CG-CANTS-N: A Versatile Graph-Based Framework for Scalable and Adaptive Problem Solving Across Domains

AbdElRahman ElSaid
elsaida@uncw.edu
University of North Carolina Wilmington
Wilmington, NC, USA

Travis Desell
tjdvse@rit.edu
Rochester Institute of Technology
Rochester, NY, USA

Abstract

Neural architecture search (NAS) and neuroevolution have emerged as key methods for designing artificial neural networks (ANNs). While several nature-inspired algorithms, such as Continuous Ant-Based Neural Topology Search (CANTS), have successfully automated the design of recurrent neural networks (RNNs), they suffer from certain limitations, including fixed search constraints and limited exploration strategies. This paper introduces Genetic Programming Collaborative Ant-Based Neural Topology Search (**CG-CANTS-N**), a novel graph-based NAS framework that employs multiple colonies of simulated ants which move through a continuous search space based on previously placed pheromones. The ant paths through the search space are used to construct graphs which are used as neural architectures. Both the individual ant agents and the ant colonies evolve over time using evolutionary strategies. CG-CANTS-N extends on CANTS by allowing more flexible graph structures, and by utilizing genetic programming functions (e.g., addition, multiplication, trigonometric functions) with trainable weights on graph edges as opposed to traditional neural network neurons. Key innovations include adaptive colony evaporation control, dynamic ant movement strategies, and cycle removal via depth-first search. We demonstrate that CG-CANTS-N is capable of designing graph based genetic programs for time series forecasting tasks which outperform existing state of the art methods.

CCS Concepts

• Computing methodologies → Neural networks; **• Mathematics of computing → Time series analysis**; **• Theory of computation → Bio-inspired optimization**.

Keywords

Ant Colony Optimization, Neural Architecture Search, Time Series Forecasting, Ant Colony Evolution, Swarm Intelligence

ACM Reference Format:
AbdElRahman ElSaid and Travis Desell. 2025. CG-CANTS-N: A Versatile Graph-Based Framework for Scalable and Adaptive Problem Solving Across Domains. In *Genetic and Evolutionary Computation Conference (GECCO '25 Companion), July 14–18, 2025, Malaga, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3712255.3726650

GECCO '25 Companion, Malaga, Spain

ACM ISBN 979-8-4007-1464-1/2025/07
https://doi.org/10.1145/3712255.3726650

1 Introduction

Designing optimal computational graph architectures is a critical challenge in modern machine learning. Traditional methods demand extensive manual tuning and expert knowledge, which often leads to suboptimal solutions—especially in high-dimensional or complex tasks. Neural Architecture Search (NAS) and neuroevolution techniques have emerged to address these challenges by automating the discovery process and reducing human bias.

Early NAS methods often relied on reinforcement learning [12] and evolutionary algorithms [11] to iteratively refine neural architectures. Although effective, these approaches typically require substantial computational resources and can suffer from slow convergence. Recent advances have introduced nature-inspired methods. For example, ANTS [4–6] leveraged ant colony optimization for RNN design, while EXAMM [1, 3, 9, 10] used neuroevolution techniques to scale these ideas for time series forecasting. The CANTS [2] method further extended these ideas by exploring continuous search spaces, although it remained limited by fixed movement rules and predefined architectural boundaries. Recent work has also shown that neural architecture search and graph based genetic programming can be performed with similar methodologies, by replacing neural components with genetic programming (GP) operations [7, 8].

Inspired by this and the adaptive and collaborative behaviors observed in natural ant colonies, our approach, CG-CANTS-N, leverages multiple parallel ant colonies that both compete and cooperate while exploring the search space. In our framework, some colonies work to broadly cover diverse regions, while others focus on refining promising candidate architectures. The synergy between exploration and exploitation in a multi-colony setting facilitates a more robust search process. Ultimately, this framework enables the evolution of flexible, interpretable computational graphs that are suitable for a wide range of applications. This work investigates utilizing this ant colony strategy which has been previously used for neural architecture search [2, 4–6] for automating the design of graph based genetic programs (GBGPs), by generalizing the ant-based paradigm to arbitrary computational graphs and utilizing GP operations while incorporating adaptive and collaborative optimization mechanisms.

2 Proposed Method: CG-CANTS-N

CG-CANTS-N constructs flexible computational graphs where each node performs a GP operation. Nodes are embedded within a unique four-dimensional space:

- x, y**:** Represent the two-dimensional spatial location within a layer, facilitating a clear mapping of the architecture.
- z**:** Encodes temporal information, where $z = 1$ corresponds to the current time step and $z = 0$ to the most previous time step for recurrency, thus modeling sequential dependencies.

- F**:** Denotes a discretized functional space; the set of available functions includes addition, multiplication, sine, cosine, tangent, tanh, sigmoid, inverse, and mean. Ant agents round their selected F coordinate to the nearest function value.

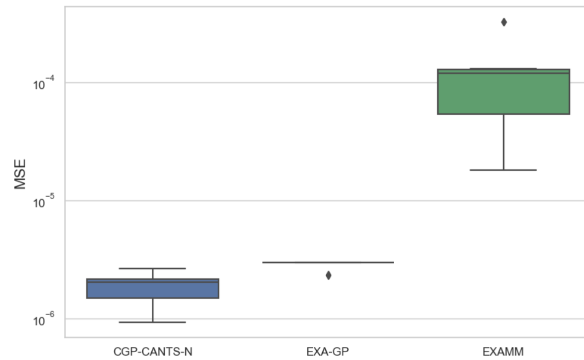
This 4D representation allows our method to capture both spatial and temporal patterns efficiently, enabling more nuanced architectural exploration.

Adaptive Colony Behavior. To maximize the effectiveness of the search, CG-CANTS-N employs an adaptive evaporation control mechanism. During the initial search phase, colonies suppress pheromone evaporation to ensure that every region of the search space is visited at least once. Once a sufficient level of exploration is achieved, the standard evaporation rate is reinstated. Simultaneously, a Particle Swarm Optimization (PSO) framework dynamically tunes colony parameters—including ant count, evaporation rate, and mortality rate—to balance exploration and exploitation effectively.

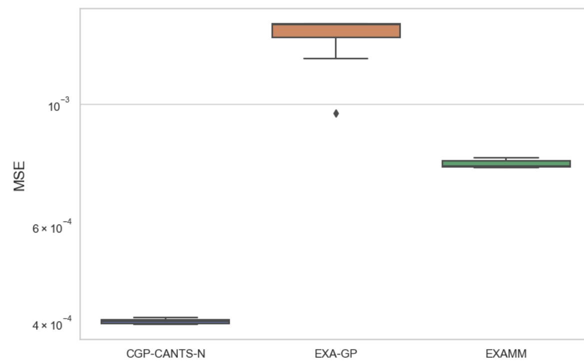
Ant Movement and Exploration Strategy: Ant agents in CG-CANTS-N are allowed to move in both the spatial (x, y) and temporal (z) dimensions. After each step, an ant will deposit a pheromone in the search space. Additionally, ants compute the center-of-mass of local pheromone traces and deliberately move towards the opposite direction, thereby ensuring a systematic coverage of the search space and reducing the risk of getting trapped in local optima. This flexibility enables the exploration of nodes (generated by clustering pheromone deposit locations) representing both present and past information. To avoid temporal inconsistencies (e.g., using future inputs), recurrent nodes are automatically introduced at the appropriate z-level.

Node Consolidation and Cycle Removal: The approach employs DBSCAN clustering to merge pheremone deposit positions into graph nodes that are in close proximity and share similar pheromone intensities, thereby reducing redundancy in the computational graph. Ant paths are used as edges between these nodes. However, merging nodes and utilizing these edges can sometimes result in cycles. To address this, a depth-first search (DFS) algorithm is applied to detect and remove redundant back-flowing edges, ensuring that the final graph is a directed acyclic graph (DAG) suitable for effective forward propagation.

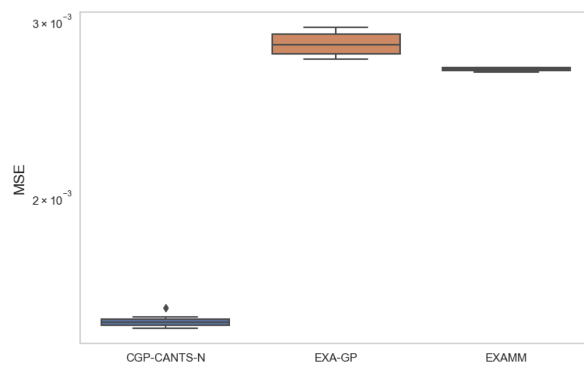
Edge Weight Optimization: Once the graph structure has been refined, edge weights are fine-tuned using a short backpropagation phase (typically 10 epochs). This step quickly adjusts the strength of the connections between nodes, optimizing the network for the target task without incurring excessive computational cost.



(a) ***C172 E1CHT1***



(b) ***Burner Main Flame Intensity***



(c) ***Generator Average Power***

Figure 1: ***Boxplots of Repeated Experiments***

3 Results

3.1 Experimental Setup

We evaluated CG-CANTS-N on three time-series benchmark datasets:

- **Cessna C172 Aircraft:** 5061 data points, 31 features; target: Engine 1 Cylinder Head Temperature.
- **Coal-Fired Power Plant Burner:** 14,402 data points, 12 features; target: Main Flame Intensity.
- **Wind Turbine:** 14,184 data points, 123 features; target: Generator Average Power.

3.2 Collaborative Multi-Colony Optimization

A key feature of CG-CANTS-N is its multi-colony design. Multiple colonies evolve candidate architectures independently while periodically sharing their best solutions and hyperparameter settings via a centralized environment process using MPI. This inter-colonial communication not only prevents premature convergence but also enhances the overall exploration, as colonies continuously update their strategies based on both local performance and global insights.

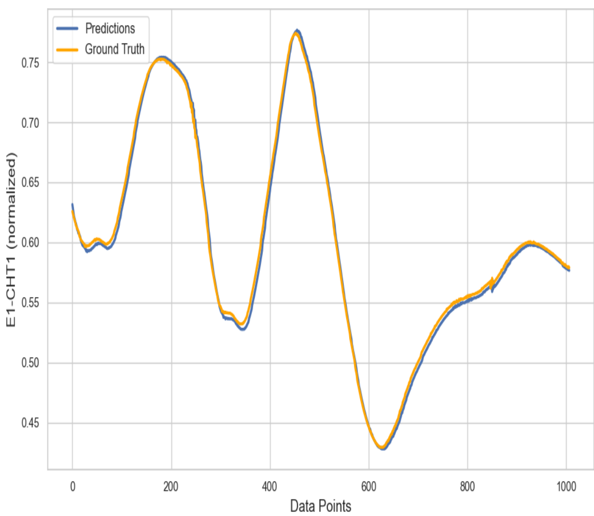


Figure 2: *Predictions: Cessna C172 E1CHT1*

3.3 Computational Complexity and Scalability

The overall computational complexity is influenced by factors including the number of iterations (I), graph size (N), dataset size (D), and the number of workers (W). Additionally, the dynamic pheromone management introduces a cost proportional to the number of ants (A) and pheromone points (R), while the graph consolidation step adds approximately G^2 complexity for G nodes. Empirical results indicate that although the dominant cost in data-intensive tasks is $\frac{I \cdot N \cdot D}{W}$, the combined overhead remains manageable, ensuring scalability in complex, high-dimensional scenarios.

Each experiment was repeated 10 times for statistical significance. We initialized 20 colonies, with each colony generating 1,000 candidate graph architectures using 201 CPU cores over four days. Parameter exchanges (e.g., ant count, evaporation rate) occurred every two generations through a centralized MPI process, ensuring robust coordination across colonies.

Table 1: Average Prediction Errors (10 Repeats)

Dataset	CG-CANTS-N	EXA-GP	EXAMM
Cessna C172	1.87×10^{-6}	2.86×10^{-6}	1.31×10^{-4}
Burner	4.02×10^{-4}	1.32×10^{-3}	7.77×10^{-4}
Turbine	1.50×10^{-3}	2.85×10^{-3}	2.69×10^{-3}

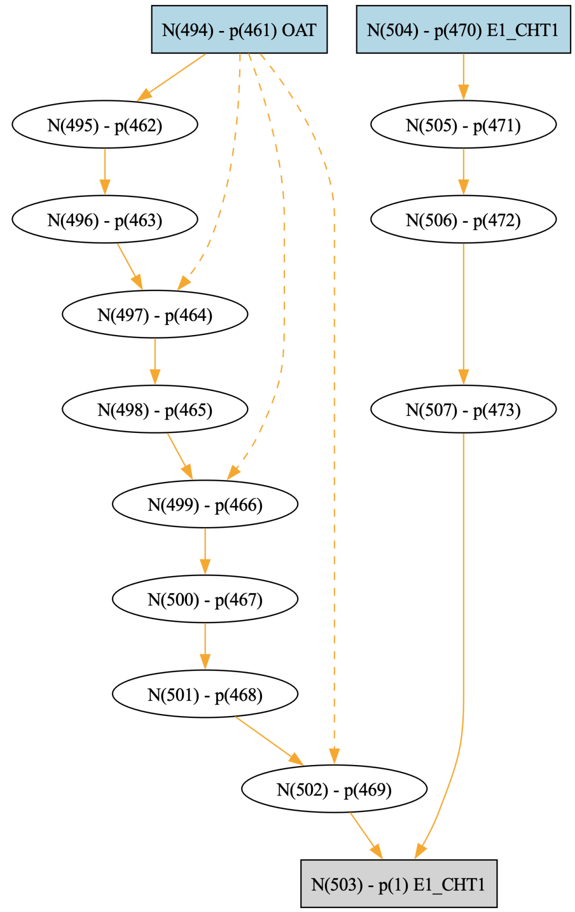


Figure 3: Graph: Cessna C172 E1CHT1

$$\begin{aligned} f(x) =& sin(sigmoid(sigmoid(cos(tanh(tan(sigmoid(\\ & tan(OAT)) + OAT)) * 1.0001 + OAT) * 0.9996) * 0.9986) \\ & * 0.9942 + OAT * 0.9945) * 0.0413 + \\ & sin(((E1\_CHT1 * 0.9789) * 0.9789) * 0.9789) * 0.9312)/2 \end{aligned} \tag{1}$$

3.4 Performance Evaluation

CG-CANTS-N consistently produced lower prediction errors than EXA-GP and EXAMM. For example, on the Cessna dataset the average error was 1.87×10^{-6} compared to 2.86×10^{-6} (EXA-GP) and 1.31×10^{-4} (EXAMM), state of the art benchmark algorithms for these datasets. Similar performance improvements were observed on the Burner and Wind Turbine datasets, as summarized in Table 1. Boxplots in Figure 1 illustrate that CG-CANTS-N achieves lower median errors with reduced variability. Representative computational graphs (Figures 3 and 4) and prediction plots (e.g., Figure 2) further confirm the method's strong performance and interpretability.

3.5 Colony Dynamics and Exploration

Figure 5 illustrates colony trait trajectories over 1000 generations, depicting the interplay between exploration and exploitation driven

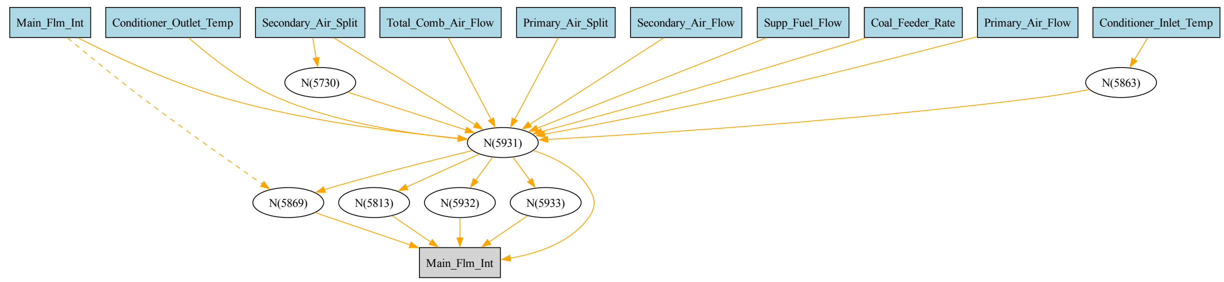


Figure 4: *Graph: Power Plant Burner*

by PSO. Marker shapes represent individual colonies, and a logarithmic color gradient indicates progression through generations. This collaborative dynamic enables colonies to refine hyperparameters and enhance overall search robustness.

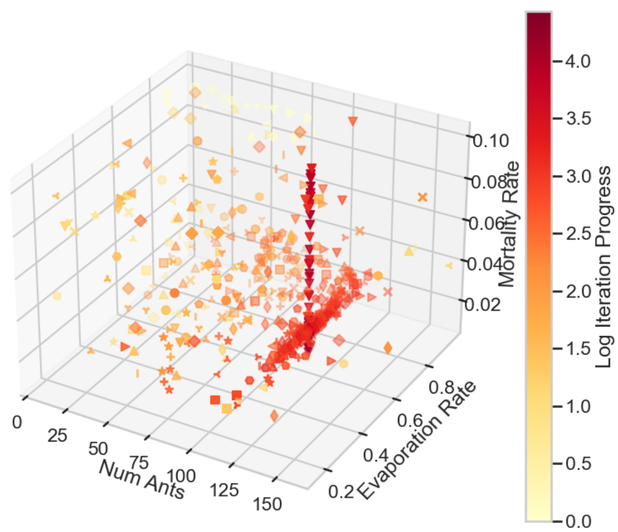


Figure 5: *Colony Trajectories: PSO-driven trait evolution over 1000 generations.*

4 Summary and Discussion

We have presented CG-CANTS-N, a collaborative graph-based NAS framework that extends Genetic Programming Collaborative Ant-Based Neural Topology Search by evolving flexible computational graphs using simple node functions, adaptive colony dynamics, and robust exploration strategies. Our multi-colony design—with features such as adaptive pheromone evaporation, directed ant movement, DBSCAN node consolidation, and DFS cycle removal—yields architectures that are both efficient and interpretable. Experimental results on three diverse time-series datasets demonstrate significant improvements in prediction accuracy and consistency over established methods.

Future work will focus on improving computational efficiency (e.g., transitioning to a C++ implementation) and exploring more advanced agent decision-making models using active inference. Extending the framework to other network types, such as CNNs and NLP architectures, represent other exciting directions for further research.

References

[1] Travis Desell. 2018. Accelerating the evolution of convolutional neural networks with node-level mutations and epigenetic weight initialization. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, 157–158.

[2] Abdelrahman Elsaid. 2024. Colony-Enhanced Recurrent Neural Architecture Search: Collaborative Ant-Based Optimization. *arXiv preprint arXiv:2401.17480* (2024).

[3] AbdElRahman ElSaid, Steven Benson, Shuchita Patwardhan, David Stadem, and Travis Desell. 2019. Evolving recurrent neural networks for time series data prediction of coal plant parameters. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer, 488–503.

[4] AbdElRahman ElSaid, Alexander G Ororbia, and Travis J Desell. 2020. Ant-based Neural Topology Search (ANTS) for Optimizing Recurrent Networks. In *International Conference on the Applications of Evolutionary Computation (Part of EvoStar)*. Springer, 626–641.

[5] AbdElRahman ElSaid, Brandon Wild, Fatima El Jamiy, James Higgins, and Travis Desell. 2017. Optimizing LSTM RNNs using ACO to predict turbine engine vibration. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. ACM, 21–22.

[6] AbdElRahman A. ElSaid, Alexander G. Ororbia, and Travis J. Desell. 2019. The Ant Swarm Neuro-Evolution Procedure for Optimizing Recurrent Networks. arXiv:1909.11849 [cs.NE]

[7] Jared Murphy and Travis Desell. 2024. Minimizing the EXA-GP Graph-Based Genetic Programming Algorithm for Interpretable Time Series Forecasting. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 1686–1690.

[8] Jared Murphy, Devroop Kar, Joshua Karns, and Travis Desell. 2024. EXA-GP: Unifying Graph-Based Genetic Programming and Neuroevolution for Explainable Time Series Forecasting. In *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. 523–526.

[9] Alexander Ororbia, AbdElRahman ElSaid, and Travis Desell. 2019. Investigating Recurrent Neural Network Memory Structures Using Neuro-evolution. In *Proceedings of the Genetic and Evolutionary Computation Conference* (Prague, Czech Republic) *(GECCO '19)*. ACM, New York, NY, USA, 446–455. doi:10.1145/3321707.3321795

[10] Alexander Ororbia, Ahmed Ahmed Elsaid, and Travis Desell. 2019. Investigating Recurrent Neural Network Memory Structures using Neuro-Evolution. arXiv:1902.02390 [cs.NE]

[11] Esteban Real, Sherry Moore, Andrew Selle, Saurabh Saxena, Yutaka Leon Suematsu, Jie Tan, Quoc Le, and Alex Kurakin. 2017. Large-scale evolution of image classifiers. *arXiv preprint arXiv:1703.01041* (2017).

[12] Barret Zoph and Quoc V Le. 2017. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* (2017).