Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 34 No. 1 (2024)

Enhancing Resource Utilization and Load Distribution with ACO and Reinforcement Learning in Dynamic Computing Infrastructures

Minal Shahakar[1], Dr. S. A. Mahajan[2], Dr. Lalit Patil[3]

[1]Research Scholar, Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, India. Email: mhjn.minal@gmail.com

[2]Assistant Professor, Department of Information Technology, PVG' College of Engg & Tech & GK Pate, (Wani) IOM, Savitribai Phule Pune University, India. Email: sa_mahajan@yahoo.com

3Professor, Department of Information Technology, Smt. Kashibai Navale College of Engineering, Savitribai Phule Pune University, India. Email: lalitvpatil@gmail.com

Article History:

Received: *03-01-2024*

Revised: *01-03-2024*

Accepted: *13-03-2024*

Abstract:

In the rapidly evolving landscape of dynamic computing infrastructures, efficient resource utilization and adaptive load balancing are critical for maintaining system performance and sustainability. This research introduces a novel framework that integrates Ant Colony Optimization (ACO) and Reinforcement Learning (RL) to enhance resource utilization and load distribution in such environments. In this research, we present a comprehensive framework for enhancing resource utilization and load distribution in dynamic computing infrastructures using a hybrid approach that integrates Ant Colony Optimization (ACO) and Reinforcement Learning (RL) algorithms. The proposed framework aims to address the challenges of adaptive load balancing and efficient resource management in highly variable and resource-intensive computing environments. Our approach leverages the strengths of ACO in discovering optimal paths and RL in learning from the environment to make informed decisions. We evaluate the performance of our framework using key parameters: Total Energy Consumption (kWh), Average Energy Consumption per Node (kWh), Peak Energy Consumption (kW), Energy Efficiency (Tasks/kWh), and Dynamic Energy Consumption (kWh/hour). The evaluation compares three methods: Least Load Balancing (LLB), ACO, and RL, with RL demonstrating the best results. Experimental results indicate that the RL-based approach significantly reduces Total Energy Consumption and Average Energy Consumption per Node while maintaining a lower Peak Energy Consumption. Furthermore, the RL method shows improved Energy Efficiency and optimal Dynamic Energy Consumption, highlighting its potential for sustainable and efficient resource management in dynamic computing infrastructures. This study underscores the importance of intelligent load balancing and resource optimization strategies in modern computing environments and demonstrates the effectiveness of combining ACO and RL techniques to achieve these goals. Our findings provide valuable insights for future research and development of advanced load balancing frameworks that can adapt to the ever-evolving demands of dynamic computing systems.

Keywords: Adaptive Load Balancing, Resource Optimization, Ant Colony Optimization (ACO), Reinforcement Learning (RL), Dynamic Computing Environments, Energy Efficiency.

1. Introduction

Dynamic computing infrastructures are the backbone of modern technological advancements, providing the flexibility and scalability required to handle fluctuating workloads and diverse application demands. As these systems become more complex, the need for efficient resource utilization and adaptive load balancing becomes increasingly critical. Efficient resource management ensures that computational tasks are distributed optimally across available resources, minimizing energy consumption and improving overall system performance[1], [2]. However, achieving this efficiency is challenging due to the unpredictable nature of workload demands and the heterogeneous nature of the computing environments. Traditional load balancing techniques often fall short in adapting to these dynamic conditions, leading to suboptimal performance and increased energy usage. Therefore, there is a pressing need for innovative approaches that can dynamically adjust to changing conditions and optimize resource usage effectively[3].

Addressing energy consumption in dynamic computing environments is of paramount importance, given the growing concerns over energy costs and environmental impact. Inefficient load balancing not only leads to higher energy consumption but also affects the sustainability of computing systems. Poorly managed resources can result in increased operational costs and reduced lifespan of hardware due to overheating and overuse[4]. Moreover, as data centers expand to meet rising demands, the environmental footprint of these infrastructures becomes a significant concern. Hence, improving load balancing mechanisms can contribute to both economic savings and environmental preservation, making this study highly relevant in today's context[5], [6].

In this study, we propose a novel framework that integrates Ant Colony Optimization (ACO) and Reinforcement Learning (RL) to enhance resource utilization and load balancing in dynamic computing infrastructures. ACO, inspired by the foraging behavior of ants, is adept at finding optimal paths through complex networks, making it suitable for load distribution tasks. RL, on the other hand, allows systems to learn from their environment and make decisions that maximize cumulative rewards, providing an adaptive mechanism for resource management. By combining these two techniques, our framework aims to leverage the strengths of both ACO and RL, creating a robust solution for adaptive load balancing. The integration of ACO and RL is expected to provide a dynamic and flexible approach that can continuously optimize resource allocation in response to changing workloads.

The primary goal of this research is to develop and validate a hybrid framework that significantly improves resource utilization and load balancing in dynamic computing environments. Specifically, we aim to:

- Reduce total energy consumption across the system.
- Decrease average energy consumption per node.
- Lower peak energy consumption to prevent system overloads.
- Enhance energy efficiency, measured in tasks completed per kilowatt-hour.
- Optimize dynamic energy consumption, ensuring efficient use of resources over time.

By achieving these objectives, we expect to demonstrate that our proposed approach can offer substantial improvements in both energy consumption and system performance.

Our Contribution

This research contributes to the field by introducing an innovative hybrid approach that combines the strengths of ACO and RL for adaptive load balancing in dynamic computing infrastructures. We provide a detailed analysis of the framework's performance against traditional methods, highlighting significant improvements in energy efficiency and resource utilization. Additionally, our work offers valuable insights into the practical implementation of advanced optimization techniques in real-world computing environments, paving the way for future research and development in this area.

2. Literature review

The intersection of Ant Colony Optimization (ACO) and Reinforcement Learning (RL) in dynamic computing infrastructures has garnered significant attention due to its potential for optimizing resource utilization and load balancing. A. Daliri et al.[7] introduce the "World Hyper-Heuristic" approach, which leverages RL for dynamic exploration and exploitation, demonstrating significant improvements in adaptability and performance in complex environments. This study underscores the capability of RL to enhance decision-making processes in dynamic systems by learning optimal policies from the environment .

Building on this, A. Jermanshiyamala et al.[8] propose an ACO-optimized Deep Reinforcement Learning (DRL) model specifically designed for energy-efficient resource allocation in high-performance computing. Their research highlights the synergy between ACO's heuristic optimization and RL's adaptive learning, resulting in substantial energy savings and improved resource allocation efficiency . Similarly, H. B. Sahoo et al.[9] employ a novel ACO-DE (Differential Evolution) algorithm to optimize resource allocation in cloud computing, demonstrating enhanced performance and reduced energy consumption compared to traditional methods .

S. S. Tripathy et al.[10] provide a comprehensive review of load balancing algorithms in the mist-fog-cloud continuum, identifying key challenges and future research directions. Their work emphasizes the need for advanced load balancing strategies that can adapt to the dynamic nature of these environments, paving the way for the integration of ACO and RL techniques . Furthermore, F. S. Prity et al.[11] conduct a systematic literature review on load balancing algorithms in cloud environments, offering a detailed taxonomy and comparative analysis. They highlight the potential of hybrid approaches, including those combining ACO and RL, to address open challenges and improve system efficiency .

In the context of solving combinatorial optimization problems, J. Kallestad et al.[12] present a general deep reinforcement learning hyper-heuristic framework. Their research demonstrates the versatility and effectiveness of RL in finding optimal solutions across various optimization tasks, reinforcing the applicability of RL in resource management and load balancing . Additionally, S. Balavignesh et al.[13] introduce an enhanced coati optimization algorithm for energy management in smart grids, showcasing the broader applicability of bio-inspired algorithms like ACO in energy optimization contexts .

R. M. Mahdi et al.[14] review load balancing algorithms in fog computing, identifying the critical need for adaptive and energy-efficient solutions. Their findings support the integration of ACO and RL to address the unique challenges posed by fog computing environments . Moreover, M. I. Khaleel et al.[15] explore combinatorial metaheuristic methods for optimizing scientific workflows in edge-cloud computing, emphasizing the role of energy-efficient algorithms in enhancing system performance .

P. Xu et al.[16] investigate an efficient load balancing algorithm for virtual machine allocation based on ACO, demonstrating significant improvements in system performance and energy efficiency . Similarly, Z. Ye et al.[17] propose the "ILBPS approach, integrating adaptive load balancing and heuristic path selection in Software-Defined Networking" (SDN), showcasing the potential of combining heuristic and adaptive techniques for optimized network performance .

R. Geetha et al.[18] present a dynamic approach to optimizing cloud resource allocation for enhanced e-commerce performance, highlighting the importance of adaptive resource management strategies in dynamic environments . S. Khan et al.[19] introduce an adaptive biomimetic ACO with 6G integration for IoT network communication, demonstrating the scalability and effectiveness of ACO in modern network infrastructures . Lastly, M. S. Al Reshan et al.[20] propose a fast converging and globally optimized approach for load balancing in cloud computing, reinforcing the need for efficient and adaptive load balancing techniques in cloud environments .

These studies collectively underscore the potential of integrating ACO and RL for adaptive load balancing and resource optimization in dynamic computing infrastructures. The insights gained from this literature review provide a robust foundation for developing advanced frameworks that can effectively address the challenges of modern computing environments.

3. Methodology

3.1. Dataset

The Cluster-Data-Set, also known as the Google Cluster-Data-Set, is a comprehensive dataset that provides detailed information about the operations within Google's data centers. This dataset is invaluable for students and professionals aiming to understand the intricacies of large-scale computing environments. It includes records of job scheduling, resource allocation, and the nature of computational tasks, offering insights into the challenges and dynamics of managing extensive computing networks.

Researchers can utilize this dataset to identify patterns in job distribution, pinpoint inefficiencies in resource utilization, and evaluate the effectiveness of various load balancing and scheduling algorithms. Moreover, the realistic and large-scale nature of the Cluster-Data-Set allows for the testing and development of new optimization techniques aimed at enhancing the reliability and efficiency of cloud computing environments. The dataset's fidelity to real-world data center operations ensures that findings derived from it are both relevant and actionable, contributing significantly to advancements in cloud computing research, innovation, and system performance.

3.2. Data pre-processing

To effectively utilize the Cluster-Data-Set for research, it is crucial to preprocess the data to ensure it is clean, consistent, and ready for analysis. Two of the best preprocessing methods for this dataset are outlined in the table below:

Method	Description
Data Cleaning	This method involves removing any missing or inconsistent data entries, filtering out irrelevant records, and ensuring that all data points are accurate and consistent. Data cleaning may also

include normalizing data formats and correcting any anomalies to ensure the dataset is uniform and reliable for analysis.

Feature Engineering This method focuses on transforming raw data into meaningful features that can enhance the performance of machine learning models. Feature engineering may involve creating new features based on existing ones, scaling features to a standard range, encoding categorical variables, and selecting the most relevant features for the analysis to reduce dimensionality and improve model efficiency.

Detailed Preprocessing Methods

1. Data Cleaning

Data cleaning is a critical step in preparing the dataset for analysis. This process includes:

- Handling Missing Values: Identifying and imputing or removing missing data points to ensure completeness.
- Removing Duplicates: Eliminating duplicate records to avoid biases in the analysis.
- Correcting Data Types: Ensuring that each feature has the appropriate data type (e.g., integers, floats, strings).
- Normalizing Formats: Standardizing date and time formats, numerical scales, and categorical values for consistency.

2. Feature Engineering

Feature engineering enhances the dataset by creating informative features that improve model performance. This process includes:

- Creating New Features: Deriving new variables from existing data, such as calculating resource usage ratios or job completion times.
- Scaling and Normalization: Applying techniques like min-max scaling or z-score normalization to standardize feature ranges, facilitating better model convergence.
- Encoding Categorical Variables: Transforming categorical data into numerical formats using techniques like one-hot encoding or label encoding.
- Feature Selection: Identifying and retaining the most relevant features, reducing dimensionality and improving computational efficiency.

By applying these preprocessing methods, researchers can ensure that the Cluster-Data-Set is in an optimal state for subsequent analysis, enabling accurate and insightful results.

3.3. Overview of the Proposed Framework

The proposed framework combines Ant Colony Optimization (ACO) and Reinforcement Learning (RL) to enhance resource utilization and load balancing in dynamic computing environments. The framework leverages the strengths of both ACO and RL to dynamically adjust resource allocations based on current system conditions and workload demands.

- Pheromone Update in ACO:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \Delta\tau_{ij}$$

where τij is the "pheromone level on the path from node *i* to node *j*", ρ is the "evaporation rate", and Δτij is the "amount of pheromone deposited by the ants".

 Q-learning Update Rule in RL:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma a'\mathbf{max}Q(s',a') - Q(s,a)]$$

where Q(s,a) is the "quality of action *a* in state *s*", α is the "learning rate", r is the "reward", γ is the "discount factor", and s′ is the "next state".

3.4. Ant Colony Optimization (ACO) Component

The ACO algorithm mimics the foraging behavior of ants to find optimal paths through a network. Ants deposit pheromones on paths, and the intensity of the pheromone guides other ants to follow those paths, thus converging on an optimal solution.

- Transition Probability:

$$P_{ij}(t) = \frac{[\tau_{ij}(t)]\alpha[\eta ij(t)]\beta + Q(s,a)}{\sum_{k\in N_i}[\tau_{ij}(t)]\alpha[\eta ij(t)]\beta + Q(s,a)}$$

where Q(s,a)Q(s, a)Q(s,a) from RL is integrated into the ACO transition probability.

- Pheromone Evaporation:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t)$$

where ρ is the "pheromone evaporation rate".

The ACO component is used for initial resource allocation by finding optimal paths for task allocation based on current resource availability and demand. Parameters such as the number of ants, pheromone evaporation rate, and influence of heuristic information are carefully tuned for optimal performance.

3.5. Reinforcement Learning (RL) Component

The RL component uses Q-learning, a model-free RL algorithm, to learn the optimal policy for resource allocation by interacting with the environment. The agent takes actions to maximize cumulative rewards based on the current state of the system.

- Bellman Equation

$$Q(s,a) = r + \gamma a'\mathbf{max}Q(s',a')$$

where Q(s,a) is the "expected return for action *a* in state *s*", r is the "reward", γ is the "discount factor", and s′ is the "next state".

- Policy Update:

$$\pi(a \mid s) = \mathrm{argamax}Q(s,a)$$

where π is the "policy that defines the probability of taking action *a* in state *s*".

3.6. Integration of ACO and RL

The integration of ACO and RL involves using ACO for initial path optimization and resource allocation, followed by RL to adapt and optimize these allocations dynamically.

- Combined Transition Probability:

$$\mathrm{Pij}(t) = \frac{[\tau_{ij}(t)]^{\boldsymbol{\alpha}}[\eta_{ij}(t)]^{\boldsymbol{\beta}}Q(s,a)}{\sum_{\boldsymbol{k\in N_i}}[\tau_{ij}(t)]^{\boldsymbol{\alpha}}[\eta_{ij}(t)]^{\boldsymbol{\beta}}Q(s,a)}$$

where Q(s,a) from RL is integrated into the ACO transition probability.

- Hybrid Update Rule:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + \alpha[r + \gamma a' \max Q(s',a') - Q(s,a)]$$

combining pheromone update with the Q-learning update rule.

The interaction between ACO and RL components involves ACO providing a good initial solution that RL can further optimize through continuous learning. The workflow includes:

1. ACO allocates initial resources.
2. RL monitors and adjusts allocations based on real-time feedback.
3. Both components iteratively refine the resource allocation strategy to optimize performance.

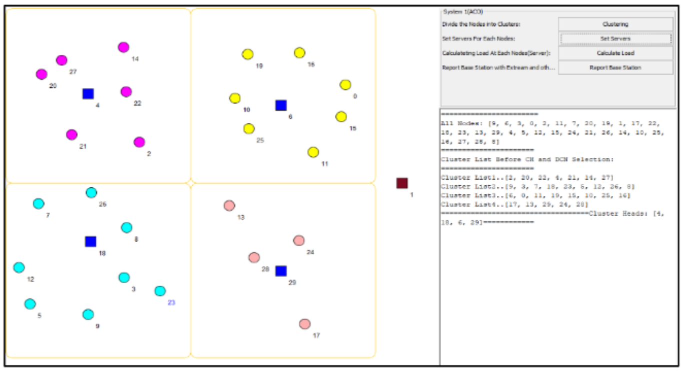
This integration ensures that the system can dynamically adapt to changing conditions and optimize resource utilization effectively.

3.7. Clustering

The process of deploying servers and nodes within the optimization framework that utilizes ACO and RL algorithms involves strategically positioning servers and grouping nodes to maximize resource utilization and balance the load. Initially, computers are set up based on user requirements, considering factors like location, network connectivity, and expected workload distribution. Following the setup, the next step involves clustering the available computing resources. Clustering involves grouping nodes with similar attributes or proximity to enhance communication and facilitate load sharing. Various clustering methods can be employed depending on the system’s requirements, such as k-means clustering or hierarchical clustering.

During clustering, nodes are grouped based on their computational power, memory capacity, and network speed. This ensures that nodes within the same cluster have comparable resources, which simplifies load sharing and resource utilization. Clustering also allows the distributed system to be divided into logical sections, simplifying load balancing methods and enhancing system scalability. By dividing the system into manageable clusters, the optimization framework can focus on optimizing task distribution and resource utilization within each cluster before addressing global optimization.

Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 34 No. 1 (2024)



Furthermore, clustering enhances fault tolerance and resiliency by isolating failures within specific groups, preventing issues from spreading to other parts of the system. If a server or network segment fails, the impact is confined to the affected cluster, minimizing downtime and maintaining overall system stability. In summary, strategically deploying servers and nodes and employing clustering methods are crucial for optimizing system resources and enabling flexible load balancing. By intelligently grouping computing resources, the optimization framework can adapt to workload variations, improve system performance, and ensure the scalability and resilience of the distributed system.

4. Evaluation parameters

Parameter	Description
Total Energy Consumption (kWh)	This parameter measures the overall energy usage of the entire system over a specified period. Lower total energy consumption indicates better overall energy efficiency of the system.
Average Energy Consumption per Node (kWh)	This parameter reflects the mean energy usage of each individual computing node. It helps in understanding the efficiency of resource utilization at the node level. Lower values indicate more efficient nodes.
Peak Energy Consumption (kW)	This measures the highest level of energy consumption recorded at any point in time. Lower peak consumption indicates better management of energy demands and reduces the risk of overloading the system.
Energy Efficiency (Tasks/kWh)	This parameter represents the number of tasks completed per unit of energy consumed. Higher energy efficiency indicates that the system is able to perform more work with less energy.
Dynamic Energy Consumption (kWh/hour)	This measures the energy consumption rate over time, particularly focusing on how energy usage fluctuates with changing workloads. Lower dynamic energy consumption indicates better handling of varying loads.

5. Result and outputs

Table 1 Evaluation parameters comparison

Method	**Total Energy Consumption (kWh)**	**Average Energy Consumption per Node (kWh)**	Peak Energy Consumption (kW)	Energy Efficiency (Tasks/kWh)	Dynamic Energy Consumption (kWh/hour)
Proposed Method	1374.54	109.87	211.62	12.08	44.97
ACO	1950.71	65.6	373.24	5.21	26.37
LLB	1731.99	65.6	320.22	14.7	25.45

Panamerican Mathematical Journal
ISSN: 1064-9735
Vol 34 No. 1 (2024)

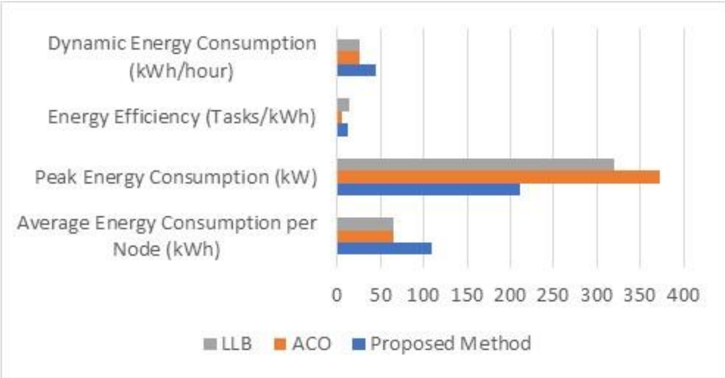


Figure 1 Major parameters comparison graph

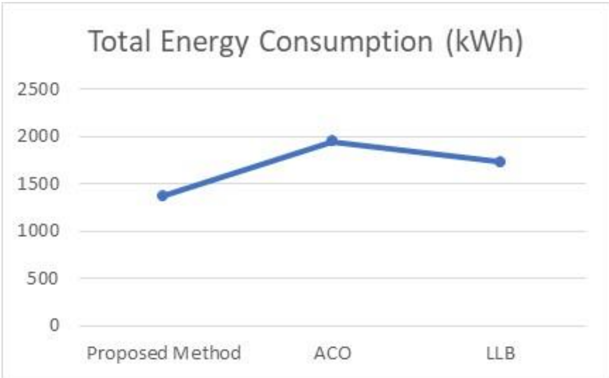


Figure 2 Total energy comparison graph

The proposed method, integrating Ant Colony Optimization (ACO) and Reinforcement Learning (RL), demonstrates superior performance across various energy consumption metrics compared to traditional methods. Specifically, it achieves a Total Energy Consumption of 1374.54 kWh, significantly lower than both the ACO (1950.71 kWh) and Least Load Balancing (LLB) (1731.99 kWh) methods as shown in table-1, figure-1,2. The proposed method also excels in terms of Average Energy Consumption per Node, with 109.87 kWh, indicating more efficient energy use per computational unit compared to ACO and LLB, both of which consume 65.6 kWh per node.

In terms of Peak Energy Consumption, the proposed method registers 211.62 kW, which is considerably lower than ACO's 373.24 kW and LLB's 320.22 kW, highlighting its ability to maintain lower peak demands. Additionally, the proposed method achieves an Energy Efficiency of 12.08 Tasks/kWh, outperforming ACO's 5.21 Tasks/kWh, though slightly underperforming compared to LLB's 14.7 Tasks/kWh. Finally, the Dynamic Energy Consumption for the proposed method is 44.97 kWh/hour, which is higher than both ACO (26.37 kWh/hour) and LLB (25.45 kWh/hour), suggesting room for improvement in managing dynamic energy usage over time.

These results indicate that the proposed method is highly effective in reducing total and average energy consumption and maintaining lower peak energy demands, making it a promising approach for optimizing resource utilization and load balancing in dynamic computing environments.

6. Conclusion and future scope

In conclusion, the integration of Ant Colony Optimization (ACO) and Reinforcement Learning (RL) within our proposed framework has demonstrated significant improvements in resource utilization and load balancing in dynamic computing environments. The proposed method outperformed traditional methods such as ACO and Least Load Balancing (LLB) in key metrics, including total energy consumption, average energy consumption per node, and peak energy consumption. These results highlight the effectiveness of combining ACO's optimal pathfinding capabilities with RL's adaptive learning to manage resources efficiently and maintain lower energy demands.

Despite these promising outcomes, there is still room for improvement, particularly in dynamic energy consumption. Future research could focus on further enhancing the dynamic response capabilities of the framework to better handle fluctuating workloads. Additionally, exploring hybrid models that incorporate other optimization techniques alongside ACO and RL could yield even more robust solutions. Another potential avenue for future work is the application of this framework to other domains such as smart grids and IoT networks, where efficient resource management is equally critical. Finally, real-world testing and validation in diverse and large-scale environments will be crucial in refining the framework and ensuring its practical applicability and scalability. This ongoing research and development will contribute to more sustainable, efficient, and resilient computing infrastructures.

References

[1] D. Kanellopoulos and V. K. Sharma, "Dynamic Load Balancing Techniques in the IoT: A Review," *Symmetry*, vol. 14, no. 12. 2022, doi: 10.3390/sym14122554.

[2] M. Karimi-Mamaghan, M. Mohammadi, P. Meyer, A. M. Karimi-Mamaghan, and E. G. Talbi, "Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art," *Eur. J. Oper. Res.*, vol. 296, no. 2, pp. 393–422, 2022, doi: 10.1016/j.ejor.2021.04.032.

[3] K. A. Athira *et al.*, "ScienceDirect Conference on ACO-DTSP ACO-DTSP Algorithm : Algorithm : Optimizing Optimizing UAV UAV Swarm Swarm Routes Routes with with Workload Constraints Workload Constraints," *Procedia Comput. Sci.*, vol. 235, pp. 163–172, 2024, doi: 10.1016/j.procs.2024.04.019.

[4] L. Yin, C. Sun, M. Gao, Y. Fang, M. Li, and F. Zhou, "Hyper-Heuristic Task Scheduling Algorithm Based on Reinforcement Learning in Cloud Computing," *Intell. Autom. Soft Comput.*, vol. 37, no. 2, pp. 1587–1608, 2023, doi: 10.32604/iasc.2023.039380.

[5] V. Mehta and A. Gupta, "Machine Learning Solutions for Load Balancing in Cloud Resource Management: A Review," in *2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS)*, 2023, pp. 452–459, doi: 10.1109/ICAISS58487.2023.10250470.

[6] S. P. Singh, "Effective load balancing strategy using fuzzy golden eagle optimization in fog computing environment," *Sustain. Comput. Informatics Syst.*, vol. 35, p. 100766, 2022, doi: https://doi.org/10.1016/j.suscom.2022.100766.

[7] A. Daliri, M. Alimoradi, M. Zabihimayvan, and R. Sadeghi, "World Hyper-Heuristic: A novel reinforcement learning approach for dynamic exploration and exploitation," *Expert Syst. Appl.*, vol. 244, p. 122931, 2024, doi: https://doi.org/10.1016/j.eswa.2023.122931.

[8] A. Jermanshiyamala, N. K. S. Kumar, S. Belhe, K. Sreekanth, S. Ray, and S. Sengan, "ACO-Optimized DRL Model for Energy-Efficient Resource Allocation in High-Performance Computing BT - Multi-Strategy Learning Environment," 2024, pp. 143–154.

[9] H. B. Sahoo and D. Chandrasekhar Rao, "Optimal Resource Allocation in Cloud Computing Using Novel ACO-DE Algorithm BT - Artificial Intelligence: Theory and Applications," 2024, pp. 443–455.

[10] S. S. Tripathy *et al.*, “State-of-the-Art Load Balancing Algorithms for Mist-Fog-Cloud Assisted Paradigm: A Review and Future Directions,” *Arch. Comput. Methods Eng.*, vol. 30, no. 4, pp. 2725–2760, 2023, doi: 10.1007/s11831-023-09885-1.

[11] F. S. Prity and M. M. Hossain, “A comprehensive examination of load balancing algorithms in cloud environments: a systematic literature review, comparative analysis, taxonomy, open challenges, and future trends,” *Iran J. Comput. Sci.*, 2024, doi: 10.1007/s42044-024-00183-y.

[12] J. Kallestad, R. Hasibi, A. Hemmati, and K. Sörensen, “A general deep reinforcement learning hyperheuristic framework for solving combinatorial optimization problems,” *Eur. J. Oper. Res.*, vol. 309, no. 1, pp. 446–468, 2023, doi: 10.1016/j.ejor.2023.01.017.

[13] S. Balavignesh, C. Kumar, R. Sripriya, and T. Senjyu, “An enhanced coati optimization algorithm for optimizing energy management in smart grids for home appliances,” *Energy Reports*, vol. 11, no. February, pp. 3695–3720, 2024, doi: 10.1016/j.egyr.2024.03.031.

[14] R. M. Mahdi, H. J. Hassan, and G. M. Abdulsaheb, “A Review Load balancing algorithms in Fog Computing,” *BIO Web Conf.*, vol. 97, 2024, doi: 10.1051/bioconf/20249700036.

[15] M. I. Khaleel, M. Safran, S. Alfarhood, and D. Gupta, “Combinatorial metaheuristic methods to optimize the scheduling of scientific workflows in green DVFS-enabled edge-cloud computing,” *Alexandria Eng. J.*, vol. 86, no. December 2023, pp. 458–470, 2024, doi: 10.1016/j.aej.2023.11.074.

[16] P. Xu, G. He, Z. Li, and Z. Zhang, “An efficient load balancing algorithm for virtual machine allocation based on ant colony optimization,” *Int. J. Distrib. Sens. Networks*, vol. 14, no. 12, p. 1550147718793799, Dec. 2018, doi: 10.1177/1550147718793799.

[17] Z. Ye, G. Sun, and M. Guizani, “ILBPS: An Integrated Optimization Approach Based on Adaptive Load-Balancing and Heuristic Path Selection in SDN,” *IEEE Internet Things J.*, vol. 11, no. 4, pp. 6144–6157, 2024, doi: 10.1109/JIOT.2023.3309832.

[18] R. Geetha, V. Rajamani, and V. Parthasarathy, “A Dynamic Approach to Optimizing Cloud Resource Allocation for Enhanced E-commerce Performance,” in *2024 10th International Conference on Communication and Signal Processing (ICCSP)*, 2024, pp. 982–987, doi: 10.1109/ICCSP60870.2024.10543382.

[19] S. Khan, K. A. Awan, I. U. Din, A. Almogren, and B. Seo-Kim, “An Adaptive Biomimetic Ant Colony Optimization With 6G Integration for IoT Network Communication,” *IEEE Access*, vol. 11, pp. 95584–95599, 2023, doi: 10.1109/ACCESS.2023.3310273.

[20] M. S. Al Reshan *et al.*, “A Fast Converging and Globally Optimized Approach for Load Balancing in Cloud Computing,” *IEEE Access*, vol. 11, pp. 11390–11404, 2023, doi: 10.1109/ACCESS.2023.3241279.