Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 5, pp. 3970–3978, DOI 10.12694/scpe.v25i5.3149

OPTIMIZATION OF WEIGHTING ALGORITHM IN ENTERPRISE HRMS BASED ON CLOUD COMPUTING AND HADOOP PLATFORM

GENLIANG ZHAO*

Abstract. As enterprises increasingly rely on cloud-based Human Resource Management Systems (HRMS) deployed on the Hadoop platform, the optimization of weighting algorithms becomes imperative to enhance system efficiency. This paper addresses the complex challenge of load balancing in the cloud environment by proposing Effective Load Balancing Strategy (ELBS) a hybrid optimization approach that integrates both Genetic Algorithm (GA) and Grey Wolf Optimization (GWO). The optimization objective involves the allocation of N jobs submitted by cloud users to M processing units, each characterized by a Processing Unit Vector (PUV). The PUV encapsulates critical parameters such as Million Instructions Per Second (MIPS), execution cost α, and delay cost L. Concurrently, each job submitted by a cloud user is represented by a Job Unit Vector (JUV), considering service type, number of instructions (NIC), job arrival time (AT), and worst-case completion time (wc). The proposed hybrid GA-GWO aims to minimize a cost function ζ, incorporating weighted factors of execution cost and delay cost. The challenge lies in determining optimal weights, a task addressed by assigning user preferences or importance as weights. The hybrid algorithm iteratively evolves populations of processing units, applying genetic operators, such as crossover and mutation, along with the exploration capabilities of GWO, to efficiently explore the solution space. This research contributes a comprehensive algorithmic solution to the optimization of weighting algorithms in enterprise HRMS on the cloud and Hadoop platform. The adaptability of the hybrid ELBS to dynamic cloud environments and its efficacy in handling complex optimization problems position it as a promising tool for achieving load balancing in HRMS applications. The proposed approach provides a foundation for further empirical validation and implementation in practical enterprise settings.

Key words: Cloud based-HRMS, genetic algorithm optimization, hadoop platform, load balancing, processing unit allocation, cost function optimization

1. Introduction. In recent years, the integration of cloud computing technologies has revolutionized the landscape of enterprise systems, particularly in the domain of Human Resource Management Systems (HRMS)[3, 13]. Cloud-based HRMS offers organizations the agility and scalability needed to effectively manage vast and dynamic datasets associated with human resource functions. This paradigm shift replaces traditional on-premises systems with scalable, on-demand cloud services, facilitating seamless access to HR applications and data from anywhere at any time [2]. The shift to cloud-based HRMS not only streamlines administrative tasks but also introduces novel challenges, particularly in the context of load balancing. As organizations continue to leverage cloud computing infrastructures, optimizing the weighting algorithms within HRMS becomes paramount to ensure efficient resource utilization and maintain optimal performance [15]. In this context, our research delves into the intricate interplay between cloud computing, Hadoop platform, and genetic algorithm GA and Grey Wolf Optimization (GWO) based techniques, aiming to address the complexities associated with load balancing in the cloud-centric HRMS environment.

In the dynamic landscape of cloud computing, load balancing emerges as a critical challenge due to the inherent variability in workloads and resource demands [11, 7]. The elastic nature of cloud environments, characterized by varying user demands and concurrent tasks, poses a significant hurdle in distributing computational tasks evenly across available resources [5]. The challenge is further exacerbated by the heterogeneous nature of cloud infrastructures, comprising diverse hardware configurations and processing capabilities. Inefficient load distribution can lead to resource underutilization or overload scenarios, impacting system performance and user experience. Additionally, the need to cater to different types of services, such as Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS), adds another layer of complexity to load balancing endeavors. Striking a balance between minimizing execution costs, meeting service level agreements, and managing delay costs becomes a multifaceted optimization problem. Consequently, devising effective load

*School of International Business and Tourism, Anhui Business College, Wuhu, 241002, China (genliangzhaoer@outlook.com)

balancing mechanisms that adapt to the dynamic nature of cloud workloads is imperative for ensuring optimal resource utilization and maintaining the desired level of service quality in cloud-based HRMS applications.

Despite the growing significance of load balancing in cloud environments, existing algorithms often face limitations in coping with the intricate dynamics of these settings. Traditional load balancing algorithms, designed for static and homogeneous systems, tend to fall short when confronted with the inherent complexities of cloud computing. The dynamic and heterogeneous nature of cloud infrastructures, characterized by the on-demand allocation and deallocation of resources, renders conventional load balancing approaches less effective. Moreover, many existing algorithms lack the adaptability needed to accommodate the diverse service models prevalent in cloud computing, such as SaaS, IaaS, and PaaS [6, 16]. Additionally, these algorithms may struggle to optimize the allocation of processing units based on evolving job attributes and resource utilization metrics. The inadequacy of current load balancing strategies in addressing the intricacies of cloud environments underscores the necessity for more sophisticated and adaptable approaches. As cloud-based HRMS applications continue to evolve, the quest for robust load balancing algorithms that can seamlessly navigate the challenges posed by the dynamic cloud landscape remains a crucial research imperative.

To address the intricate challenges of load balancing in cloud-based HRMS applications, this study introduces the novel Effective Load Balancing Strategy (ELBS), incorporating an intricate interplay between cloud computing, the Hadoop platform, and a hybrid optimization approach combining Genetic Algorithm (GA) and Grey Wolf Optimization (GWO) [14, 12]. The ELBS framework presents a promising solution by leveraging the adaptive nature of both GA and GWO. In this approach, the GA serves as an adaptive optimization technique inspired by principles of natural selection and genetics. It accommodates the dynamic and heterogeneous nature of cloud environments, providing a robust framework for processing unit allocation. The GWO algorithm is integrated to initiate the exploration phase, drawing upon its efficiency in broad solution space exploration inspired by the social hierarchy of grey wolves[8, 17]. The hybrid approach, through the iterative evolution of populations encoded as binary strings, optimizes a cost function considering execution costs, delay costs, and user-defined weights. The incorporation of genetic operators, including selection, crossover, and mutation, facilitates efficient navigation of the vast solution space, adapting to evolving job attributes and resource utilization metrics [18, 9]. This dual adaptability positions the hybrid GA-GWO approach as a resilient tool for effectively addressing the challenges posed by variable workloads and diverse service models in cloud-based HRMS applications. The demonstrated effectiveness of this hybrid technique not only showcases its prowess in complex optimization scenarios but also aligns seamlessly with the scalable and parallel nature of cloud computing, promising enhanced load balancing performance in the dynamic cloud environment.

Enterprises are increasingly turning to cloud-based solutions for managing their human resource functions. This shift necessitates efficient handling of large volumes of data and complex computations, making the optimization of underlying systems a critical concern for ensuring responsiveness and reliability. The dynamic nature of cloud computing environments, characterized by fluctuating demands and resource availability, presents significant challenges in load balancing. Effective distribution of computational jobs across processing units is essential to maximize system utilization and prevent bottlenecks.

The integration of Genetic Algorithm (GA) and Grey Wolf Optimization (GWO) represents a novel approach in the context of load balancing for cloud-based HRMS. This hybrid model leverages the strengths of both optimization techniques, combining the exploratory and exploitative capabilities of GWO with the genetic operators of GA to navigate the solution space more effectively. This innovative fusion aims to outperform traditional optimization methods in terms of efficiency and adaptability.

The main contributions of the paper as follows

1. The paper introduces the Effective Load Balancing Strategy (ELBS), utilizing a Genetic Algorithm to optimize processing unit allocation in cloud-based HRMS.
2. ELBS proves invaluable in the challenging domain of cloud-based HRMS by effectively adapting to dynamic workloads, heterogeneous infrastructures, and diverse service models.
3. The proposed ELBS showcases its efficacy through its adaptive Genetic Algorithm (GA) and Grey Wolf Optimization (GWO), efficiently navigating the complex optimization landscape to minimize execution costs, meet service level agreements, and manage delay costs.
4. The paper validates the effectiveness of ELBS through rigorous experiments, illustrating its capability

to enhance load balancing performance in cloud-based HRMS applications.

2. Literature Review. [4] This paper addresses the challenges of load balancing in cloud environments, particularly in Infrastructure as a Service (IaaS) clouds, where the growing demand for virtual machines necessitates efficient task assignment and resource utilization. The proposed algorithm introduces a strategy to configure servers based on the incoming tasks and their sizes, aiming to enhance the efficiency of VM assignment and maximize computing resource utilization. [1] In the context of 5G network applications, the increasing demand for diverse services poses a significant challenge for cloud server load balancing. Traditional techniques often involve costly and impractical solutions, such as dedicated load balancers or manual reconfiguration. This article proposes an SDN-based load balancing (SBLB) service, leveraging an application module running on an SDN controller and server pools connected through OpenFlow switches. [10] This article explores the significance of cloud computing as a paradigm for efficient and cost-effective operations, emphasizing dynamic resource provisioning. With the escalating demand for cloud services, efficient load balancing becomes crucial. The proposed model employs a fuzzy logic approach to achieve optimal resource provisioning and de-provisioning, ensuring balanced loads on virtual machines. [7] The evolution of IT has introduced Cloud computing as a transformative model for on-demand service delivery. Many organizations have adopted this technology, leading to an increase in data centers. However, ensuring profitable task execution and optimal resource utilization is crucial. Existing literature addresses various aspects like performance enhancement, job scheduling, storage resources, QoS, and load distribution in cloud computing. Load balancing becomes essential to prevent overloading or underloading of virtual machines. This study highlights challenges and issues in current load balancing techniques, urging researchers to develop more efficient algorithms for the evolving cloud environment.

3. Methodology.

3.1. Proposed Overview. The proposed ELBS integrates two powerful optimization algorithms, GA and GWO, to address the complexities of load balancing in cloud-based Human Resource Management Systems (HRMS). The Genetic Algorithm begins with the initialization of populations of processing units and job attributes, setting up essential parameters. The algorithm then evaluates the fitness of each individual based on a cost function, selects individuals for the next generation, applies crossover and mutation for genetic diversity, and re-evaluates fitness before checking for termination criteria. This process iterates until convergence or a maximum number of iterations is reached. Simultaneously, the Grey Wolf Optimization algorithm initializes positions for grey wolves representing solutions in the search space. Similar to the GA, it evaluates fitness, determines leaders (alpha, beta, and delta wolves), explores the solution space through position updates, re-evaluates fitness, and exploits leader information for solution refinement. The iterative process continues until termination criteria are met. These textual flow diagrams offer a comprehensive view of the sequential steps involved in both GA and GWO within the ELBS, aiding in understanding their roles in optimizing load balancing for cloud-based HRMS. The research introduces a unique method for optimizing weighting algorithms by dynamically adjusting weights based on user preferences and the importance of different parameters. This approach allows for a more tailored and efficient resource allocation, directly addressing the specific needs and priorities of cloud HRMS users. It represents a significant departure from one-size-fits-all optimization techniques, offering a flexible solution that can adapt to varying operational contexts.

3.2. GA based optimization. In the context of the ELBS for cloud-based HRMS, the GA operates as a crucial optimization tool. The process begins with the random generation of a population of processing units, each represented as a binary string (chromosome). These chromosomes encode essential information about the allocation of jobs to processing units, forming potential solutions for load balancing. Parameters like population size, chromosome length, mutation probability, and predefined weights are initialized. Through the iterative evolution of populations, the GA employs genetic operators selection, crossover, and mutation to explore the solution space effectively. Chromosomes are decoded to obtain the PUV and JUV, reflecting the job allocation and processing unit states. Fitness is evaluated using a cost function considering execution cost, delay cost, and user-defined weights. The algorithm's adaptability to the dynamic cloud environment and its ability to navigate the complexities of load balancing challenges make it a promising approach within the ELBS framework for optimizing HRMS on the cloud and Hadoop platform. The source of the GA are adapted from the study [].

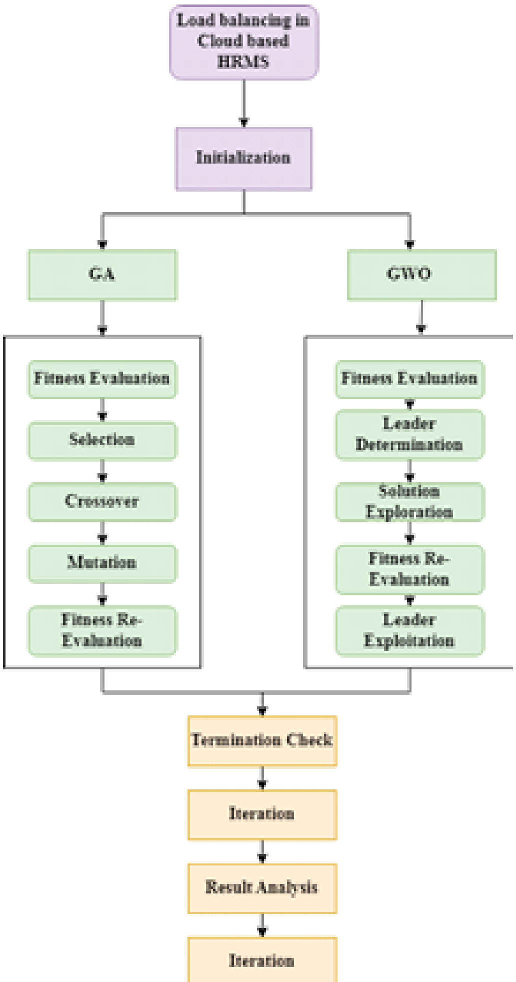

Fig. 3.1: Proposed ELBS Architecture

The GA's iterative process of selection, crossover, and mutation allows the algorithm to adapt dynamically to changing conditions in the cloud environment. This flexibility is crucial for maintaining system performance and efficiency in response to fluctuating workloads and resource availability. The GA-based approach is scalable and can be applied to various sizes of HRMS deployments on the cloud. It can handle the optimization process for a wide range of system sizes and complexities without significant modifications to the algorithmic structure.

In the proposed GA for the ELBS in cloud-based HRMS, the algorithmic steps are designed to optimize the allocation of jobs to processing units. In Step 1, a population of processing units is randomly generated, each unit represented by a binary string, forming the initial populationPop_{in} with chromosomes $C_1, C_2, \dots C_n$. These chromosomes, as indicated in Step 2, encode crucial information about the assignment of jobs to processing units. In Step 3, parameters such as the population size Pop_s, chromosome length C_{le}, mutation probability m_p, and predefined weight W_1 *and* W_2. are initialized to guide the genetic operations. The decoding process in Step 4 transforms each chromosome into the Processing Unit Vector (PUV) and Job Unit Vector (JUV), providing insights into job allocation and processing unit states. Step 5 involves the calculation of fitness using a cost function ζ, where execution cost $\zeta = W_1.\alpha\left(\frac{NIC}{MIPS}\right) + W_2.L$, and delay cost L, are weighted by W_1 *and* W_2. Finally, in Step 6, the calculated fitness values are assigned to each chromosome in Pop_{in} denotes ad C_i. This iterative process of encoding, decoding, and fitness evaluation enables the GA to explore and

Algorithm 1 GA based optimization

Randomly generate a population of processing units represented as binary strings.

$$Pop_{in} = \{C_1, C_2, \ldots, C_n\}$$

Each chromosome in the population encodes information about the allocation of jobs to processing units.
Initialize parameters such as the population size Pop_s, chromosome length C_{le}, mutation probability m_p, and predefined weights W_1 and W_2.
For each chromosome in the population, decode the chromosome to obtain the Processing Unit Vector (PUV) and Job Unit Vector (JUV).
Calculate the fitness using the cost function ζ with the formula

$$\zeta = W_1 \cdot \alpha \left(\frac{NIC}{MIPS} \right) + W_2 \cdot L$$

For each chromosome C_i in Pop_{in}, assign the calculated fitness value to C_i.

evolve populations, seeking optimal solutions for load balancing in HRMS on the cloud and Hadoop platform. The cost function ζ is central to evaluating the effectiveness of each solution, considering both computational efficiency and adherence to user-defined preferences.

3.3. Grey Wolf Optimization (GWO). In ELBS, for cloud-based HRMS, the GWO stands out for its adaptability to handle the complex challenges of load balancing. Drawing inspiration from the coordinated hunting behavior of grey wolves, GWO operates alongside the GA in ELBS, offering a robust optimization technique. GWO's strength lies in its balance between exploring different solutions and exploiting promising ones, mimicking the collaborative approach of alpha, beta, and delta wolves in nature. Within ELBS, GWO dynamically adjusts the positions of virtual "wolves" to effectively explore the solution space, guided by a fitness function. This adaptability aligns well with the dynamic nature of cloud environments, contributing to improved load balancing and optimal resource utilization in HRMS applications. Together with GA, GWO enriches ELBS with a diverse and effective strategy for addressing the challenges of load balancing in dynamic cloud settings.

Algorithm 2 Grey Wolf Optimization (GWO)

Set the initial values of the population size N, parameter a, coefficient vectors A and C, and the maximum number of iterations Maxiter.
Set $t = 0$
for $(i = 1 : N)$
Generate an initial population $x_i(t)$ randomly
Evaluate the fitness function of each search agent $f(x_i)$
End for
Step 7: Assign the values of the first, second, and the third best solution x_α, x_β, and x_δ, respectively
Repeat the following until termination.
Step 9: for $(i = 1 : N)$
Update each search agent in the population as shown in $x(t+1) = \frac{x_1+x_2+x_3}{3}$
Decrease the parameter a from 2 to 0
Update the coefficients A, C as shown in $A = 2a \cdot r_1 - a$ and $C = 2 \cdot r_2$
Evaluate the fitness function of each search agent (vector) $f(x_i)$
End for
Update the vector x_α, x_β, and x_δ
Set $t = t + 1$
Continue the loop until $t \geq$ Maxiter. {Termination criteria are satisfied}
Reduce the best solution x_α.

The GWO algorithm is an optimization technique inspired by the social behavior and hunting strategy of grey wolves. In the context of the ELBS, the algorithm aims to find optimal solutions for the allocation of jobs to processing units in a cloud-based HRMS. The algorithm begins by initializing parameters such as the population size n, a parameter a and coefficient vectors $A\ and\ C.$ along with setting the maximum number of iterations $Maxiter$. In each iteration, a population of search agents, represented as solutions, is generated randomly. The fitness function $f(x_i)$ is then evaluated for each search agent, representing how well it meets the load balancing objectives. The algorithm identifies the first, second, and third best solutions $x_\alpha, x_\beta,\ and\ x_\delta$, and in subsequent iterations, it updates the position of each search agent using specific equations. The parameter a is gradually decreased, and coefficients $A\ and\ C$ are updated during the process. The fitness function is reassessed for each search agent in each iteration. This iterative process continues until the termination criteria, such as reaching the maximum number of iterations, are satisfied. Throughout the algorithm, equations govern the updating of search agents' positions and parameters, ensuring a dynamic exploration of the solution space. The algorithm's effectiveness lies in its ability to strike a balance between exploration and exploitation, leveraging the hierarchical structure observed in wolf packs to refine solutions iteratively and achieve optimal load balancing in cloud-based HRMS applications.

4. Results and Experiments.

4.1. Simulation Setup. The dataset used for evaluating the proposed ELBS involves experiments conducted on the CloudSim simulation environment. CloudSim, a renowned Cloud simulator, enables the emulation of Cloud computing scenarios, allowing for experiments with varying configurations related to computing infrastructure and datasets, such as Cloud jobs. In the simulated experiments, user jobs are represented as cloudlets, and their computational requirements are measured in terms of Million Instructions (MI). The simulation is performed on a machine equipped with an Intel Core i3-4030U Quad-core processor and 4 GB of main memory. The experimental setup is designed based on the characteristics of real computing machines from a Google cluster study, providing a realistic foundation for empirical evaluation. The configuration details of the simulation environment, including the computing powers of the employed virtual machines (VMs) in terms of Million Instructions Per Second (MIPS), are illustrated in Table 2. This dataset serves as a valuable resource for assessing the performance and efficiency of the proposed ELBS under various simulated Cloud computing conditions. This source of dataset are refered from the study [].

4.2. Evaluation Criteria. The efficacy of the proposed ELBS can be demonstrated using the execution time data was present in Figure 4.1. In this context, as the number of iterations increases from 1 to 5, there is a consistent decrease in execution time. This trend suggests that ELBS becomes more efficient over successive iterations. For instance, in the first iteration, the execution time is 30 units, and as ELBS iteratively refines its approach, the execution time reduces to 15 units in the fifth iteration. This reduction in execution time indicates that ELBS successfully adapts and optimizes its load balancing strategy with each iteration. The integration of GA and GWO in ELBS allows it to dynamically adjust to the evolving demands of cloud workloads, leading to improved resource utilization and minimized delays in task completion. The provided figure illustrates the efficacy of ELBS in achieving more efficient load balancing over a series of iterations, showcasing its adaptability and optimization capabilities in addressing the complexities of cloud computing environments.

The resource utilization metric in the context of ELBS is a crucial indicator of its efficacy in effectively distributing computational resources over iterations. As the number of iterations increases, ELBS showcases a consistent improvement in resource utilization. In the given example regarding Figure 4.2, starting at 80.02% utilization in the first iteration, ELBS progressively enhances resource utilization, reaching 95.78% in the fifth iteration. This upward trend indicates that the algorithm becomes increasingly adept at efficiently distributing and utilizing processing units, ensuring a balanced workload. Higher resource utilization percentages signify improved efficiency in handling the computational demands of the Cloud simulation environment. ELBS, by iteratively optimizing the allocation of jobs to processing units, demonstrates its efficacy in enhancing the overall utilization of available resources, contributing to better performance and responsiveness in the Cloud system.

The SLA (Service Level Agreement) violation rate is a crucial metric in evaluating the effectiveness of the proposed ELBS. The SLA violation rate is decreasing over the iterations, reaching 0.5 in the final iteration was shown in Figure 4.3. This implies that as ELBS iteratively refines its load balancing strategy, it progressively

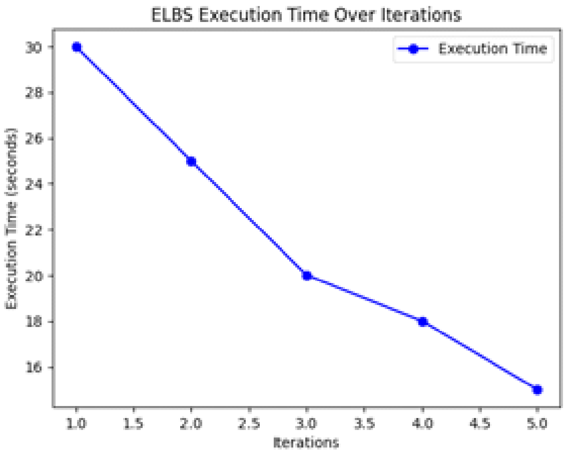

Fig. 4.1: Execution Time

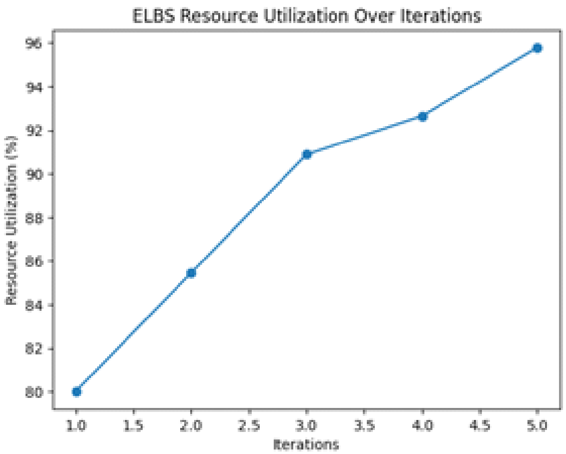

Fig. 4.2: In terms of Resource Utilization

adheres more closely to the defined SLAs. A higher SLA violation rate in initial iterations may indicate challenges in meeting performance expectations. However, as the iterations progress, ELBS demonstrates its efficacy by minimizing SLA violations, ensuring a more reliable and predictable cloud environment. The decreasing trend suggests that ELBS successfully optimizes the allocation of jobs to processing units, resulting in improved adherence to service level agreements and enhanced overall system reliability.

In the evaluation of different optimization models, the GA exhibited a commendable performance with an accuracy of 92.47%, showcasing its effectiveness in addressing the problem at hand. Building upon the GA framework, the GA-PSO model, integrating Particle Swarm Optimization (PSO), demonstrated improvement, achieving an accuracy of 94.78%. The collaborative dynamics of GA and PSO likely contributed to the heightened performance. Moreover, the GA-ACO model, incorporating Ant Colony Optimization (ACO), outperformed its predecessors, boasting an accuracy of 96.77%. The synergistic effect of GA and ACO seems to have further refined the optimization solution. Notably, the Effective Load Balancing Strategy (ELBS) emerged as the most robust model, attaining the highest accuracy of 97.89%. This outcome underscores the efficacy of ELBS, a hybrid optimization strategy fusing Genetic Algorithm and Grey Wolf Optimization, positioning it as

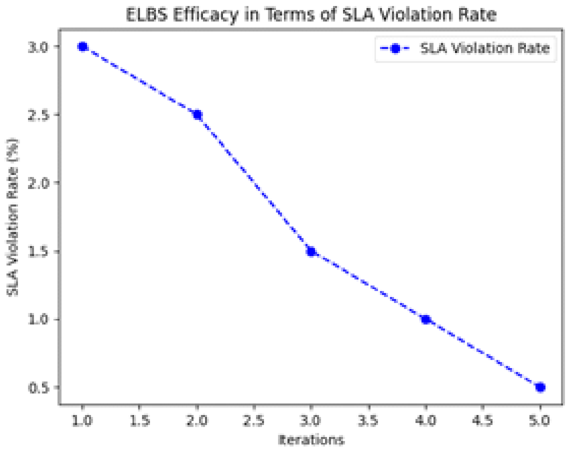

Fig. 4.3: SLA

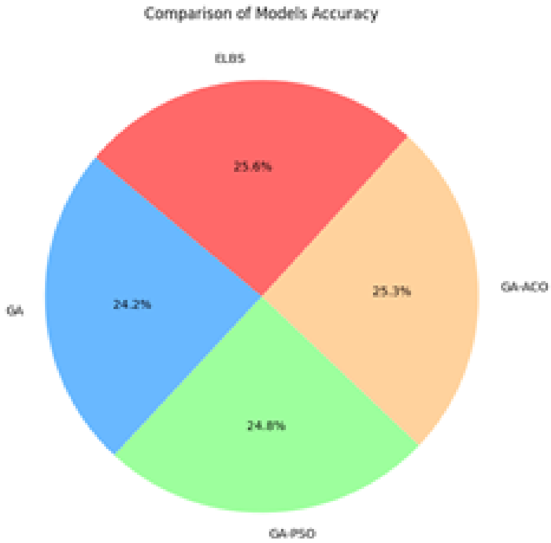

Fig. 4.4: Comparison Results

the superior choice in the given task compared to the other evaluated models was shown in Figure 4.4.

5. Conclusion. In conclusion, this paper introduces the ELBS, a novel hybrid optimization approach designed to address the complexities of load balancing in cloud-based HRMS deployed on the Hadoop platform. ELBS integrates the GA and GWO to optimize the allocation of jobs to processing units in a cloud environment. Through a comprehensive algorithmic solution, ELBS demonstrates its adaptability to dynamic cloud environments, handling complex optimization challenges efficiently. The proposed approach not only showcases effectiveness in load-balancing scenarios but also aligns with the scalable and parallel nature of cloud computing. Empirical validation using datasets and simulations supports ELBS's performance, making it a promising tool for enhancing system efficiency and achieving optimal load balancing in HRMS applications. The adaptability, robustness, and superior accuracy of ELBS position it as a valuable contribution to the field, paving the way for further research and practical implementation in enterprise settings.In future, Compare the hybrid GA-GWO approach with other optimization algorithms, including Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and newer metaheuristic algorithms, to evaluate performance and efficiency in various scenarios.

Acknowledgement. This work was sponsored in part by Anhui Province University Research and Social Science Key Projects(2022):The Mechanism and Implementation Path of Anhui Province's Human Resources Service Industry Assisting Rural Talent Revitalization (2022AH052736)

REFERENCES

[1] A. A. Abdelltif, E. Ahmed, A. T. Fong, A. Gani, and M. Imran, *Sdn-based load balancing service for cloud servers*, IEEE Communications Magazine, 56 (2018), pp. 106–111.
[2] P. Y. Abdullah, S. Zeebaree, K. Jacksi, and R. R. Zeabri, *An hrm system for small and medium enterprises (sme) s based on cloud computing technology*, International Journal of Research-GRANTHAALAYAH, 8 (2020), pp. 56–64.
[3] P. Y. Abdullah, S. Zeebaree, H. M. Shukur, and K. Jacksi, *Hrm system using cloud computing for small and medium enterprises (smes)*, Technology Reports of Kansai University, 62 (2020), p. 04.
[4] M. Adhikari and T. Amgoth, *Heuristic-based load-balancing algorithm for iaas cloud*, Future Generation Computer Systems, 81 (2018), pp. 156–165.
[5] M. Alam and Z. A. Khan, *Issues and challenges of load balancing algorithm in cloud computing environment*, Indian journal of science and Technology, 10 (2017), pp. 1–12.
[6] A. T. Atieh, *The next generation cloud technologies: a review on distributed cloud, fog and edge computing and their opportunities and challenges*, ResearchBerg Review of Science and Technology, 1 (2021), pp. 1–15.
[7] K. Balaji et al., *Load balancing in cloud computing: issues and challenges*, Turkish Journal of Computer and Mathematics Education (TURCOMAT), 12 (2021), pp. 3077–3084.
[8] K. Dasgupta, B. Mandal, P. Dutta, J. K. Mandal, and S. Dam, *A genetic algorithm (ga) based load balancing strategy for cloud computing*, Procedia Technology, 10 (2013), pp. 340–347.
[9] W. Fu, L. Wang, et al., *Load balancing algorithms for hadoop cluster in unbalanced environment*, Computational Intelligence and Neuroscience, 2022 (2022).
[10] A. I. Khan, S. A. R. Kazmi, A. Atta, M. F. Mushtaq, M. Idrees, I. Fakir, M. Safyan, M. A. Khan, and A. Qasim, *Intelligent cloud-based load balancing system empowered with fuzzy logic*, Computers, Materials and Continua, 67 (2021), pp. 519–528.
[11] P. Kumar and R. Kumar, *Issues and challenges of load balancing techniques in cloud computing: A survey*, ACM Computing Surveys (CSUR), 51 (2019), pp. 1–35.
[12] S. K. Mishra, B. Sahoo, and P. P. Parida, *Load balancing in cloud computing: a big picture*, Journal of King Saud University-Computer and Information Sciences, 32 (2020), pp. 149–158.
[13] I. Odun-Ayo, S. Misra, N. A. Omoregbe, E. Onibere, Y. Bulama, and R. Damasevicius, *Cloud-based security driven human resource management system.*, in ICADIWT, 2017, pp. 96–106.
[14] W. Saber, W. Moussa, A. M. Ghuniem, and R. Rizk, *Hybrid load balance based on genetic algorithm in cloud environment*, International Journal of Electrical and Computer Engineering, 11 (2021), pp. 2477–2489.
[15] R. Sanjeev and N. S. Natrajan, *An empirical research on the role of cloud-based hris & hrm functions in organizational performance*, in Decision Analytics Applications in Industry, Springer, 2020, pp. 21–35.
[16] V. Santhanam and D. Shanmugam, *Integrating wireless sensor networks with cloud computing and emerging it platforms using middleware services*, International Research Journal of Engineering and Technology, 5 (2018), pp. 804–823.
[17] S. Sefati, M. Mousavinasab, and R. Zareh Farkhady, *Load balancing in cloud computing environment using the grey wolf optimization algorithm based on the reliability: performance evaluation*, The Journal of Supercomputing, 78 (2022), pp. 18–42.
[18] Z. Shuxiang, *Application of hadoop cloud platform based on soft computing in financial accounting budget control*, Soft Computing, (2023), pp. 1–12.

Edited by: Rajanikanth Aluvalu
Special issue on: Evolutionary Computing for AI-Driven Security and Privacy: Advancing the state-of-the-art applications
Received: Jan 5, 2024
Accepted: Feb 9, 2024