EVALUATING THE IMPACT OF PARAMETER SENSITIVITY IN META-HEURISTICS ON RELIABILITY PERFORMANCE

Shakuntla Singla [1], Manisha Rani [2*], Shilpa Rani [3], Umar Muhammad Modibbo [4]

•

[1,2,3]Department of Mathematics and Humanities, MMEC, Maharishi Markandeshwar (Deemed to be University), Mullana, Ambala, India
[4]Department of Statistics and Operations Research, School of Physical Sciences, Modibbo Adama University of Technology, YOLA-NIGERIA
[1]shakus25@gmail.com, [2*]manishasethi2026@gmail.com, [3] gargshilpa46@gmail.com
[4]umarmodibbo@mau.edu.ng

Abstract

The optimizing reliability of a system is a crucial target for any industry to have good values and outcomes with the balanced cost of maintenance. The assessment of costs and reliability in intricate systems, such as Distributed Redundancy Architecture (DRA), is essential for optimal system design. Achieving this involves addressing various constraints to enhance the overall system's reliability. Researchers have explored system dependability and cost optimization challenges, emphasizing the evolution of metaheuristic approaches. Enhancing system resilience and optimizing costs are central objectives of this study, aiming to improve the reliability and cost-efficiency of DRA systems. The study's framework builds upon existing metaheuristic methods, including Moth Flame Optimization (MFO), Whale Optimization Algorithm (WOA), Dragonfly Algorithm (DA), Gazelle Optimization Algorithm (GOA), and Coati Optimization Algorithm (COA). The findings reveal that COA demonstrates superior performance compared to other methods. By employing the Coati Optimization Algorithm, this study effectively addresses cost and reliability-optimization challenges, providing highly efficient solutions. Key metrics such as reliability and cost were compared across the five metaheuristic approaches, showcasing COA's advantages. Notably, the Coati Optimization Algorithm delivers faster resolutions and outperforms its counterparts in reducing costs while enhancing system reliability. The results underscore COA's ability to optimize intricate system parameters more effectively than methods like MFO, WOA, GOA, and DA. This research highlights the value of using COA for improving reliability and minimizing expenses in complex systems. It offers a novel approach to addressing these challenges, establishing COA as a powerful tool for optimizing distributed system architectures. By analyzing comparative solutions, the study found that COA consistently achieved better outcomes, affirming its success in reliability and cost optimization. In conclusion, this research emphasizes the importance of advanced metaheuristic techniques, particularly the Coati Optimization Algorithm, in enhancing the performance of Distributed Redundancy Architectures. By presenting a numerical example, the comparison has been made and the outcomes are shown graphically and in the tabular form to have a clear understanding. The findings contribute significantly to the field by providing a robust method for achieving cost-efficient and reliable system designs.

Keywords: Reliability, Cost, Distributed Redundancy Architecture (DRA), Metaheuristic algorithms.

I. Introduction

Reliability is crucial in complex systems like aerospace and telecommunications. Distributed Redundancy Architecture (DRA) enhances fault tolerance by distributing redundancy across system components. Meta-heuristic algorithms like GA, PSO, and SA optimize reliability but are highly sensitive to parameter settings. Improper calibration can lead to suboptimal redundancy allocation, affecting system performance. Complex systems consist of interconnected elements that exhibit emerging behaviors and challenges. Their reliability is vital as failures can trigger cascading effects. As these systems grow and integrate more components, ensuring dependability while managing costs becomes increasingly important.

II. Literature Review

Optimizing system reliability is crucial for fault-tolerant systems, with Distributed Redundancy Architecture (DRA) enhancing reliability by distributing redundant components. Effective redundancy allocation depends on optimization techniques, often using meta-heuristic algorithms. While these algorithms offer adaptability, their sensitivity to parameter settings remains a challenge, requiring further study. Beji et al. [1] The Redundancy Allocation Problem (RAP) is a well-known reliability optimization challenge that aims to maximize system reliability by strategically adding redundant components to subsystems. The trade-off between cost, weight, and reliability complicates the optimization process, as real-world systems often impose constraints that limit the number and types of components used. RAP has applications in various engineering fields, such as telecommunications, power systems, and defense technologies. Umamaheswari et al. [2] Preventive maintenance is crucial for ensuring the safe and reliable operation of power generation units, which are critical components of the energy infrastructure. The planning and execution of maintenance activities for generating units directly impact their operational efficiency, safety, and reliability. Traditional maintenance strategies, such as time-based and condition based maintenance, have been employed for decades to mitigate equipment failures. However, these strategies are often limited in their ability to balance reliability, risk, and cost. To address these challenges, recent research has focused on developing more advanced approaches, particularly those that integrate reliability and risk assessments into preventive maintenance planning, while ensuring cost-effectiveness.

Dahiya et al. [3] In the field of optimization, finding a global optimum in a complex search space is a critical problem across various domains, including engineering design, machine learning, and operational research. Metaheuristic algorithms have proven to be effective in solving such optimization problems, as they offer powerful and flexible solutions that can efficiently explore and exploit the search space to avoid local optima. Among the numerous metaheuristics developed over the years, the grasshopper optimization algorithm (GOA) has gained attention due to its simplicity and robustness in handling both continuous and discrete optimization problems. However, the traditional GOA can struggle with premature convergence and local optima, especially in high-dimensional and complex search spaces. To address these limitations, hybrid approaches such as the Hybrid Artificial Grasshopper Optimization Algorithm (HAGOA) have been proposed. HAGOA combines the strengths of GOA with other metaheuristic techniques to improve its exploration and exploitation capabilities, thus enhancing its ability to discover the global optimum in a given search space.

Wei et al. [4] In the context of modern industrial and engineering systems, integrated systems—comprising interconnected components working together to achieve specific tasks—have become increasingly complex. The efficient operation of such systems depends not only on their individual components but also on the collective reliability, availability, and performance of the entire system. These factors are crucial for ensuring the continuous and reliable functioning of systems across a range of sectors, including power generation, telecommunications, transportation, and manufacturing. Reliability assessment and availability capacity evaluation play a pivotal role in the design and maintenance of integrated systems, ensuring optimal performance while minimizing downtime and operational costs.Nath and Muhuri [5] The reliability-redundancy allocation problem (RRAP) is a classical optimization problem in the domain of reliability engineering. It involves determining the optimal allocation of redundancy to components in a system to maximize reliability while considering various constraints such as cost, weight, volume, and system complexity. Traditionally, RRAP has been formulated as a multi-objective optimization problem where the key objectives are often to maximize system reliability and minimize cost. However, in practical applications, especially with the increasing complexity of modern systems, RRAP has evolved into a many-objective optimization problem where more than two conflicting objectives must be optimized simultaneously.

A novel evolutionary solution approach based on objective prioritization and constraint optimization has emerged as a promising technique for addressing many-objective reliability-redundancy allocation problems. This approach combines evolutionary algorithms (EA) with mechanisms that prioritize objectives based on their relative importance and incorporate efficient handling of system constraints to explore optimal trade-offs in the solution space. In this literature review, we explore the evolution of RRAP from a multi-objective to a many-objective problem and examine the role of evolutionary algorithms in addressing these challenges. Mirjalili et al. [6-9] The reliability-redundancy allocation problem (RRAP) is a critical optimization task in reliability engineering, focusing on determining the optimal number and configuration of redundant components within a system to maximize reliability while satisfying various constraints such as cost, weight, and volume. The problem becomes more complex when multiple redundancy strategies are available, as these strategies introduce additional decision variables and trade-offs between reliability, cost, and other system requirements. Traditional optimization methods struggle efficiently solve large-scale RRAPs with multiple objectives and constraints, particularly when redundancy strategies can be selected. To address this complexity, metaheuristic algorithms have been widely employed due to their ability to explore vast solution spaces and find near-optimal solutions. Among these, the Water Cycle Algorithm (WCA) has emerged as a promising metaheuristic for solving RRAP, especially when various redundancy strategies are considered.

This literature review explores the application of the WCA for solving RRAPs, emphasizing how its unique flow-inspired mechanism can handle the complexity introduced by multiple redundancy strategies. The review also highlights the advantages of WCA over other traditional and metaheuristic approaches and discusses its applications in various reliability engineering fields. Gandhi and Bhattacharjya [10] Metaheuristic optimization techniques have gained significant attention in recent decades due to their capability of solving complex optimization problems in various fields, such as engineering design, machine learning, and operations research. One such method, the Shuffled Frog Leaping Algorithm (SFLA), is a nature-inspired algorithm based on the behavior of frogs in a pond searching for food. Agushaka et al. [11] The Dwarf Mongoose Optimization (DMO) algorithm is a relatively recent addition to the field of optimization algorithms. Inspired by the social and foraging behavior of the dwarf mongoose (*Helogale parvula*), this algorithm aims to solve complex optimization problems by mimicking the way these animals interact and find resources. This literature review will explore the development, key features, and applications of the DMO algorithm, comparing it with other optimization techniques and highlighting its strengths and limitations. Khodadadi et al. [12] Crystal structure prediction (CSP) is a critical aspect of materials science and chemistry, influencing fields ranging from drug discovery to materials engineering. Recent advancements in computational methods have significantly enhanced our ability to predict crystal structures. Multi-objective crystal structure algorithms (MOCSAs) represent a sophisticated approach to CSP, incorporating multiple criteria into the optimization process. This literature review provides an overview of the development, methodologies, and applications of MOCSAs in crystal structure prediction.

Meta-heuristic algorithms, such as Genetic Algorithms (GA), Simulated Annealing (SA), Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO), have become popular for solving complex optimization problems. These algorithms use stochastic processes and rely heavily on parameters, such as population size, crossover rate, mutation rate, and exploration-exploitation trade-offs, that significantly influence their efficiency Singha et al. [13]. Bhunia et al. [14] Reliability redundancy optimization is a critical aspect of designing systems to ensure operational effectiveness and minimize failure risks. In series systems, where components are arranged sequentially, the reliability of the entire system is dependent on the reliability of each individual component. With multiple choices for redundancy configurations, optimizing these choices becomes complex. Genetic algorithms (GAs) have emerged as a promising method for tackling such optimization problems due to their adaptability and efficiency in exploring large search spaces.

Mettas et al. [15] Reliability allocation and optimization are critical aspects of designing and managing complex systems across various domains, including aerospace, automotive, electronics, and industrial systems. As systems become increasingly complex, ensuring their reliability while optimizing performance and cost presents significant challenges. Abd Alsharify and Hassan [16] the reliability of complex systems is a crucial aspect in various fields, including engineering, manufacturing, and information technology. Optimizing this reliability involves improving the system's performance while minimizing failures and downtime. In recent years, heuristic and metaheuristic optimization algorithms have gained traction in solving complex reliability optimization problems. One such algorithm is the Bat Algorithm (BA), which is inspired by the echolocation behavior of bats. This review explores the application of the Bat Algorithm in optimizing complex system reliability, comparing it with other optimization techniques and highlighting recent advancements and findings.

Ravi [17] Complex systems, characterized by intricate interdependencies and dynamic behaviors, present significant challenges in ensuring reliability and performance. Optimization of such systems is crucial for enhancing their reliability while minimizing costs and downtime. The Great Deluge Algorithm (GDA), inspired by the metaphor of a gradually rising flood, is a robust heuristic method for solving optimization problems.

Coit and Zio [18] the evolution of system reliability optimization has been a dynamic and transformative journey, reflecting the growing complexity of systems and the increasing demand for dependable performance across various industries. Initially, the field emerged from the broader discipline of reliability engineering, which focuses on ensuring that systems perform their intended functions without failure over a specified period. Rocco et al. [19] The application of cellular evolutionary approaches to the reliability optimization of complex systems represents a fascinating convergence of biological inspiration and engineering principles. This approach draws from the concept of cellular automata and evolutionary algorithms to tackle the challenges associated with ensuring the reliability of intricate systems with numerous interacting components. Khorsidi et al. [20] System reliability optimization is a crucial area in engineering and operations research, focusing on improving the performance and dependability of complex systems. As systems become increasingly intricate, traditional optimization methods often fall short due to their inability to handle the non-linearity, high dimensionality, and uncertainty inherent in such problems. Meta-heuristic algorithms, which are inspired by natural processes or human strategies, have emerged as effective tools for addressing these challenges. This review explores two prominent meta-heuristic approaches—Genetic Algorithms (GAs) and Particle Swarm Optimization (PSO)—and compares their efficacy in solving complex system reliability optimization problems.

Zhang et al. [21] Redundancy allocation problems (RAP) and redundant array of parts (RRAP) are critical areas in reliability engineering and system design, particularly in systems with mixed redundancy strategies. The k-Out-of-n: G system, a class of reliability models where the system functions if at least k out of n groups are operational, plays a pivotal role in understanding and optimizing system reliability. This literature review explores the advancements in RAP and RRAP optimization for k-Out-of-n: G systems with mixed redundancy strategies, focusing on novel general models and their implications. Singla et al. [22] has discussed with help of deep learning optimization of packing unit in series with repair and single never fails. Singla et al. [23] sensitive analysis of availability of working time understand of effective pararmeter. Singla et al. [24] with RPGT performance in all three unit in working state two or more state fail then system is reduce state. Singla et al. [25] with help of fuzzy linguistic approach analysis of two unit are repair system.

III. Methodology

I. Moth Flame Optimization Algorithm (MFO)

The Moth Flame Optimization (MFO) algorithm is a metaheuristic optimization algorithm inspired by the natural behavior of moths flying towards a light source.
Here's a basic mathematical representation of the Moth Flame Optimization:

- **Initialization:** A population of moths is initialized randomly. Each moth M_j has a position represented as a vector y_j in the search space, where $y_j \in \mathbb{R}^m$, and m is the number of dimensions of the search space.
- **Fitness Function:** Each moth is evaluated using a fitness function $f(y_j)$. This function is problem-dependent and measures how good the solution represented by y_j is.
- **Update Positions:** Moths are updated based on their positions relative to the position of the best solution (flame) found so far. The position update for moth M_j can be mathematically represented as

$$y_j^{t+1} = y_j^t + \lambda \cdot (y_f - y_j^t) \cdot \exp\left(\frac{-dist(y_j^t; y_f)}{a.t^2}\right)$$

where: y_j^t is the position of moth j at time t; y_f is the position of the best flame (best solution); λ is a scaling factor; $dist(y_j^t; y_f)$ is the Euclidean distance between y_j^t and t is the current iteration

- **Flame Update:** After updating the positions of all moths, the best solution found so far (flame) is updated.
- **Termination:** The algorithm iterates until a stopping criterion is met, such as a maximum number of iterations or a convergence threshold.

II. Whale Optimization Algorithm (WOA)

The Whale Optimization Algorithm (WOA) is a metaheuristic optimization technique inspired by the hunting behavior of humpback whales.

III. Mathematical Representation

Encircling Prey: The position of the whale is updated to encircle the prey (optimal solution) using the following equations: $Y(t+1) = Y^* - A \cdot |C \cdot Y^* - Y(t)|$

where:
$Y(t)$ is the current position of the whale at iteration t; Y^* is the position of the best solution found so far; A and C are coefficient vectors.
Spiral Update: Humpback whales use a spiral motion to search for prey. The position update in the spiral pattern is given by:

$$Y(t+1) = Y^* - D \cdot e^{a \cdot l} \cdot \cos(2\pi l) \cdot Y$$

where: D is the distance between the whale and the prey; a is a constant defining the shape of the spiral; l is a random number between -1 and 1.
Coefficient Vectors: The coefficient vectors A and C are updated as follows:

$$A - 2 \cdot b \cdot r - b$$

$$C - 2 \cdot r$$

where: $b = 2 \cdot (1 - \frac{t}{T})$, which decreases linearly from 2 to 0 over the course of the iterations; r is a random vector between 0 and 1; T is the maximum number of iterations.

IV. Steps of the Whale Optimization Algorithm

- Initialize: Randomly initialize a population of whales and evaluate their fitness.
- Update Coefficients: Calculate A and C.
- Update Position: Depending on a random number, either update the position using the encircling prey strategy or the spiral update strategy.
- Evaluate: Calculate the fitness of the new positions.
- Update Best Solution: Update the best solution found so far.
- Repeat: Iterate the above steps until the stopping criteria are met (e.g., a maximum number of iterations or convergence to a satisfactory solution).

V. Coati- optimization algorithm

Coati Optimization Algorithm (COA) is a relatively recent optimization technique inspired by the behavior of coatis. The algorithm mimics the foraging behavior and social interactions of coatis to explore and exploit search spaces in optimization problems.

VI. Mathematical Representation of COA

Population Initialization: Randomly generate a population of candidate solutions $\{Y_1, Y_2, \ldots, Y_m\}$ within the problem's search space.

VII. Population Matrix of Coatis

$$Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_j \\ \vdots \\ Y_M \end{bmatrix}_{M\times n} = \begin{bmatrix} y_{1,1} & \cdots & y_{1,i} & \cdots & y_{1,n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ y_{j,1} & \cdots & y_{j,i} & \cdots & y_{j,n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ y_{M,1} & \cdots & y_{M,i} & \cdots & y_{M,n} \end{bmatrix}_{M\times n}$$

- **Evaluation**: Compute the fitness value of each candidate solution $f(Y_j)$, $where\ j = 1,2, \ldots, M$.
- **Coati Movement**: Each coati (candidate solution) updates its position based on its previous position and the positions of other coatis. The update rule typically combines exploration and exploitation phases. The update can be represented as:

$$Y_j(t+1) = Y_j(t) + \alpha \cdot random \cdot \left(Y_{best} - Y_j(t)\right) + \beta \cdot random \cdot \left(Y_{mean} - Y_j(t)\right)$$

where: $Y_j(t)$ is the position of the j^{th} coati at iteration t; Y_{best} is the best solution found so far; Y_{mean} is the mean position of all coatis; α and β are parameters controlling the step size,

- **Random** is a random value typically between 0 and 1.
- **Selection**: Update the population by selecting the best solutions based on their fitness values. This step ensures that better solutions are more likely to be retained in the next generation.
- **Termination**: Repeat the evaluation and movement steps until a stopping criterion is met (e.g., a maximum number of iterations, a satisfactory fitness level, or convergence of solutions).

Example

If you're optimizing a function $f(y)$ with COA, your steps might look like:

Initialize Y_j randomly within the search space for $j = 1,2,\ldots,M$.
Evaluate $f(Y_j)$ for each j.
Update positions using:
$Y_j(t+1) = Y_j(t) + \alpha \cdot rand \cdot \left(Y_{best} - Y_j(t)\right) + \beta \cdot rand \cdot \left(Y_{mean} - Y_j(t)\right)$
Evaluate new positions and update the population.
Continue until stopping criteria are met.

VIII. Gazelle Optimization Algorithm

The Gazelle Optimization Algorithm (GOA) is a metaheuristic inspired by gazelle behaviors, particularly escape and foraging. Here's the mathematical representation of its key components:

- **Initialization:** Population Initialization: Randomly initialize a population of gazelles (solutions) within the feasible region. Let Y_j denote the position of the jth gazelle, where j=1,2,…,M_j= 1, 2 and M is the population size. The gazelle population is initialized in matrix form.

$$Y = \begin{bmatrix} y_{1,1} & \cdots & y_{1,i} & \cdots & y_{1,n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ y_{j,1} & \cdots & y_{j,i} & \cdots & y_{j.n} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ y_{M,1} & \cdots & y_{M,i} & \cdots & y_{M,n} \end{bmatrix}_{M\times n}$$

- **Fitness Function:** Evaluate the fitness $f(Y_j)$ of each gazelle based on the objective function of the optimization problem.
- **Update Positions:** The update of the gazelle positions is influenced by their exploration and exploitation behaviors. Let $Y_j(t)$ be the position of the j^{th} gazelle at iteration t. The position update is given by:

$$Y_j(t+1) = Y_j(t) + \alpha \cdot direction(t) \cdot \left(Y_j^* - Y_j(t)\right) + \beta \cdot random(t) \cdot \left(Y_j^{best} - Y_j(t)\right)$$

Where:

α is a scaling factor; $direction(t)$ represents the direction in which the gazelle is moving (could be based on velocity or other parameters); Y_j^* is the position of the j^{th} gazelle's best solution found so far; β is a scaling factor for the random component; $random(t)$ represents a random number; Y_j^{best} is the global best solution found by the entire population

- **Boundary Conditions:** Ensure that the new positions of the gazelles remain within the feasible region of the problem. If a gazelle moves out of bounds, its position is adjusted to stay within the bounds.
- **Termination:** The algorithm terminates after a predetermined number of iterations or when a stopping criterion (such as convergence or no significant improvement) is met.

IX. Dragonfly Algorithm

The Dragonfly Algorithm is a nature-inspired optimization algorithm that is based on the foraging behavior of dragonflies.

- **Initialization:** Define the initial population of dragonflies, where each dragonfly represents a potential solution in the search space.
  $Y_j^0 = y_j \quad for\ j = 1,2,\dots,M$ where Y_j^0 is the position of the j^{th} dragonfly at the initial iteration, and M is the total number of dragonflies.
- **Movement Model:** Update the position of each $Y_j^{(t)}$ dragonfly based on the following update rule:
  $$Y_j^{t+1} = Y_j^{(t)} + \alpha \cdot \left(Y_{best}^{(t)} - Y_j^{(t)}\right) + \beta \cdot random\_vector$$
- **Convergence Check:** Check if the stopping criteria are met (e.g., a maximum number of iterations or convergence to a threshold). If not, return to the movement model step; otherwise, return the best-found solution in figure 1 and 2.

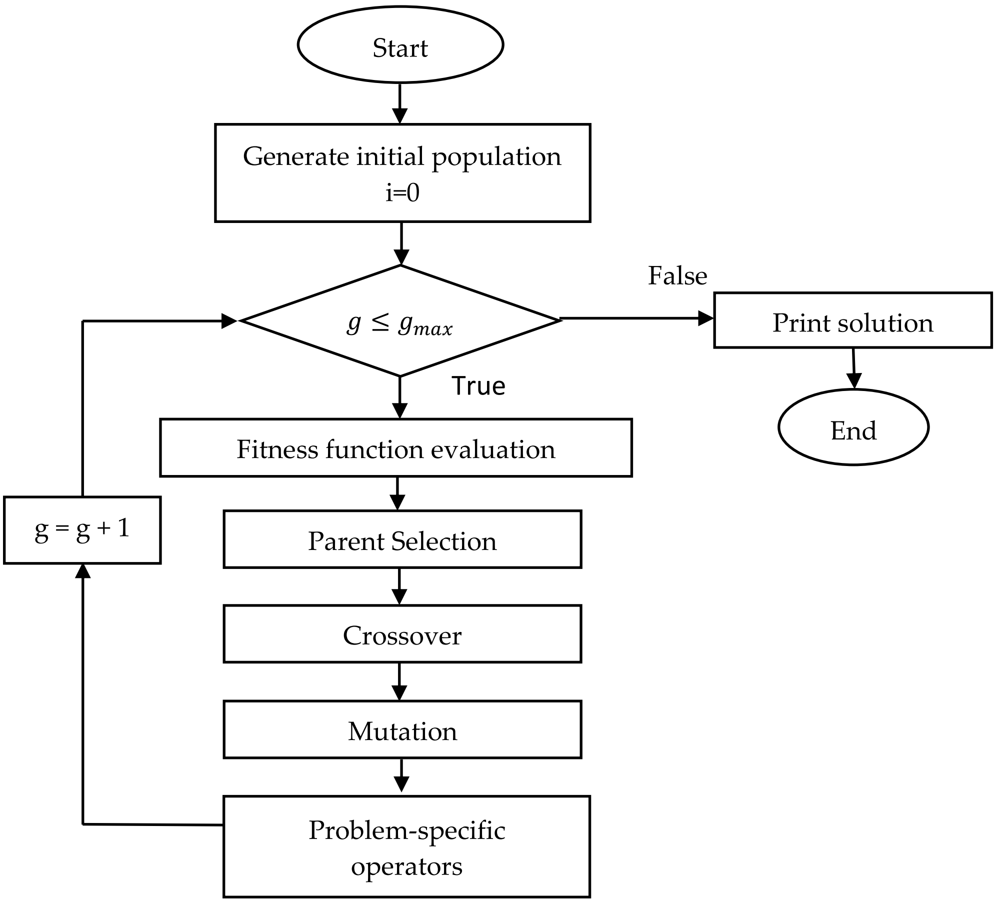

Figure 1: *Flowchart of Metaheuristic Algorithms*

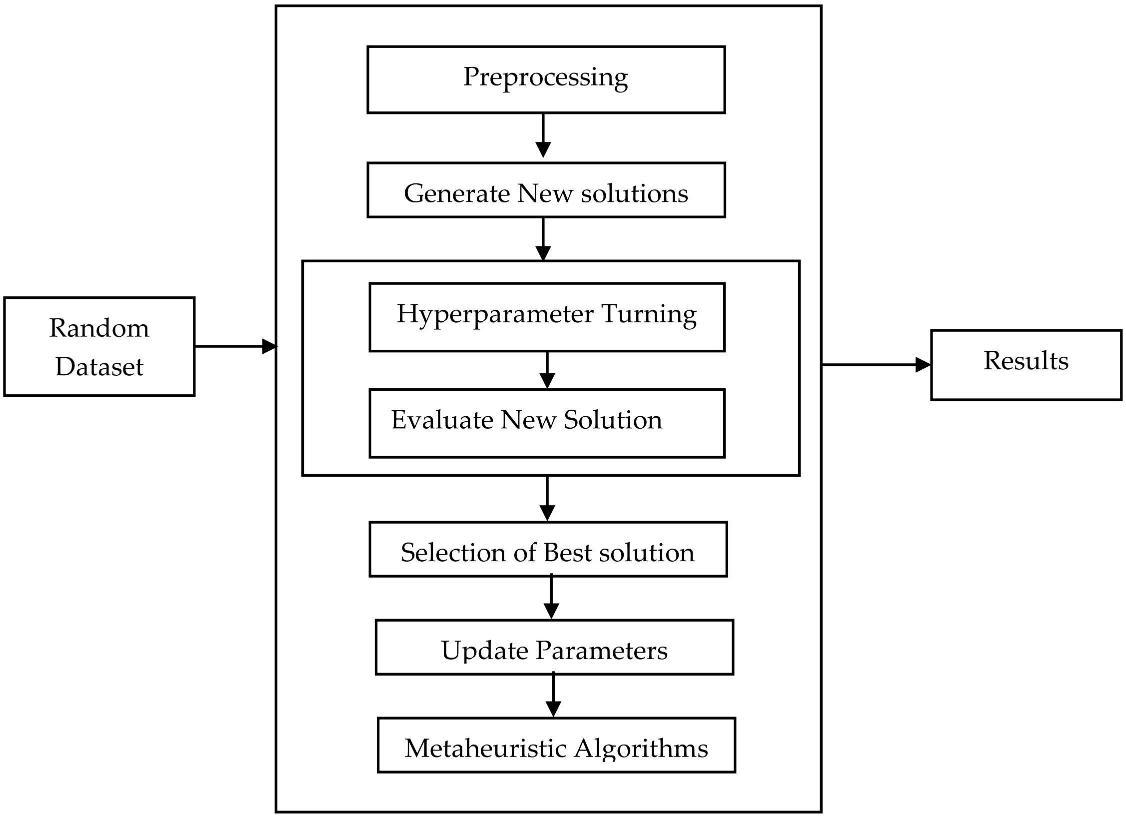

Figure 2: *Flowchart of Distributed Redundancy Architecture (DRA) based on the metaheuristic algorithms*

IV. Dataset Description

The dataset includes reliability performance data from simulations on DRA-based systems. It captures the impact of meta-heuristic parameters, including MFO, WOA, DA, GOA, and COA, on system reliability. Experiments assess sensitivity to parameter changes and redundancy configurations across system sizes and failure scenarios, as shown in Tables 1 and 2.

V. Results and Discussion

This study highlights how parameter sensitivity in meta-heuristic algorithms affects reliability optimization in DRA-based systems. While these algorithms efficiently explore complex solutions, their performance varies with parameter settings. This sensitivity influences redundancy allocation, directly impacting system reliability, as shown in Tables 1 and 2. Reliability performance was measured using MTTF, Time, and SRI, as shown in Table 3. Systems with well-calibrated parameters outperformed those with static or poorly chosen settings. Adaptive parameter tuning improved redundancy allocation, enhancing fault tolerance and system reliability. In contrast, static configurations led to uneven redundancy, making certain nodes more vulnerable and reducing fault tolerance.

Table 1: *Parameter*

R_s	A_i	W_i	P
(0-.100)	(50-100)	(0-100)	(0-.68)

Table 2: *Parameter of meta-heuristic algorithms*

Population size	*Crossover rate*	*Mutation rate:*	Acceleration coefficients	Swarm size
(20–100)	(0.6–0.9)	(0.01–0.1)	(1.5–2.5)	(20–50)

Table 3: *Reliability performance was measured using metrics*

Mean Time to Failure (MTTF)	System Reliability Index (SRI)	Time of Program(ms)	Algorithm
.94	.963	1200	WOA
.93	.965	1250	DA
.92	.974	1264	GOA
.91	.978	1274	MFO
.90	.980	1287	COA

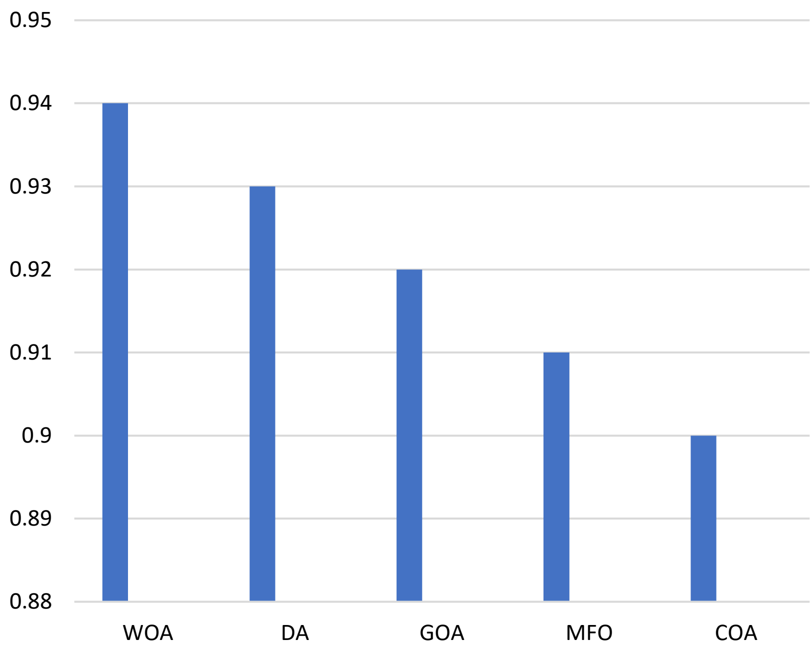

Figure 3: *Comparison Among Different Method According to MTTF*

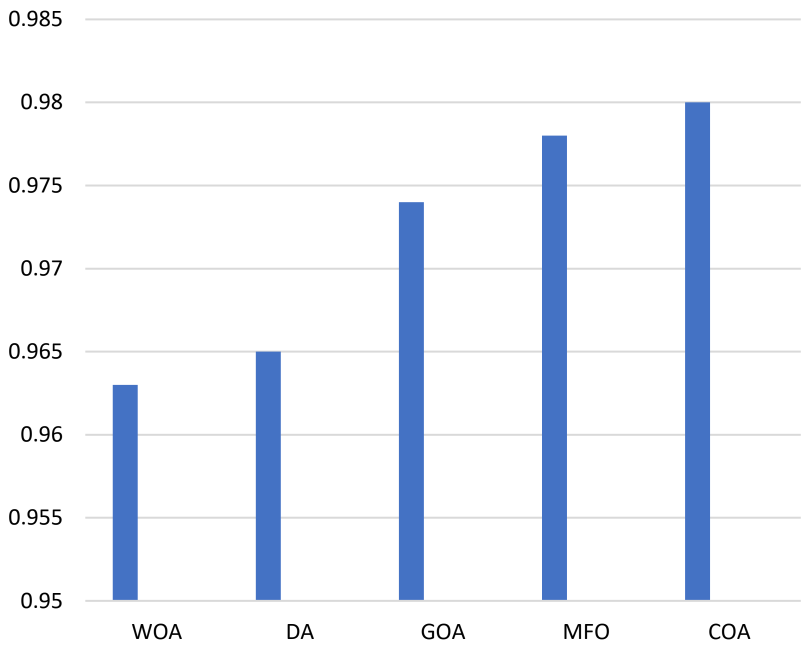

Figure 4: *Comparison Among Different Method According to SRI*

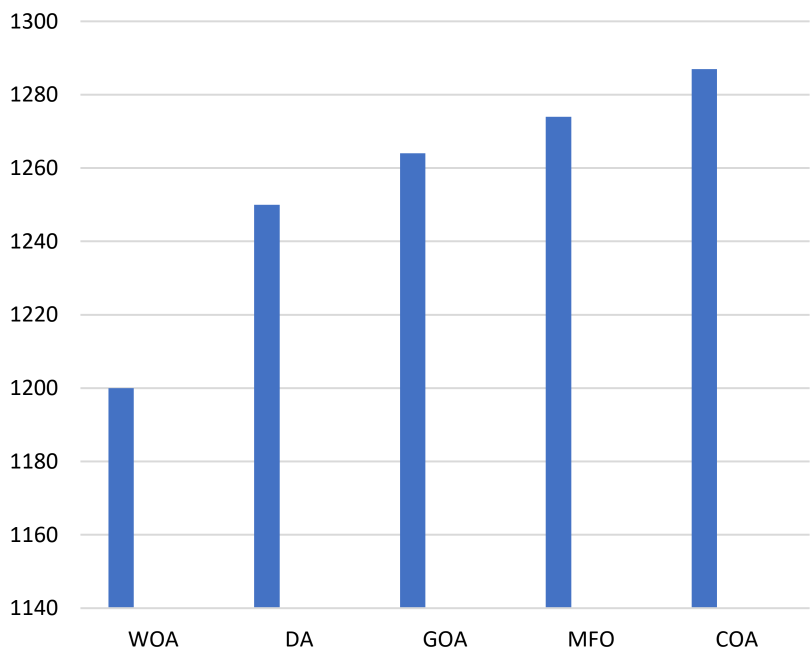

Figure 5: *Comparison Among Different Method According to Time*

The experiment utilized Tables 1 and 2, with results presented in Table 3. The Whale Optimization Algorithm (WOA) achieved the highest Mean Time to Failure (MTTF), while the Coati Optimization Algorithm (COA) had the highest System Reliability Index (SRI) and the longest execution time. Based on Table 3, Figures 3, 4, and 5 can be drawn to visualize these findings.

VI. Conclusion

This study introduces five innovative methods to reduce the total expense of complex systems, focusing on Distributed Redundancy Architecture (DRA). The framework's effectiveness was evaluated and compared to each optimization strategy. The COA method demonstrated the best MTTF and SRI among all approaches, proving its efficiency. Despite having the lowest MTTF, COA optimized cost and reliability parameters effectively. The findings confirm that COA outperforms other methods, offering a more efficient solution for reducing system costs.

Conflicts of interests: The authors declare that there are no conflicts of interest.

References

[1] Beji, N., Jarboui, B., Eddaly, M., and Chabchoub, H. (2010). A Hybrid Particle Swarm Optimization Algorithm for the Redundancy Allocation Problem. *Journal of Computational Science*, 159-167.

[2] Umamaheswari, E., Ganesan, S., Abirami, M., and Subramanian, S. (2018). Reliability/risk centered cost-effective preventive maintenance planning of generating units. *International Journal of Quality and Reliability Management*, 2052-2079.

[3] Dahiya, B. P., Rani, S., and Singh, P. (2019). A hybrid artificial l grasshopper optimization (HAGOA) meta-heuristic approach: A hybrid optimizer for discover the global optimum in given search space. *International Journal of Mathematical Engineering and Management Sciences*, 471-482.

[4] Wang, F., Araújo, D. F., and Li, Y. F. (2023). Reliability assessment of autonomous vehicles based on the safety control structure. *Journal of Risk and Reliability*, 389-404.

[5] Nath, R., and Muhuri, P. K. (2024). A novel evolutionary solution approach for many-objective reliability-redundancy allocation problem based on objective prioritization and constraint optimization. *Reliability Engineering and System Safety*, 109835-109845.

[6] Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. *Knowledge-Based Systems*, 228-249.

[7] Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective discrete and multi-objective problems. *Neural computing and applications*, 1053-1073.

[8] Mirjalili, S., and Lewis, A. (2016). The whale optimization algorithm. *Advances in engineering software*, 51-67.

[9] Mirjalili, S., Mirjalili, S. M., and Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software*, 46-61.

[10] Gandhi, B. R., and Bhattacharjya, R. K. (2020). Introduction to shuffled frog leaping algorithm and its sensitivity to the parameters of the algorithm. *Nature-Inspired Methods for Metaheuristics Optimization: Algorithms and Applications in Science and Engineering*, 105-117.

[11] Agushaka, J. O., Ezugwu, A. E., and Abualigah, L. (2023). Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer. *Neural Computing and Applications*, 4099-4131.

[12] Khodadadi, N., Azizi, M., Talatahari, S., and Sareh, P. (2021). Multi-objective crystal structure algorithm (MOCryStAl): Introduction and performance evaluation. *IEEE Access*, 117795-117812.

[13] Singha, A. K., Pathak, N., Sharma, N., Tiwari, P. K., and Joel, J. P. C. (2022). COVID-19 disease classification model using deep dense convolutional neural networks. *In Emerging Technologies in Data Mining and Information Security: Proceedings of IEMIS,* 671-682

[14] Bhunia, A. K., Duary, A., and Sahoo, L. (2017). A genetic algorithm-based hybrid approach for reliability redundancy optimization problem of a series system with multiple-choice. *International Journal of Mathematical, Engineering and Management Sciences*, 185-212.

[15] Mettas, A. (2000). Reliability allocation and optimization for complex systems. In Annual reliability and maintainability symposium. (2000). *International symposium on product quality and integrity*, 216-221.

[16] Abd Alsharify, F. H., and Hassan, Z. A. H. (2022). Optimization of complex system reliability: Bat algorithm-based approach. *International Journal of Health Sciences*, 6-7.

[17] Ravi, V. (2004). Optimization of complex system reliability by a modified great deluge algorithm. *Asia-Pacific Journal of Operational Research*, 487-497.

[18] Coit, D. W., and Zio, E. (2019). The evolution of system reliability optimization. *Reliability Engineering and System Safety*,192-203.

[19] Rocco, C. M., Miller, A. J., Moreno, J. A., and Carrasquero, N. (2000). A cellular evolutionary approach applied to reliability optimization of complex systems. In Annual Reliability and Maintainability Symposium. 2000 Proceedings. *International Symposium on Product Quality and Integrity* ,210-215.

[20] Khodadadi, N., Azizi, M., Talatahari, S., and Sareh, P. (2021). Multi-objective crystal structure algorithm (MOCryStAl): *Introduction and performance evaluation IEEE Access*,117795-117812.

[21] Zhang, J., Lv, H., and Hou, J. (2023). A novel general model for RAP and RRAP optimization of k-out-of-n: G systems with mixed redundancy strategy. *Reliability Engineering and System Safety*, 229-235.

[22] Singla, S., Rani, S., Modibbo, U. M. and Ali, I. (2023). Optimization of System Parameters of 2:3 Good Serial System using Deep Learning. *Reliability Theory and Applications*,670-679.

[23] Singla, S., Mangla, D., Panwar, P. and Taj, S. Z. (2024). Reliability Optimization of a Degraded System under Preventive Maintenance using Genetic Algorithm. *Journal of Mechanics of Continua and MathematicalSciences*,1-14.

[24] Singla, S., Rani, S. (2023). Performance optimization of 3:4 good system. *IEEE second international conference,* 979-8-3503-4383-0.

[25] Singla, S., Mangla, D., Dhawan, P., Ram, G. (2024). Reliability analysis of a two-unit repairable system using fuzzy linguistic approach. *Applications of fuzzy theory in applied sciences and computer applications,*45-57.