. This is in line with the nature of NSGA-II, which has been extensively tested to be effective in estimating Pareto fronts in complex optimization problems [2].

5.2.Convergence of Evolutionary Search.

The combination of Genetic Algorithms and Pareto-based selection is likely to exhibit stable intergenerational convergence. By repeated use of selection, crossover and mutation:

- The inferior (dominated) solutions are gradually filtered out.
- High quality solutions are maintained through elitism.
- Crowding distance is used to maintain diversity.

This convergence behavior has been previously seen in previous evolutionary NAS studies, in which population-based search strategies are effective in balancing between exploration and exploitation to find high-performing architectures [4][5][10].

5.3.Efficiency using Surrogate Modelin

It is assumed that the use of the NAS-Bench-301 will make the search process much more efficient. MONAS minimizes the computational cost by substituting the costly training-based evaluation with predictive modeling, likely without compromising the accuracy of performance estimation.

Surrogate-assisted NAS methods have been demonstrated to be highly correlated to actual model performance, allowing them to converge more quickly and explore more of the search space [3]. This enables MONAS to scale well without prohibitive computational costs.

5.4.Trade-off Characteristics

The resulting Pareto front is likely to depict obvious trade-offs between goals:

- More accurate architectures may tend to have greater computational complexity.
- Lightweight architectures are expected to have reduced latency and resource consumption at the expense of slight performance deterioration.

These trade-offs are natural to the design of neural architecture, and have been repeatedly observed in NAS literature, with no single architecture being optimal to achieve all goals at once [1][6].

5.5.Diversity of Solutions

A key anticipated consequence of MONAS is that a wide range of architectures is generated. The evolutionary algorithms are population-based, which, along with NSGA-II, guarantees that solutions are well-spread in the Pareto front. This diversity enables:

- Multiple regions of the search space covered.
- Access to architectures that are appropriate to various deployment constraints.
- Decreased chances of premature convergence.

5.6.Overall Expected Outcome

Altogether, the MONAS framework is expected to:

- Efficiently approximate the Pareto-optimal set of neural architectures
- Maintain balance between predictive performance and computational efficiency
- Reduce search cost using surrogate-assisted evaluation
- Support a wide range of candidate architectures for various application constraints.

These anticipated results are consistent with the previous studies in multi-objective evolutionary NAS and surrogate modeling, which justifies the viability and efficiency of the given method.

5.7.Comparative analysis and discussion

The proposed MONAS framework is analyzed in comparison with existing Neural Architecture Search (NAS) approaches to highlight its design advantages and expected performance characteristics. While full experimental validation is computationally intensive and part of ongoing work, a comparative study based on key design parameters provides insights into the effectiveness of the proposed approach.

Table 1 Comparison of MONAS with Existing NAS Approaches

Parameter	RL based NAS	DARTS	Evolutionary NAS	MONAS (Proposed)
Optimiza-tion Type	Reinforce-ment Learning	Gradient based	Evolutionary	Evolutionary (NSGAII)
Multi objectve Support	Limited	Limited	Partial	Full
Computa-tional Cost	High	Moderate	High	Reduced (Surrogate based)
Search Efficiency	Moderate	High	Moderate	High
Flexibility	Low	Moderate	High	High

As observed in Table I, traditional NAS approaches primarily focus on single-objective optimization, often emphasizing accuracy while overlooking computational constraints. In contrast, the proposed

e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4205 - 4212
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0545

MONAS framework incorporates multi-objective optimization using NSGA-II, enabling simultaneous consideration of accuracy and efficiency metrics Furthermore, the use of surrogate modeling significantly reduces computational overhead compared to conventional NAS methods that require full training of candidate architectures. This allows MONAS to explore a larger search space more efficiently while maintaining solution diversity through Pareto-based selection.

Conclusion

This paper introduced a design-oriented approach towards MONAS, a Multi-Objective Neural Architecture Search methodology which combines evolutionary optimization with surrogate-assisted evaluation to design neural architectures efficiently. In contrast to the traditional NAS approaches that are mostly aimed at a single goal, the proposed one develops architecture search as a multi-objective problem, allowing to optimize predictive performance and computational constraints simultaneously.The framework proposes use of genetic algorithms to develop candidate architectures by evolving in a structured search space, and use of Pareto-based optimization with NSGA-II, to identify a wide range of non-dominated solutions [2]. To overcome the computational difficulties of NAS, MONAS proposes to use the NAS-Bench-301 surrogate model, which is much cheaper to evaluate and still provides reliable performance estimation [3]. The anticipated results suggest that MONAS can produce a well-distributed Pareto front of neural architectures, which trade-offs between accuracy and efficiency. The framework offers a scalable and flexible automated neural architecture design solution by integrating multi-objective optimization with surrogate modeling. On the whole, MONAS may help to improve the development of Neural Architecture Search, as it can overcome the main shortcomings of the current methods, especially in the aspects of computational efficiency and multi-objective optimization. The presented work adopts a design-oriented approach, focusing on development of framework design, while comprehensive experimental validation, being computationally intensive, is considered as ongoing work.

Future Work

Although the suggested MONAS framework design offers a solid base of multi-objective neural architecture search, there are a number of directions that can be pursued to improve its functionality. Hardware-Aware Optimization: Hardware-specific constraints (e.g., energy consumption, device-level latency) can be integrated into future work to allow deployment-oriented optimization to be more accurate. Hybrid Search Strategies: It could be beneficial to combine evolutionary algorithms with reinforcement learning or gradient-based NAS approaches to enhance search efficiency and convergence speed, as proposed in recent NAS research [1][6]. Real-Time Performance Evaluation: It is possible to enhance the accuracy of latency and efficiency estimates by extending the framework to encompass real-time profiling of architectures on target devices. Dynamic Objective Adaptation: Incorporating adaptive weighting of objectives based on user requirements or environmental conditions can further enhance flexibility. Scalability to Larger Search Spaces: It can be extended to larger search spaces that can encompass more complex architectural components and deeper networks to enhance the diversity and performance of generated models. These guidelines indicate that future studies can be done in integrating multi-objective optimization, effective assessment methods, and scalable search methods in the NAS field.

Acknowledgment

The authors would like to thank Prof. Mohini Avatade for her invaluable guidance and continuous support during the course of this work.

References

[1]. C. White et al., “Neural Architecture Search: Insights from 1000 Papers,” arXiv preprint arXiv:2301.08727, 2023.
[2]. K. Deb et al., “A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II,” IEEE Transactions on Evolutionary Computation, 2002.
[3]. A. Zela et al., “Surrogate NAS Benchmarks: Going Beyond the Limited Search Spaces of Tabular NAS Benchmarks,” ICLR, 2022.
[4]. R. Cai and J. Luo, “Multi-Task Learning for Multi-Objective Evolutionary Neural Architecture Search,” IEEE CEC, 2021.

International Research Journal on Advanced Engineering Hub (IRJAEH)
e ISSN: 2584-2137
Vol. 04 Issue: 06 June 2026
Page No: 4205 - 4212
https://irjaeh.com
https://doi.org/10.47392/IRJAEH.2026.0545

[5]. E. Real et al., "Large-Scale Evolution of Image Classifiers," ICML, 2017.
[6]. T. Elsken et al., "Neural Architecture Search: A Survey," Journal of Machine Learning Research, 2019.
[7]. B. Zoph and Q. V. Le, "Neural Architecture Search with Reinforcement Learning," ICLR, 2017.
[8]. H. Liu, K. Simonyan, and Y. Yang, "DARTS: Differentiable Architecture Search," ICLR, 2019.
[9]. H. Pham et al., "Efficient Neural Architecture Search via Parameter Sharing," ICML, 2018.
[10]. E. Real et al., "Regularized Evolution for Image Classifier Architecture Search," AAAI, 2019.