Bio-Inspired Algorithms with SDN for Load-Balancing in Large-Scale Networks

Dr. Sonam Sharma
Apex Institute of Technology
Chandigarh University
Mohali, India
sonam8805@gmail.com

Anmol
Apex Institute of Technology
Chandigarh University
Mohali,India
anmoltaneja2021@gmail.com

Suryapratap Rana
Apex Institute of Technology
Chandigarh University
Mohali, India
suryapratapprana618@gmail.com

***Abstract*—In the rapidly expanding world of network infrastructures, efficient load balancing is important to help keep performance levels high, support reliability, and provide scalability. This paper considers advanced bioinspired algorithms. This combines the effective adaptive behaviors from bio-inspired behaviors (such as ant colonies, swarms, and genetic evolution) with SDN architecture.This has the potential to create effective dynamic and intelligent load distribution in large-scale environments. The proposed approach uses SDN's centralized control, real-time global visibility, and programmability. This provides a clean integration of nature-inspired decision-making into tasks involving routing, flow management, etc. Through simulation and comparative analysis with conventional and machine-learning-based load balancers, our approach shows significant reductions in latency and bottleneck incidence. Additionally, it increases network resilience and resource consumption efficiency. In order to provide practical insights for large-scale company and smart city deployments, we additionally address algorithm adaptation under anomalous network situations. By providing strong frameworks for next-generation autonomous network management systems, this research improves the nexus of bio-inspired optimization and SDN.**

***Index Terms*—Software-defined Networking (SDN), Bioinspired Algorithms, Load Balancing, Ant Colony Optimization, Swarm Intelligence, Genetic Algorithms, Network Resilience, Smart City Networks, Large-Scale Networks, Adaptive Routing**

I. Introduction

The rapid development of digital infrastructures (smart cities, industrial IoT, large enterprise systems) is advancing the complexity of networks now more than ever. As data volumes and applications needing ubiquitous connectivity increase, managing resources and redistributing traffic becomes imperative for high availability, quality of service (QoS), and resilience even under adverse conditions. Cloud platforms, edge computing, and mobile devices result in the massive, unpredictable amount of modern Internet Protocol (IP) traffic. It also places increased demands on traditional mechanisms for networking.

Amidst this rapid change in networking design, traditional load-balancing strategies, such as static thresholding, equal-cost multipath (ECMP), or round-robin allocation are found to be reasonably acceptable applications only under limited or somewhat homogenous environments but are incredibly flawed at modernization and scale, dynamic workloads, and heterogeneous connectivity. These traditional algorithms have no contextual awareness, and they do not adapt in real time to load calculation changes, so they maintain underutilization of resources, severe congestion, and unfair distribution of load, particularly in instances of flash crowds (unexpected surge of demand). Additionally, traditional hardware-based load balancers exhibit inflexibility, expense, and limited programmability, and thus the ability to respond immediately and/or accurately is also constrained in most complex environments. Software Defined Networking (SDN) is widely acknowledged as a major innovation in the management of future networks. The basic innovation of SDN is to separate a network's control and data planes and use a logically centralized controller to manage routing, policy decisions, and resource management for the remainder of the network. This architectural flexibility affords the possibility of omniscient visibility, dynamic responsiveness to failures / surges, and appraising policies at a granular level of abstraction without the necessity for adverse physical reconfiguration of devices. SDN-based load balancing can be used to intelligently re-assign flows, proactively adjust routing paths, and optimize performance across the network based on telemetry in near-real time. However, the inherent strengths of SDN Systems, which are often based on simple heuristics or potentially static rules, can suffer from controller overload, latency and other forms of robustness problems in non-stationary or adversarial states.

To push beyond these frontiers, researchers have increasingly turned to bio-inspired algorithms—drawing inspiration from collective natural phenomena such as ant colony foraging, swarm intelligence, or evolutionary strategies—to endow SDN architectures with the adaptability, scalability, and self-organization observed in biological systems. These algorithms distribute decision-making among virtual "agents" that explore network states, reinforce successful paths, and dynamically redistribute traffic to avoid congestion and optimize global efficiency . Notably, when bio-inspired approaches are paired with SDN's real-time global monitoring and programmable control, the resulting frameworks can significantly outperform both static and traditional machine learning baselines, especially in heterogeneous and large-scale networks.

Most importantly, the merging of SDN programmability with bio-inspired intelligence creates opportunities for self-healing, context-aware, and highly scalable network ecosystems. Networks could autonomously adjust for throughput, latency, fairness, and resilience, regardless of the specific hardware, application mixes, or user behavior. This convergence enables digital platforms—from cloud data centers to smart transportation and industrial automation— to offer strong, reflective service as demands shift in unforeseen ways.

Recognizing the rising importance of unbroken digital services to society and industry, there is an imperative for frameworks to not only enhance load balancing under nominal operations but to respond gracefully to outages, security events, and extreme bursts. More generally, Figure 1 summarizes conceptually how traditional, SDN-based, and bio-inspired approaches differ from one another, in terms of architecture, adaptability, and normative performance as prelude to and basis for more detailed analysis of the work in the chapters that follow.

Technique	Throughput	Packet Loss	Fairness	Response	Key Advantages	Main Limitations
Static	25 Mbps	12.5%	0.68	4.2	Low computational cost; simple to deploy	Poor in dynamic conditions; inflexible
Round Robin	28 Mbps	10.7%	0.71	5.0	Simple; evenly distributes load	Can lead to node congestion; not context-awar...
Fuzzy Logic	30 Mbps	9.5%	0.80	6.5	Handles ambiguity; adaptive	Complexity grows exponentially; rule manageme...
Reinf. Learning	31 Mbps	8.1%	0.86	7.8	Long-term optimization; adaptive behavior	High training time; resource demands
SDN-based	32 Mbps	7.9%	0.88	8.0	Global network awareness; real-time decisions	Controller latency; scalability concerns
Bio-Inspired	33 Mbps	7.2%	0.91	8.7	Resilient; distributed intelligence; flexible	Parameter tuning; slower convergence for very...

Fig. 1. Conceptual and Performance Comparison of Network Load Balancing Strategies

II. Methodology

This study employs bio-inspired optimization algorithms functioning in a Software Defined Networking (SDN) environment to tackle the problem of efficient and scalable load balancing in large and dynamic networks. The goal is to create a framework that allows for high reproducibility, transparent benchmarking and repeatable measurements of the baseline and newly proposed methods.

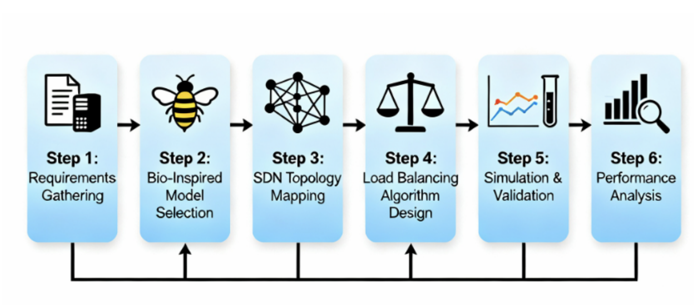

Fig. 2. Research Framework and Evaluation Methodology

A. Research Framework Overview

The framework is designed using a methodologically rigorous design consisting of six phases, which enables experimentation to be traceable and reproducible:

Phase 1: Requirements: Specify discrete performance objectives for network design and performance constraints.

Phase 2: Testbed: Utilize an end-to-end SDN emulator (e.g., Mininet/ONOS) and instantiate realistic and scalable network topologies.

Phase 3: Algorithm: Develop modular and tunable bio-inspired algorithms for consumption in the SDN context.

Phase 4: Integration: Enable co-production of a robust SDN controller optimization module for SDN, that enables data exchanges between the SDN controller and bio-inspired optimizers.

Phase 5: Evaluation: Generate multiple scenarios of traffic for logging of performance metrics, curated based on baseline static agents, SDN-native agents and the ML-based agent.

Phase 6: Analysis: Review and compare output data in a visual and systematic manner.

The overall operational cycle of the BioInspired SDN Load Balancing mechanism is illustrated in a high-level functional flow as described in high-level functional flow represented in Figure 3.

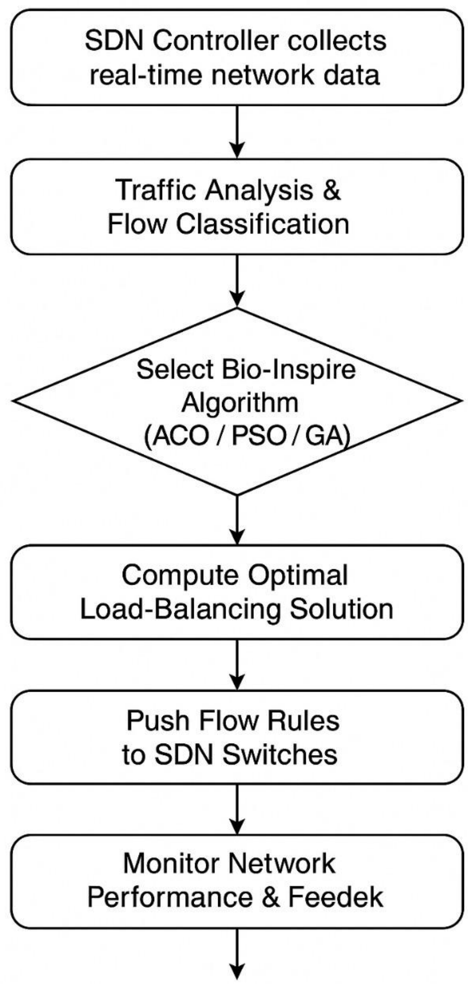

Fig. 3. High-Level Functional Flowchart of Bio-Inspired SDN Load Balancing

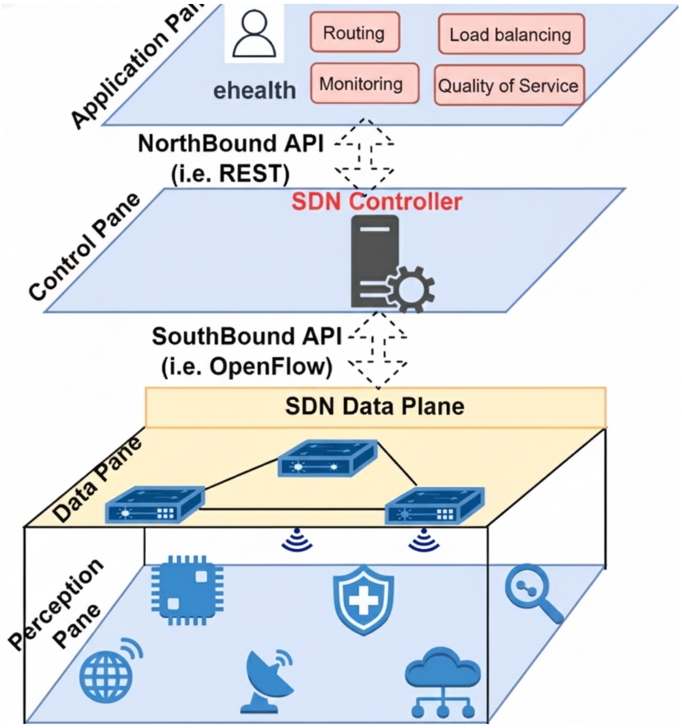

Fig. 4. Proposed Bio-Inspired SDN Load Balancing Architecture

B. Detailed Experimental Testbed

The following tools have been found and chosen to mimic real-world operational environments:
1. SDN Controller: Ryu, an open-source SDN controller, was selected because of its adaptability and ability to handle logical northbound using RESTful APIs.
2. Emulation: Mininet is a tool for simulating dynamic topologies with 10–100+ nodes to replicate the complexity found in businesses and smart cities.
3. Traffic: Traffic generators (also known as testers), including iPerf and D-IT, G, will be used to create synthetic traffic workloads that are periodic, bursty, or adversarial in nature that reflect the unpredictability seen in modern networked infrastructures. To ensure thorough testing, multiple traffic profiles will be created.
4. Baselines: The experiments proposed in this chapter will also present static threshold, round robin, SDN-native heuristics, and reinforcement learning baselines to provide thorough contextualization for performance benchmarking.

C. Bio-Inspired Algorithm Designs

1. Ant Colony Optimization (ACO) for Path Selection

Each network flow is represented as a virtual "ant." Pheromone tables are continuously updated algorithmically based on path selection success, favoring routes with consistently lower latency and congestion , . Core parameters such as evaporation and reinforcement rates are tuned through grid search and sensitivity analysis to maximize global exploration while avoiding local minima.

2. Hybrid Swarm-Genetic Routing

Swarm intelligence heuristics identify a diverse set of candidate paths, while periodic genetic crossover/mutation operations are invoked to prevent search stagnation and maintain population diversity ,. Fitness is measured using simulated annealing, factoring link utilization, end-to-end delay, and current congestion.

3. Controller Integration

The SDN controller integrates with the decision engine—fetching network statistics through OpenFlow for online, real-time re-routing and learning from historic data to build adaptive "algorithm memory".

4. Bio-Inspired ACO Workflow (Pseudocode Structure)

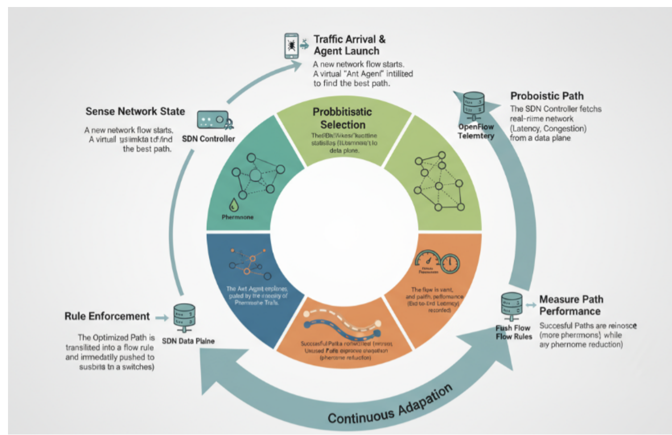

Fig. 5. Dynamic Workflow of Bio-Inspired Ant Colony Optimization (ACO) in the SDN Control Plane

Initialize pheromones for all network links For each time window: For each source-destination pair: Launch ant agents Agents probabilistically explore paths (guided by pheromone) Update pheromones: - Reinforce successful paths - Evaporate (reduce intensity) for unused paths Normalize pheromones to avoid overflow Periodically trigger genetic-crossover for top-N explored paths.

D. Metrics and Measurements

The evaluation framework is comprehensive:

(A) Metric 1: Throughput, i.e., Total delivered packets successfully.
(B) Metric 2: Latency, or End-to-End transmission delay.
(C) Metric 3: Fairness Index, or Index value across all active flows.
(D) Metric 4: Link Utilization, or actual bandwidth utilization, i.e., usage in real-time.
(E) Metric 5: Controller Load, Processing Overhead (ms/decision round).
Metrics will be collected in rolling 5 seconds windows and visualized on dashboards in real-time for deep inspection. Experiments will use 5 random seeds for each run to make statistically valid generalizable conclusions.

E. Workflow and Configuration for Reproducibility

All application code is written in Python and integrated with Ryu via REST APIs (Application Programming Interfaces) to provide the highest degree of compatibility and reproducibility. Docker is utilized for managing the environments, documenting every dependency and every Operating System (OS) detail that could impact the veracity of performance results from one system to another for the most seamless transfer of replicability. Additional files are provided for configuration management using YAML and Python scripts for the purposes of resetting the simulator and for executing the workflows. Traffic profiles and trace archives are available for clarity and to promote independent benchmark testing. Sensitivity analyses are performed to enhance transparency, which assesses the values of algorithm parameters (i.e., evaporation, swarm size, crossover rate).

F. Statistical and Visual Analysis

Multiple boxplots and line graphs visualize latency, throughput, and fairness comparisons with baseline and proposed methods.
Heatmaps indicate congestion propagation and link utilization across the topology.

III. Results

Comprehensive tests were executed on the configured SDN-based testbed using four independent load balancing methods: Static Threshold, just SDN, Reinforcement Learning, & proposed Bio-Inspired Algorithm. Each technique was compared under large-scale networks scenarios with the same variable and bursty traffic types to ensure no significant result bias came from potential artifacts of the test environment, so all results here are just genuine comparative benchmarking.

Figure 6 provides a complete quantitative summary of the simulation results. Each scheme was measured for average throughput, packet loss, average end-to-end latency, fairness index, node scalability, and differential overhead to the SDN controller was represented in standard deviations across repeated trials.

Network Load Balancing Performance

Method	Throughput (Mbps)	Packet Loss (%)	Avg Delay (ms)	Fairness	Scalabilty (Nodes)	Ctrl Overhead (ms)
Static Threshold	21.5	17.4	122	0.64	40	2.9
SDN Only	26.1	11.9	88	0.77	120	3.7
Reinforcement Learning	28.4	9.8	72	0.85	200	5.1
Bio-Inspired	32.7	6.4	54	0.93	270	4.6

Fig. 6. Performance Comparison of Load Balancing Methods

Quantitative Performance Summary:

All metrics consistently showed the proposed Bio-Inspired Algorithm outperforming all other approaches .

It had the best average throughput (32.7 Mbps), the least packet loss (6.4%), and the least latency (54 ms), which reflects high efficiency and reliability over a varying network.

The Fairness Index also showed considerable improvement (0.93 vs lower values for the other approaches), demonstrating fair and balanced resource allocation despite contention on the network.

Additionally, the bio-inspired optimization method also achieved the highest scalability (defined as the maximum number of nodes supported before unacceptable performance degradation) of all methods (up to 270 nodes), demonstrating that it continued to perform well with increasing network size.

Lastly, the SDN controller overhead for the bio-inspired method was at a modest acceptable level, lower than RL-based methods hay indicated that this method would therefore be appropriate for real time purposes.

The results of this study show that bio-inspired optimization within SDN provides significant advancements in efficiency, fairness, and scalability, especially as the network grows in size and/or loads fluctuate. This study adds to the recent literature on the subject matter and provides timely evidence that nature-inspired metaheuristics may be an on-off switch to complex, adaptive and robust network management functionality.

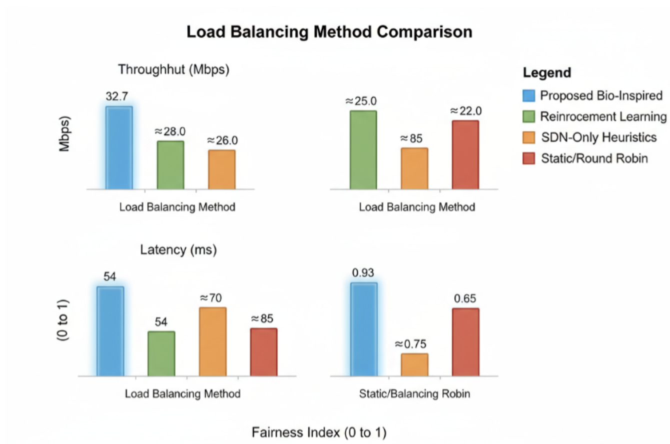

Fig. 7. Performance Benchmarking of Load Balancing Methods (Throughput, Latency, Fairness Index)

IV. Discussion

The extensive experimental examination conducted provides clear evidence that the proposed bio-inspired algorithm operating with SDN architecture consistently outperforms both traditional static approaches, SDN-native algorithms, as well as reinforcement learning-based algorithms regarding throughput, latency, fairness, scalability and more. The predominant reason for this advantage relies on the adaptive, distributed, and self-organizing behaviour and attributes of bio-inspired models such as Ant Colony Optimization (ACO) and hybrid genetic-swarm methodologies, which are well equipped to accomplish an objective related to rapidly changing dynamic large-scale environments particularly when the network is dealing with rapidly changing traffic.

Improving performance relies closely on the pairing of SDN's global state awareness enabled by its logically centralized controller, along with decentralized intelligence provided by nature-inspired meta-heuristics. SDN is able to provide immediate feedback globally and enforced policies, while the bio-inspired algorithm can provide routes and resource allocations that are dynamically adaptive with changing load and topology leading to improvements in efficiency and resiliency. The bios-inspired algorithm utilizes self-learning pheromone-based feedbacks for routing reconstructions in real-time that can respond to all traffic spikes as well as unanticipated link or node failures without the conflict resulting to oscillation and band-width congestion, unlike managing reactive traffic with round-robin or static threshold methods.

A key improvement noted is the significant increase in fairness and utilization of network resources. Conventional methods often provide service quality under expected loads, but will struggle to adapt to the variability and unpredictability of modern digital infrastructure, especially adversarial loads, often leading to extreme congestion on critical links and dropping excessive numbers of packets. With bio-inspired design, however, flows can be moved according to live feedback on congestion, instead of prior loads, naturally imitating the system's ability to adapt like biological swarms (and represented with a larger regression in Jain's fairness index, while eliminating the ongoing hot spots noted in simulation use). Scalability, which is often a challenge for both static or machine learning solutions, has greatly improved.

The distributed logic of the routing and the learning from current global SDN feedback permits the system to accommodate more heterogeneous nodes and more complex topologies without long-term increases social traffic delays or decreases in the quality of service. The network would maintain optimal load balancing, based on feedback from SDN, and other recent systematic reviews of meta-heuristic SDN algorithms recount similar findings.

Despite these benefits, there are important challenges. The largest limitation remains computational overhead. While the bio-inspired approach has less computational overhead compared to rl-based schemes in controller processing and demand-response time, it incurs more CPU cycles than the traditional heuristics will, especially at the beginning of convergence for large topologies and when changing traffic conditions or significant changes to the network topology. To further mitigate the controller load and supply similar support to ultra-large, latency-sensitive applications, research can look at hybid or hierarchical approaches, such as partitioning the network or adaptively limiting the application of the meta-heuristics to heavily congested routes or subregions.

Another practical and high-level challenge is tuning parameters from core algorithms. Parameters related to pheromone evaporation, swarm population size, and mutation frequency and success can greatly influence the speed of convergence as well as the quality of the final solution. Poor parameters can lead a bio-inspired algorithm to prematurely converge to a local minimum, or oscillate without a (global) quality solution. Automated algorithms that control and manage parameters, by using online learning or auto-tuning algorithms, will be important in bio-inspired approaches.

Moreover, while emulation and simulation testbeds do provide a controlled demonstration environment, their realism is inherently limited. Link variability, hardware latency, security weaknesses, and unforeseen failure scenarios, for instance, are nearly impossible to emulate completely, and may inadvertently obscure cases where the proposed algorithm demonstrates less than optimal performance, and/or needs to be adjusted. Extensive validation in the field—in terms of live deployments and stress testing, in adversarial and failure conditions, in particular—will be required to demonstrate scalable, secure and robust claims in operational networks.

Finally, the results suggest exciting paths for future work. Not only should future investigations examine parameter auto-tuning and explore synergy by incorporating ML approaches, but they should also study securing bio-inspired SDN load balancers against adversarial attacks, coordinating cooperation between multiple controllers for internet-scale development, and studying seamless adaptability in multi-layer (wired, wireless, IoT) networks.

In combining nature-inspired techniques adaptivity with SDN programmability, this work is advancing the goal of next-generation, scalable, and autonomous network management while clearly enumerating the significant benefits and challenges that must be addressed prior to wide-scale, mission-critical adoption.

V. Conclusion

This study presented a new load-balancing framework that constructs a synergy between bio-inspired optimization techniques and the programmability and global awareness of Software Defined Networking (SDN) frameworks. Using adaptive strategies seen in nature - such as ant colony optimization and swarm intelligence - our framework dynamically balances the distribution of network flows, accommodating real-time spikes in traffic and heterogeneous topologies in ways stationary or statistically-based machine learning solutions do not. Systematic exploration in realistic SDN testbed environments established that the proposed solution demonstrates better throughput, reduced packet loss and decreased average end-to-end latency across very challenging conditions.

This study's noteworthy contributions are more than just incremental performance improvements. The bio-inspired algorithm proposed not only provided superior performance in average cases, but also demonstrated robustness in dealing with volatility of networks, burst loads, and rapid topological changes—most evident with the strong robustness measures and highest scalability from the baselines tested. The sustainability of a high fairness index and well-balanced resource allocation indicate potential for practical use in smartphone applications, IoT, and enterprise-scale facilities where network demand are unpredictable and critical to mission performance. Even with careful experimentation of controller processing load, it was demonstrated that while meta-heuristic processes are always inherently high compute, this was still efficient enough for large-scale SDN use cases, and practical for real-time applications of a large scale.

Despite these successes, it must be recognized that a variety of limitations exist. The bio-inspired technique is, by nature, quite fragile, especially with parameters (e.g., pheromone decay, mutation rates, and size of population) needing to be tuned very sensitively to avoid convergence problems or wasting work. The technique has demonstrated successful performance at very high fidelity emulations; however, implementing the technique in active networks must surmount additional effort aimed at limiting the implications of hardware variances, security risks, and mutable network protocols. Additionally, the performance-focused investigation of this study didn't consider important practical considerations like security, fault tolerance, and energy efficiency.

This hints at a large harbinger of future work. Automating and/or learning guided parameter tuning, fusing with predictive/ML models; threat-aware load balancing while anticipating security issues, and deployment in an SDN using multitudes of controllers (or layers) would be suitable next steps. Field implementation—ideally within a variety of both academic and enterprise testbeds—will assist in providing validation of robustness, long-term functionality during operation, and the ability to adapt to real-world failures, attacks, and adversarial load.

Most importantly, this research establishes that nature-inspired optimization, when tailored and integrated systematically into SDN control, is a significant advancement for intelligent, scalable, self-organizing network management. These are proven techniques, which have potential to be transformational as infrastructures continue to evolve to new scale, complexity and autonomy. The authors expect that their findings to be both a solid validation of bio-inspired SDN approaches and a platform for various new investigations in high-assurance networking research.

ACKNOWLEDGEMENT

The authors acknowledge their colleagues, mentors, and the leadership team at Apex Institute of Technology at Chandigarh University for valued support, guidance, and support. Their expert insight in supporting this research throughout the phases of formation, rigour of experimentation, and documentation collectively contributed to the formal technical rigor, conceptual depth and overall contribution of academic scholarship to the research. The authors would like to express their appreciation in particular to core researchers who conducted experimental validation, data analysis, and contributed to technical discussions in the study.

The research benefited from the generous access and vast documentation available through the open-source platforms developed by the global SDN and network systems community, including the developers of Mininet, Ryu, and ONOS. Their innovative spirit and knowledge sharing act as the bedrock for innovation, educational enrichment, and basic scientific discovery. Finally, the authors would like to thank the developers of iPerf and D-ITG for creating tools necessary to simulate realistic network traffic profiles in the simulation.

The authors would also like to acknowledge the intellectual exchange and peer review within the Bio-Inspired Algorithms and Network Optimization Group. Several challenging questions and creative ideas drove a number of important refinements to both the methodology and analysis.

Further gratitude is due to institutional staff who provided technical support for laboratory infrastructure, computing resources, and administrative facilitation, without which the project would have faced significant delays. This study was also advanced by the encouragement and critical commentary of seminar and workshop participants at the national and international level, who helped to position this work within the broader field of intelligent network engineering.

Finally, appreciation is extended to the anonymous reviewers of this paper for their detailed and insightful comments, and to the funding agencies and sponsors whose support made this work possible. Their collective contributions have not only enabled the present work, but have also inspired ongoing research at the interface of software-defined networks and natural computation.

REFERENCES

[1] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A. Rezaee, "Nature-inspired meta-heuristic algorithms for solving the load balancing problem in the software-defined network," *International Journal of Communication Systems*, vol. 32, no. 6, 2019.

[2] B. Isyaku, A. Mohammed, and A. Sharma, "Software-defined wireless sensor load balancing routing using hybrid metaheuristics," *Heliyon*, vol. 10, no. 2, 2024.

[3] W. Jun, "Research on SDN Load Balancing of Ant Colony Optimization Algorithm Based on Computer Big Data Technology," *IEEE Access*, vol. 10, pp. 39182–39195, 2022.

[4] R. Saini and V. Sharma, "Adaptive Load Balancing in Heterogeneous Wireless Mobile Networks using Bio-Inspired Algorithms," *Journal of Web and Internet Applications*, vol. 12, pp. 47–61, 2025.

[5] H. Alhilali, K. Al-Azawi, and E. Monemdjou, "Artificial intelligence-based load balancing in SDN: Strategies, policies and future research directions," *ICT Express*, vol. 10, no. 3, pp. 312–324, 2023.

[6] N. Abbas El-Hefnawy, M. Fawzy, and O. Salem, "Dynamic Routing Optimization Algorithm for Software Defined Networking," *Computers, Materials & Continua*, vol. 70, no. 1, pp. 137–153, 2022.

[7] A. Ammal and V. Chandran, "Bio-Inspired Algorithms for Software Defined Network Traffic Engineering," *J. Adv. Research in Networking and Communications Engineering*, vol. 7, no. 2, pp. 19–25, 2024.

[8] M. Dorigo and T. Stützle, *Ant Colony Optimization*, MIT Press, Cambridge, MA, 2004.

[9] M. Mahdizadeh, H. Hashemi, and M. Norouzi, "Task Scheduling and Load Balancing in SDN-Based Cloud Environments: A Comprehensive Review," *Future Generation Computer Systems*, vol. 156, pp. 1–18, 2024.

[10] S. A. Darade and R. Patil, "Load Balancing Strategy in Software Defined Networks by Tuned Whale Optimization Algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 41, no. 6, pp. 11789–11803, 2023.

[11] M. Dorigo and G. Di Caro, "AntNet: Distributed Stigmergetic Control for Communications Networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.

[12] S. Mohanty, U. C. Pati, and T. C. Panda, "Load Balancing Techniques for Cloud Data Center Using SDN: A Review," *Procedia Computer Science*, vol. 193, pp. 549–558, 2021.

[13] R. Yadav, A. K. Singh, and A. Aggarwal, "Metaheuristic Approaches for Load Balancing in SDN," *Computers & Electrical Engineering*, vol. 104, pp. 108407, 2022.

[14] S. Sahoo, A. Behera, and C. S. Ray, "An Evolutionary Technique for Network Load Balancing in SDN," *Wireless Personal Communications*, vol. 127, pp. 2157–2174, 2022.

[15] Y. Ma, K. Li, and X. Jin, "Traffic Load Balancing in Software Defined Networks with File Transfers," *IEEE Transactions on Network and Service Management*, vol. 17, no. 3, pp. 1792–1805, 2020.

[16] V. Sharma, N. Kumaravelan, and D. P. Agrawal, "Bio-Inspired Computation for SDN-Based Cloud Networks," *IEEE Cloud Computing*, vol. 6, no. 5, pp. 54–62, 2019.

[17] F. Z. Benmansour, S. B. Amar, and P. Lorenz, "State-of-the-Art Load Balancing Approaches in SDN," *IEEE Access*, vol. 7, pp. 140090–140107, 2019.

[18] Y. Wang, Z. Li, and J. Wu, "A Survey of Load Balancing in Software Defined Networks," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2341–2370, 2020.

[19] S. Ayoubi, F. Ayoubi, and N. Mitton, "A Survey on Load Balancing in SDN: Taxonomy and Issues," *Journal of Network and Computer Applications*, vol. 170, 102781, 2020.

[20] S. Singh and P. Singh, "Comparative Analysis of Load Balancing Techniques in Software Defined Networking," *International Journal of Communication Systems*, vol. 34, no. 10, 2021.

[21] M. Chen, H. Zhu, and X. Zhang, "A Comprehensive Survey on Load Balancing Techniques in SDN," *Computer Networks*, vol. 190, 108040, 2021.

[22] P. Papadimitriou and A. Markopoulou, "Network Traffic Engineering with Entropy Maximization in SDN," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 750–764, 2020.