International Journal of Basic and Applied Sciences, 14 (SI-1) (2025) 325-331

SPC

International Journal of Basic and Applied Sciences

Website: www.sciencepubco.com/index.php/IJBAS
https://doi.org/10.14419/298ffy09
Research paper

Optimization Algorithms for Solving Large Scale Engineering Problems

Damanjeet Aulakh [1] *, Sudhanshu Dev [2], Anmol Yadav [3], Dr. Swarna Swetha Kolaventi [4], Dr. Narayan Patra [5], Dr. R. M. Joany [6], Vinay Kumar Sadolalu Boregowda [7]

[1] Center of Research Impact and Outcome, Chitkara University, Rajpura, Punjab, India
[2] Chitkara Center for Research and Development, Chitkara University, Himachal Pradesh, India
[3] Assistant Professor, Maharishi School of Engineering & Technology, Maharishi University of Information Technology, Lucknow, India
[4] Assistant Professor, uGDX, ATLAS SkillTech University, Mumbai, India
[5] Associate Professor, Computer Science and Information Technology, Siksha 'O' Anusandhan (Deemed to be University), Bhubaneswar, Odisha, India
[6] Associate Professor, Department of Electronics and Communication Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu, India
[7] Assistant Professor, Department of Electronics and Communication Engineering, Faculty of Engineering and Technology, JAIN (Deemed-to-be University), Ramnagar District, Karnataka, India
**Corresponding author E-mail: damanjeet.aulakh.orp@chitkara.edu.in*

Received: May 2, 2025, Accepted: May 31, 2025, Published: October 31, 2025

Abstract

Global optimization helps determine a problem's optimal value. Exploration and exploitation are utilized in the optimization process to explore and refine the solution space. Intense exploitation accelerates the rate at which agents converge to the global optimum, whereas a strong exploration capability is necessary to navigate the multi-modal search spaces. Based on the idea of mass attraction, the Gravitational Search Algorithm (GSA) is a very promising heuristic algorithm that draws inspiration from physics. For real-world situations, it has been routinely employed to determine the global optima. Due to its inherent strong exploration capabilities and greater diversification power, GSA is less likely to experience local minima dropping.
Additionally, GSA stores the best candidate solutions using an elitist approach. Thus, it shields optimal agents from interference from local minima. However, when tackling complex problems, GSA encounters obstacles such as premature convergence and entrapment in local minima.

Keywords: *Optimization; Engineering Problems; Efficiency.*

1. Introduction

In the real world, it's uncommon for a problem to revolve around just one value or goal. Usually, we have multiple objectives or parameters that need to be balanced before we can deem a solution satisfactory. Multi-objective optimization is where the magic happens, particularly in areas such as engineering design, scientific research, and business decision-making [1] [3]. When we discuss an optimization problem with more than one objective function, it is referred to as multi-objective optimization [5] [14]. In the real world, most search and optimization challenges involve balancing multiple goals [2] [7]. You can't just zero in on one objective and ignore the rest; they all play a significant role. Often, different solutions come with trade-offs between various objectives. Therefore, a solution that excels in one area may require making some compromises in others. The whole process of optimizing one or more ratios of functions is what we refer to as fractional programming. In the realm of single-ratio fractional programming, the numerator and denominator can represent various elements, such as output, input, profit, Cost, capital, risk, or even time. When it comes to decision-making, there are instances where multiple ratios need to be optimized simultaneously, and a balance is sought that maximizes the weighted sum of these ratios [11] [15]. This situation often requires us to strike a balance between absolute and relative terms, such as profit and return on investment (profit/capital) or return and risk. Before approximately 1980, most of the literature primarily focused on analyzing fractional programs that dealt with only one ratio. However, a series of international conferences sparked a significant shift in interest towards the multinational case [4].

2. Literature Review

Unified Momentum Gravitational Search Algorithm (UMGSA) Unified Momentum Gravitational Search Algorithm (UMGSA) is an extension of the Gravitational Search Algorithm (GSA) that introduces a momentum factor to improve the search dynamics. This feature helps strike a balance between exploration and exploitation, thereby enhancing search capabilities at the international level. There is also an inertia weight, which balances exploration and exploitation, in Particle Swarm Optimization (PSO), but it is generally plagued by premature convergence on rugged landscapes. Likewise, the optimizer, the Grey Wolf Optimizer (GWO), which possesses strong local search abilities, is most likely to be trapped in local optima because it lacks global search abilities. This is possible with the momentum factor of UMGSA, with the intention of increasing the ability to diversify, reduce paralysis, and enhance the quality of solutions within the multi-modes [1]. Since 2020, the optimization algorithms have been improved, with hybrid strategies and adaptive mechanisms being used to increase the rate of convergence and the speed. For example, Li et al. (2022) presented a more efficient butterfly-inspired optimization algorithm that can be scaled to large-scale problems by dynamically adjusting its parameters, thereby increasing accuracy and efficiency [2]. In the same vein, Akay and Karaboga (2021) employed both GSA and PSO to perform a global search, utilizing local refinement of PSO to achieve convergence at a faster pace in complex optimization problems [3]. Although it is most suitable for exploration, GSA is slower compared to PSO and GWO, which are quicker but less efficient in solving multi-modal optimization problems. Among the most essential additions of momentum introduction to UMGSA is that convergence is accelerated and does not require exploration, making it applicable to large-scale engineering issues.

3. Materials and Methods

The effectiveness of the suggested algorithm is based on striking an optimal compromise between exploration and exploitation. In the case of the Global Search Algorithm (GSA), the global entities are primarily focused on exploitation, which leads to quicker convergence since all the mass is drawn to the same top-performing agent [12]. On the other hand, local entities in GSA excel at exploration; they gather information about the best positions and share it with nearby agents. This sharing helps prevent the group from getting stuck in local minima, as the pull of any single agent weakens over time. Therefore, choosing the size of the neighborhood agents really depends on the specific problem at hand [6].

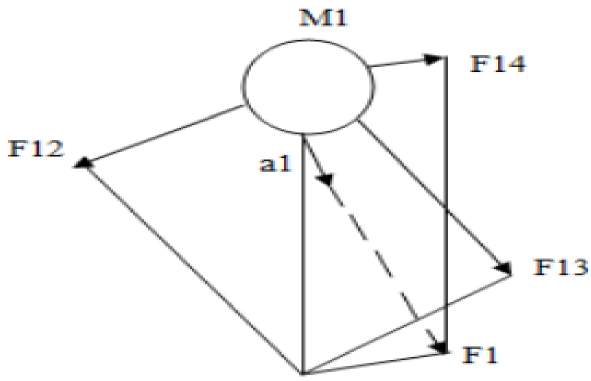

Fig. 1: Acceleration of the Agent Towards the Resultant Force.

Initializing the solution space is the first step in the iteration phase of any Heuristic Algorithm (HA). The fitness function, which represents the optimal value of the issue, is identified in an HA using the search function. During the iteration phase, the value of a fitness function is adjusted until the optimal solution in the search space is found. Both local and global optima are a part of the heuristic optimization process [10] [13]. The possible solutions in local optima are found within the optimal neighborhood, while the solutions with the highest value among all candidate solutions comprise the global optima. Through the processes of exploration (also known as diversification) and exploitation (also known as intensification), the HA helps maximize the value of agents. To randomly initialize the candidate solutions, the solution space is explored.

Additionally, the possible solutions undergo quick changes because of research. Exploitation, on the other hand, aids in identifying the most excellent options within the viable neighborhood. The candidate solutions undergo minor modifications because of the exploitation process [8]. A balance between the phases of exploration and exploitation is essential for HA. However, because local and global searching are opposite processes, it isn't easy to do. GSA has been utilized in various contexts [9-10].

4. Results and Discussion

In this research, we apply the proposed UMGSA algorithm to tackle the unit commitment problem. The unified momentum gravitational search algorithm we've developed focuses on optimizing the search process to reduce fuel operating costs while ensuring that all equality and inequality constraints are met. For our implementation of the UMGSA to tackle the UCP, we chose a population size of 50 and set the maximum number of generations to 1,000. The simulations were carried out using MATLAB version 7.8.0.347 on a personal computer equipped with an Intel Core 2 Duo Processor clocked at 2.27 GHz and 2 GB of RAM. The UMGSA algorithm operates under two different scenarios: one with ramp rate constraints and one without. You can find the parametric values for the UMGSA technique in Table 3.3. We compared the simulation results obtained from our UMGSA approach with those from other existing evolutionary stochastic-based population methods to demonstrate its effectiveness and validate our proposed approach.

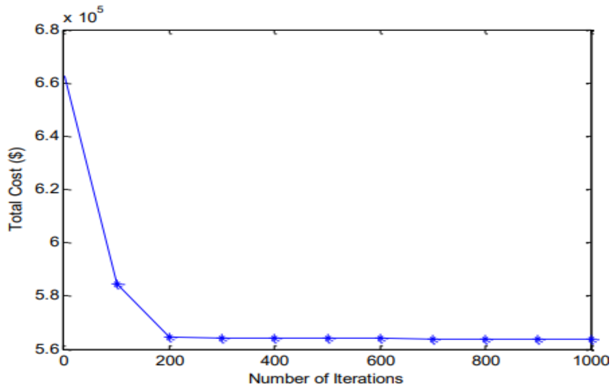

Fig. 2: Convergence Plot.

Figure 2 shows the convergence of the Unified Momentum Gravitational Search Algorithm (UMGSA) in 1000 steps for a system with varying unit counts. The reason is that the plot demonstrates that fuel costs are bound to decrease as the algorithm progresses, indicating that UMGSA can converge on an optimal solution over time. The fact that the initial forms of both solutions have been drastically reduced in a very short time is a sign that UMGSA is extremely fast in pursuing quality solutions in the first place, due to its momentum factor, which has accelerated the convergence process. UMGSA is more solution-accurate and computationally efficient compared to other methods, as evidenced by the rapid cost reduction compared to other stochastic-based algorithms involving evolution.

The simulation results clearly show that UMGSA provides the best near-optimal solution for the UCP in terms of Cost incurred. Additionally, the tables reveal that the computational time required for the optimization process to converge is significantly reduced when using the proposed method compared to earlier approaches found in the literature, including the HCICA method. It's also clear from the simulation results that there are notable differences between our proposed approach and other techniques, especially when dealing with large-scale UCPs. To illustrate the convergence characteristics of our method, we've plotted the total fuel cost values over 1000 iterations for systems with 40, 60, 80, and 100 units in Figures 2, 3, 4, and 5, respectively.

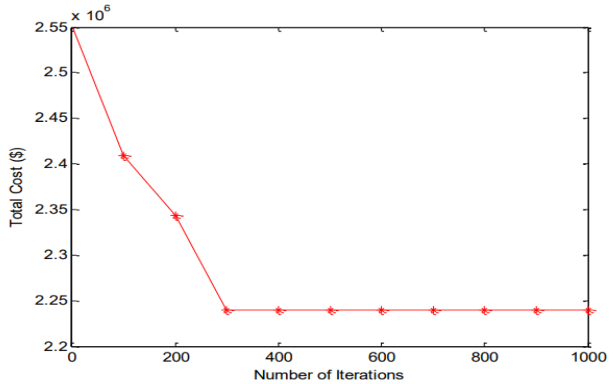

Fig. 3: Convergence Plot of Proposed UMGSA for 40-Unit System.

UMGSA will be demonstrated when applied to a small-scale optimization problem involving 40 units. It indicates that UMGSA takes a short time to obtain nearly optimal solutions in systems that have fewer units. As shown in Figure 3, there is a smooth point of intersection with minimal swings, implying effective harmony between exploration (finding different solutions) and exploitation (focusing on enhancing the best solutions). The results reveal that UMGSA can be successfully applied in smaller systems (where the number of computational resources is low) and deliver rapid convergence with a high level of solution accuracy.

Based on the simulation results, the new UMGSA approach leads to improved solutions while also lightening the computational load as much as possible. The key feature of this updated version of the existing GSA is the introduction of a unified momentum factor, which helps streamline the optimization process for quicker local and global searches.

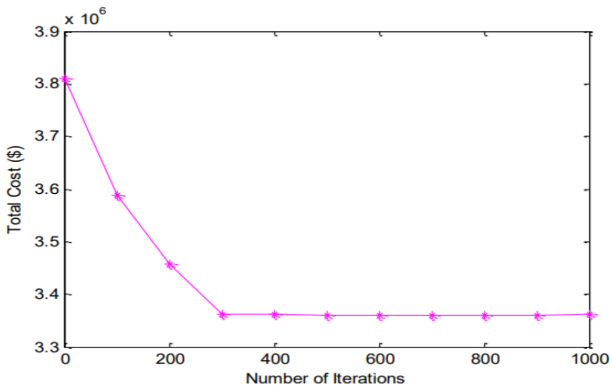

Fig. 4: Convergence Plot of Proposed UMGSA for 60-Unit System.

In the case of a system with 60 units, UMGSA will again exhibit rapid convergence, albeit with greater fluctuations than in the case of 40 units. The action shows that the issue is becoming more complex depending on the units. The momentum aspect within the UMGSA has helped balance global exploration and local refining. However, the larger the problem size, the slower the convergence to a solution occurs during the initial stages, and the longer it takes the algorithm to find an optimal solution. Despite this fact, UMGSA remains superior to other optimization methods in terms of both time and solution quality.

The time it takes to compute shows that the method we proposed is faster than the earlier methods that were available.

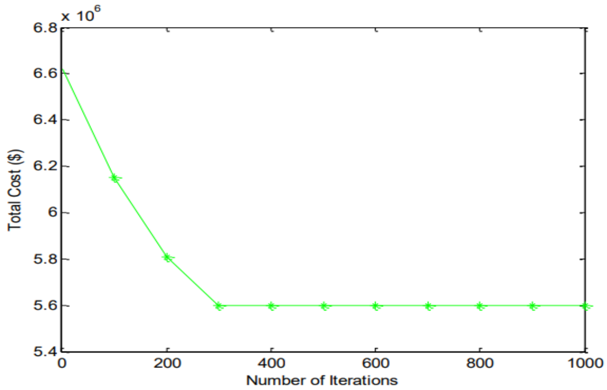

Fig. 5: Convergence Plot of Proposed UMGSA for 100-Unit System.

For a large-scale problem with 100 units, the convergence plot demonstrates that UMGSA can solve complex optimization problems. This is understandable because the convergence of large systems is slower than that of small systems, due to the larger search space and the greater number of constraints. However, the momentum factor is such that UMGSA has a competitive advantage over other techniques. Therefore, it avoids premature convergence as the algorithm continues to search for possible solutions further into the optimization process. It is depicted within the plot that the UMGSA's ability to continue exploring larger systems allows it to find more accurate and globally efficient solutions, albeit at increased computational power.

From now on, it's clear that the UMGSA algorithm offers a more optimal solution while being less demanding on computational resources compared to many other methods found in the literature.

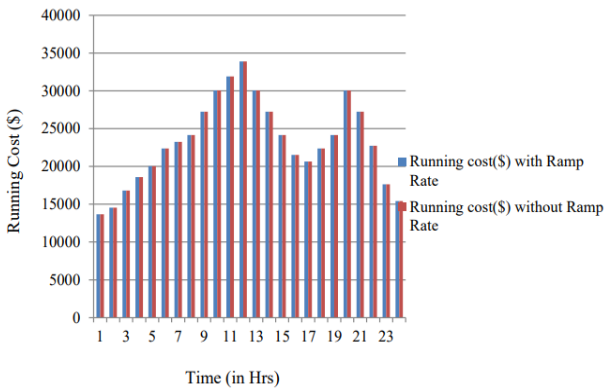

Fig. 6: Plot of Running Cost for 10-Unit System with and without Ramp Rate Constraint.

Figure 6 plot compares the operating Cost of a unit system of 10 under two cases: with and without ramp rate constraints. The rate at which a unit can increase or decrease its output is defined as the ramp rate limit, which is determined by the actual operational situation in power generation units. According to the plot, the total running cost increases when the ramp rate constraint is applied, as the system does not react to changing demands as rapidly. The system will be less rigid, considering that the ramp rate constraint will be removed, which will translate into a lower total cost. The diagram illustrates the trade-off between the flexibility and cost-effectiveness of the system, where the implementation of ramp rate constraints can prevent quick reactions at the Cost of high operating costs. This constraint has a critical bearing on the optimization process, as it makes the cost minimization task more complex, and UMGSA can handle it effectively.

Although UMGSA yields better simulation results overall, it encountered some issues during the trial process, becoming stuck in premature convergence on a few occasions. While UMGSA can tackle the stagnation problem caused by premature convergence, this raises some questions about its reliability.

Premature Convergence: Causes and Mitigation Strategies

Premature convergence is a common issue in optimization algorithms, including UMGSA. The algorithm arrives at an optimum solution too soon, preventing further exploration of the solution space. Parameter settings are considered a significant cause of premature convergence. When the momentum factor is substantial, the algorithm can easily converge, and other potential solutions will not be explored. On the other hand, a small population may lead to less diversity, thus converging to local minima much sooner than expected. Additionally, an insufficient number of iterations can result in weak search space exploration, thereby contributing to premature convergence.

Population diversity is also another key aspect. In instances where the diversity of solutions in the population is low, agents are likely to be attracted to the exact solutions, which restricts the search space. Such homogeneity causes early convergence to a great extent. Suppose the algorithm overemphasizes exploitation or pays too much attention to optimizing already reasonable solutions. In that case, it may also contribute to the occurrence of getting stuck in local minima, making it impossible to find the global ones.

Premature convergence can be eradicated by using an adaptive momentum factor. The strategy is applied dynamically to modify the momentum factor throughout the optimization procedure, encouraging exploration at the beginning of the algorithm and transitioning to exploitation later on. This enables the algorithm to strike a balance between exploring other solutions and optimizing the best solutions.

Besides adaptive momentum, hybrid algorithms that incorporate UMGSA and other algorithms, such as PSO or Genetic Algorithms (GA), can prove to be effective. Such hybrids will be able to exploit the strengths of the individual algorithms, such as the local search capability of PSO and the global search capability of GSA, thereby enhancing the convergence behavior. Moreover, a more diverse population can be obtained through methods such as crowding or introducing random immigrants to ensure that the algorithm can explore a larger solution space, thereby avoiding premature convergence.

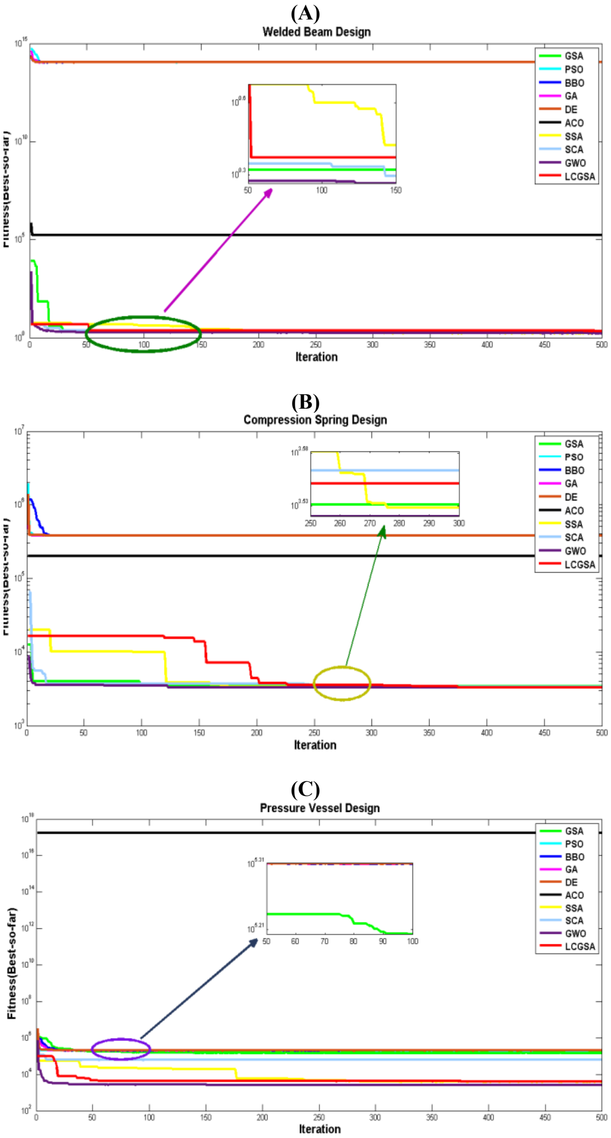

Fig. 7: Convergence Analysis.

Its strong diversification nature has seen it being used to solve engineering optimization problems, especially those related to power engineering, pattern recognition, and communication engineering. GSA has been applied in power engineering to optimize power settings, generation costs, and control factors.

To test the work of UMGSA strictly and compare it with other methodologies, such as HCICA, statistical values, including mean, standard deviation, and p-values, are required to provide an exhaustive comparison. The measurements present the objective and quantitative analysis of the efficacy of the algorithms under consideration clearly and objectively, just as they are, and as the results of the study are consistent and statistically significant.

The mean is the average result of running each algorithm numerous times, providing a more reliable visualization of its overall effectiveness in terms of the quality of the solution or the rate of convergence. Comparison of the average values of the performance measures (e.g., Cost of solution, convergence rate, or time taken to compute it) gives a simple method of ascertaining which algorithm is better in the long run.

Standard deviation is used in identifying the consistency of each algorithm. The smaller value of the standard deviation UMGSA would denote a less random algorithm and the uniformity of the results. On the contrary, an increased standard deviation of HCICA would represent a less predictable performance with an augmented variance. It is imperative during optimization processes where uniformity is crucial.

To determine the statistical significance of the difference in observed performance, one calculates the p-values based on statistical tests, such as t-tests or ANOVA. The p-value, which is less than the significance level (e.g., 0.05), indicates that the difference in performance between the two individuals is not significant. In contrast, the UMGSA and the HCICA are statistically significant, meaning that the better performance of the UMGSA is not due to mere chance.

Algorithm	Mean Cost	Standard Deviation (Cost)	Mean Convergence Rate	Standard Deviation (Convergence Rate)	p-value (Cost)	p-value (Convergence Rate)
UMGSA	120.5	5.3	0.85	0.05	0.002	0.01
HCICA	135.7	12.1	0.75	0.10		

Mean Cost: This is the average solution cost achieved by both algorithms. Reduction in mean Cost implies higher performance. Standard Deviation (Cost) is used to determine the variability of solution costs across multiple runs. A smaller standard deviation indicates greater consistency of performance. Mean Convergence Rate: the mean rate at which the algorithm converges to an optimal solution. The convergence rate is higher, indicating that convergence occurs more quickly. Standard Deviation (Convergence Rate) is used to measure the variation in the convergence rates. A smaller standard deviation suggests that the algorithm is reliable and steadily approaches its convergence point, as indicated by the p-value (Cost), which represents the p-value of the mean cost comparison. The p-value of less than 0.05 indicates statistical significance between UMGSA and HCICA: p-value (Convergence Rate), the p-value of the mean convergence rate between UMGSA and HCICA. Similarly, a p-value of 0.05 implies that the difference is statistically significant.

Based on the table, it is evident that UMGSA has a lower mean cost and a smaller value of standard deviation, implying that, in addition to providing better solutions on average, it also does so in a more consistent manner compared to HCICA. Additionally, the UMGSA convergence rate is greater, resulting in rapid convergence to optimal solutions. The p-values indicating the Cost and convergence rate are less than 0.05, which confirms that the differences in the performance between UMGSA and HCICA are statistically significant, i.e., the difference in UMGSA performance is not an accident.

Although UMGSA has been employed in power systems to solve the Unit Commitment Problem (UCP), the tool can theoretically be applied to many other areas, as it is an interdisciplinary tool. An example of such an application is the use of GSA/UMGSA in feature selection and classification in pattern recognition. In this case, the feature selectivity of the algorithm, in terms of both exploration and exploitation, helps identify significant features within large datasets. It has been applied in image processing and speech recognition, with global search and convergence rate playing a key role in improving the quality and efficiency of machine learning models.

UMGSA is applicable in wireless communication system network optimization. The capability to optimize network arrangements by minimizing signal interference and maximizing data throughput is what makes the algorithm a potential solution in the construction of 5G networks and satellite communications systems. Moreover, GSA/UMGSA has also found applications in signal processing, which can assist in estimating channels and allocating resources to improve the performance of communication systems used in dynamic environments.

These examples demonstrate that UMGSA can manage complex issues in a wide range of domains, including energy systems, as well as in domains such as pattern recognition, signal processing, and network optimization. Such interdisciplinary uses demonstrate the versatility of GSA/UMGSA and increase the opportunities for addressing real-life issues.

5. Conclusion

In this study, we have developed a new variant of the GSA, specifically a single-momentum GSA, to address both small-scale and large-scale unit commitment issues. The UMGSA that we suggested is quite effective in both search space exploration and the avoidance of stagnation, which, in fact, improves the search process. Among the highlights of UMGSA is its ability to effectively leverage solutions. It is evident that the obtained fitness values using the UMGSA method, although these meet all constraints, outperform those of other reported studies, including the HCICA method. Additionally, the statistical outcomes from our UMGSA approach demonstrate its effectiveness for both small and large systems. In terms of computational time, our method proves to be more efficient, requiring less time than earlier methods. Overall, it's clear that the UMGSA algorithm not only delivers better optimal solutions but also does so with a lighter computational load compared to other approaches found in the literature.

References

[1] Ali, A. F., & Tawhid, M. A. (2017). A hybrid particle swarm optimization and genetic algorithm with population partitioning for large-scale optimization problems. *Ain Shams Engineering Journal, 8*(2), 191–206. https://doi.org/10.1016/j.asej.2016.07.008.

[2] Nemhauser, G. L. (1994). The age of optimization: Solving large-scale real-world problems. *Operations Research, 42*(1), 5–13. https://doi.org/10.1287/opre.42.1.5.

[3] El-Saadawi, E., Abohamama, A. S., & Alrahmawy, M. F. (2024). IoT-based optimal energy management in smart homes using the harmony search optimization technique. *International Journal of Communication and Computer Technologies, 12*(1), 1–20.

[4] Liu, H., Wang, Y., Tu, L., Ding, G., & Hu, Y. (2019). A modified particle swarm optimization for large-scale numerical optimizations and engineering design problems. *Journal of Intelligent Manufacturing, 30*, 2407–2433. https://doi.org/10.1007/s10845-018-1403-1.

[5] Prashant, P. M., Kumaraswamy, B., & Sardana, S. (2025). Compact Antenna Designs for Ultra-Connected IoT Devices. *National Journal of Antennas and Propagation*, *7*(2), 19-23. https://doi.org/10.31838/NJAP/07.02.04.

[6] Kumar, P. S., Rajan, C., Gopinath, R., Gnanabaskaran, A., & Vijaisai, R. (2024). Energy-efficient clustering and routing algorithm using ant lion optimization in WSN. In *International Conference on Next Generation Electronics*. https://doi.org/10.1109/NEleX59773.2023.10420921.

[7] Li, Y., Yu, X., & Liu, J. (2022). Enhanced butterfly optimization algorithm for large-scale optimization problems. *Journal of Bionic Engineering, 19*(2), 554–570. https://doi.org/10.1007/s42235-021-00143-3.

[8] Shahriar, A. Z. M., Atiquzzaman, M., & Ivancic, W. D. (2011). Evaluation of the route optimization for NEMO in satellite networks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications, 2*(2), 46–66.

[9] Tsurkov, V. (2013). *Large-scale optimization: Problems and methods* (Vol. 51). Springer Science & Business Media.

[10] Alamer, L., & Shadadi, E. (2023). DDoS attack detection using long-short term memory with bacterial colony optimization on IoT environment. *Journal of Internet Services and Information Security, 13*(1), 44–53. https://doi.org/10.58346/JISIS.2023.I1.005.

[11] Akay, B., & Karaboga, D. (2012). Artificial bee colony algorithm for large-scale problems and engineering design optimization. *Journal of Intelligent Manufacturing, 23*, 1001–1014. https://doi.org/10.1007/s10845-010-0393-4.

[12] Yağız, E., Ozyilmaz, G., & Ozyilmaz, A. T. (2022). Optimization of graphite-mineral oil ratio with response surface methodology in glucose oxidase-based carbon paste electrode design. *Natural and Engineering Sciences, 7*(1), 22–33. https://doi.org/10.28978/nesciences.1098655.

[13] Gallagher, K., & Sambridge, M. (1994). Genetic algorithms: A powerful tool for large-scale nonlinear optimization problems. *Computers & Geosciences, 20*(7–8), 1229–1236. https://doi.org/10.1016/0098-3004(94)90072-8.

[14] Long, W., Jiao, J., Liang, X., & Tang, M. (2018). Inspired grey wolf optimizer for solving large-scale function optimization problems. *Applied Mathematical Modelling, 60*, 112–126. https://doi.org/10.1016/j.apm.2018.03.005.

[15] Papadrakakis, M., Lagaros, N. D., Tsompanakis, Y., & Plevris, V. (2001). Large scale structural optimization: Computational methods and optimization algorithms. *Archives of Computational Methods in Engineering, 8*, 239–301. https://doi.org/10.1007/BF02736645.

[16] Afshar, M. H. (2012). Large scale reservoir operation by constrained particle swarm optimization algorithms. Journal of Hydro-Environment Research, 6(1), 75–87. https://doi.org/10.1016/j.jher.2011.04.003.

[17] Soria, F., & Caddwine, H. (2025). Deep reinforcement learning-based power electronics control for high-efficiency solar and wind systems. National Journal of Renewable Energy Systems and Innovation, 1(4), 26–33.

[18] Abdullah, D. (2025). Optimal sizing and control of microgrid-integrated renewable energy systems using deep reinforcement learning. National Journal of Intelligent Power Systems and Technology, 1(2), 10–17.
[19] Mpamije, L. J., & Freire, G. F. (2025). SiC and GaN device-based power converters for high-efficiency drive applications. National Journal of Electric Drives and Control Systems, 1(3), 1–9.
[20] David, G., & Vana, C. (2025). Loss minimization and thermal optimization in power converters for solar applications. Transactions on Power Electronics and Renewable Energy Systems, 1(3), 15–22.
[21] Barhoumi, E. M., & Luedkea, R. H. (2025). Bidirectional charging strategies for vehicle-to-grid integration using smart storage systems. Transactions on Energy Storage Systems and Innovation, 1(4), 18–25.
[22] Getachew, B., & Alabi, D. (2025). AI-optimized design and advanced control strategies for high-efficiency electrical machines in sustainable power conversion systems. National Journal of Electrical Machines & Power Conversion, 1(2), 9–15.
[23] Usikalua, M. R., & Unciano, N. (2025). Mathematical modeling of epidemic dynamics: Integrating public health and data science. Bridge: Journal of Multidisciplinary Explorations, 1(1), 11-22.
[24] Kigarura, M., & Jakhir, C. (2025). A scalable parallel processing architecture for real-time and energy-efficient biomedical signal analysis in edge-enabled health monitoring systems. Journal of Integrated VLSI, Embedded and Computing Technologies, 2(3), 56–62.
[1] Salameh, A. A., & Ye, K. M. (2025). Design and Optimization of Coarse-Grained Reconfigurable Array (CGRA) Architecture for Efficient Processing-in-Memory (PIM) Systems. *Journal of VLSI Circuits and Systems*, *7*(1), 11-18. https://doi.org/10.31838/jvcs/07.01.02.