DOI: 10.61954/2616-7107/2024.8.1-1 ISSN 2616-7107

UDC: 519
JEL: C60, C61, C69

ANALYSIS OF RELIABILITY AND COST OF COMPLEX SYSTEMS WITH METAHEURISTIC ALGORITHMS

Shivani Choudhary
Graphic Era (Deemed to be University),
Dehradun, India
ORCID iD: 0009-0000-5814-3141

Mangey Ram*
Graphic Era (Deemed to be University),
Dehradun, India
ORCID iD: 0000-0002-8221-092X

Nupur Goyal
Graphic Era (Deemed to be University),
Dehradun, India
ORCID iD: 0000-0002-7468-3508

Seema Saini
Graphic Era (Deemed to be University),
Dehradun, India
ORCID iD: 0000-0002-1597-1032

*Corresponding author:
Email: mangeyram@gmail.com

Received: 17/10/2023
Accepted: 28/02/2024

DOI: 10.61954/2616-7107/2024.8.1-1

Introduction. The assessment of the reliability and cost of complex systems, such as Complex Bridge Systems (CBS) and Life Support Systems in Space Capsules (LSSSC), is fascinating. To achieve the ideal system design through diverse constraints and increase overall system reliability, researchers have extensively explored system reliability and cost optimization problems. Hence, the significant advancement in metaheuristic methods is the primary source of further system reliability and cost optimization process refinement.

Aim and tasks. This research attempts to enhance the reliability and cost of complex systems named CBC and LSSSC has been presented.

Results. The structure is based on few recent metaheuristic techniques, such as Moth Flame Optimization (MFO), Whale Optimization Algorithm (WOA), Gazelle Optimization Algorithm (GOA), Dragonfly Algorithm (DA), and Coati Optimization Algorithm (COA). Comparing the acquired findings to those found in other proposed techniques demonstrates the usefulness of a methodology based on COA. The proposed COA algorithm exhibits enhanced efficiency by offering superior solutions to reliability and cost-optimization problems. In addition, a non-parametric Friedman ranking was performed for validation. The results of this research are based on improving the reliability of the parameters and decreasing complex systems' costs used by the five metaheuristic methods. Observing the convergence graph, Friedman ranking, statistical results test, and tables determined that COA is the most effective algorithm for a complex system's cost and reliability parameters compared to other existing approaches, and also provided a faster solution.

Conclusions. This study proposes unique ways to reduce costs while increasing parameter reliability in complex systems. After analysing the comparative solution, the authors found that when comparing these approaches (GOA, DA, MFO, WOA, and COA), the COA provided the best minimum solution for the cost and reliability of complex systems. Hence, the suggested COA procedure was more successful than that described in this study.

Keywords: CBS, LSSSC, cost, reliability, metaheuristic algorithms.

1. Introduction.

A complex system is a network of interrelated elements or components that frequently defy basic expectations and exhibit emerging features. These systems exist in many spheres, including the natural world, human civilization, and technology. Because of their ability to shed light on the complexity and unpredictability of the world, these systems are an exciting and crucial field of expertise for educational institutions from various disciplines.

Complex systems are characterized by their intricate interdependencies, where the failure of a single component can trigger a cascade of disruptions throughout the entire system.

The reliability and cost of complex systems have become increasingly critical concerns in today's interconnected and technologically advanced world. As systems grow in scale and incorporate intricate networks of components, processes, and interactions, their reliable functioning becomes paramount. Whether it is a sprawling industrial facility, a sophisticated transportation network, or an intricate software ecosystem, the reliability of these systems can have far-reaching consequences on efficiency, safety, and even societal well-being (Hwang et al., 1981)

Hence, to calculate the cost and reliability of complex systems, the two issues taken in this study as examples are the Complex Bridge System (CBS) and the Life Support System in Space Capsule (LSSSC) (Tillman et al., 1970; Kumar et al., 2023). CBS refers to a bridge or network of bridges with intricate structural configurations and numerous interconnected components.

These systems are characterized by a high level of complexity and are subjected to rigorous analysis to assess their dependability and resilience. The study of CBS using reliability theory involves evaluating the structural integrity, performance, and safety of these critical pieces of infrastructure. Engineers and reliability experts aim to ensure that these CBS meet stringent reliability standards, thereby minimizing risk and cost.

LSSSC is a critical technology and equipment that enables astronauts to survive and function in hostile space environments.

These systems are designed to provide astronauts with the essential elements needed for life, such as air, water, and temperature regulations while managing waste and ensuring their overall well-being. LSSSC in space capsules is a marvel of engineering and innovation, allowing astronauts to embark on lengthy missions beyond the Earth's atmosphere with confidence in their safety and sustainability (Kumar et al., 2023).

This study suggests five approaches for analysing the cost and reliability of complex systems: COA, DA, MFO, GOA, and WOA. The objectives of this research paper as a solution to the complex system's cost and reliability are highlighted below.

• Utilizing CBC and LSSSC as examples for designing complex systems, which allows the authors to quickly determine cost and reliability.

• Five novel algorithms, namely COA, DA, MFO, GOA, and WOA, were introduced to deliver the optimum results.

• Nonlinear constraint optimization problems are utilized to create objective functions, with the main objective of reducing the cost when reliability is a constraint. Improving the cost and reliability of the CBC and LSSSC.

• To analyse the effectiveness of the suggested approaches, the cost function and its results are compared with those of each of the five approaches to evaluate which method provides the best solution. Furthermore, it offers a Friedman ranking test and a performance evaluation of the planned work's cost, duration, and efficiency.

The following components of the study are organized as below: The literature on modern methods and complex systems is presented in Section 2. Section 3 gives the optimization strategies, and the problem's mathematical formulation is shown in Section 4. The model analysis and experimental results are explained in Section 5. Conclusions and suggestions for further work are given in Section 6.

2. Literature Review.

This segment summarizes contemporary research on complex systems and optimization methods (Metaheuristic Methods).

2.1. Metaheuristic Approaches.

Beji et al. (2010) investigated a strategy that combined particle swarm optimization with local search techniques to study the problems associated with series-parallel redundant reliability concerns while considering component mixing. Umamaheswari et al. (2018) utilized the Ant Lion Optimizer (ALO) to achieve the best possible maintenance schedules in context of preventive maintenance scheduling (PMS). Dahiya et al. (2019) presented a meta-heuristic method to improve exploration and exploitation inside a search space known as hybrid artificial grasshopper optimization (HAGOA). Wei and Liu (2023) discussed optimising reliability for parallel and series systems when random shocks and constituents come from different discrete subpopulations within a heterogeneous population.

Durgadevi and Shanmugavadivoo (2023) improved the reliability of the electricity system by using a metaheuristic-based algorithm and reliability indices before and after adding DGs to the feeder system. Using PSO, Choudhary et al. (2023) optimized the cost and reliability of a series-parallel system.

Nath and Muhuri (2024) rephrased the prioritized many objective RRAPs (PrMaORRAP) and offered a customized evolutionary solution to address the reformulated issue.

Many researchers have recently presented a variety of metaheuristic methods: Gray wolf optimizer (GWO) (Mirjalili et al., 2014; Pahuja, 2020), Water Cycle Algorithm (Mahdavi-Nasab et al., 2020), Shuffled Frog Leaping Algorithm (SFLA) (Gandhi and Bhattacharjya, 2020), Moth Flame Optimization (MFO) algorithm (Mirjalili, 2015; Sahoo et al., 2023), Whale Optimization Algorithm (WOA) (Mirjalili and Lewis, 2016; Dao et al., 2016), Dwarf Mongoose Optimization Algorithm (DMOA) (Agushaka et al., 2022), Cat and Mouse Based Optimizer (CMBO) (Dehghani et al., 2021), Coati Optimization Algorithm (COA) (Dehghani et al., 2023), Gazelle Optimization Algorithm (GOA) (Agushaka et al., 2023), Dragonfly Algorithm (DA) (Mirjalili, 2016), Crystal Structure Algorithm (CSA) (Khodadadi et al., 2021), and Stochastic Paint Optimizer (SPO) (Kaveh et al., 2020), which are few meta-heuristics approaches used in reliability

2.2. Complex System Optimization.

A genetic algorithm was utilized to tackle the inherited problem after it was discovered using the PSO technique. Hariesh (2021) employed an interactive approach to solving the bi-objective reliability-cost problem of a series-parallel system.

Bhunia et al. (2017) presented the characteristics of a hybrid approach to handle the problems related to reliability optimization for a series system with multiple-choice restrictions and parallel redundancy. The goal is to minimize system costs while maintaining a minimum level of system reliability and maximizing system reliability within budgetary constraints.

Mettas (2000) identified an ideal component reliability that reduces the system's cost while fulfilling the system's reliability target requirement. Abd Alsharify and Hassan (2022) used three alternative cost functions to determine the most excellent reliability for a complicated network while maintaining low costs.

The deluge method, a global optimization meta-heuristic, was enhanced and used by Ravi (2004) to improve the robustness of complex systems. To show how well the method works, two optimization problems are solved: (i) optimum distribution of redundancy in a multi-stage mixed system subject to weight and cost constraints, and (ii) Optimizing the complex systems' reliability with weight and cost constraints.

A variety of system reliability optimization challenges are taken into consideration by Coit and Zio (2019), including the redundancy allocation problem (RAP), and reliability-redundancy allocation problem (RRAP).

Redundancy, component reliability, and both redundancy and component reliability are the three optimisation challenges that Rocco (2000) covers.

This study provides a novel method that employs cellular evolutionary strategies (CES) to address each of these difficulties. An optimization approach for multi-state weighted k-out-of-n systems is presented by Khorshidi and Nikfalazar (2015) to decrease costs and increase system reliability.

Abdullah and Hassan (2020) measured the reliability of complex systems as a complex system utilizing minimal pathways. The dependability of a system refers to its capacity to operate adequately for a particular duration under precise circumstances (Wang et al., 023). To highlight the necessity for additional studies using actual case studies, a short deliberation on integrating human aspects into systems to examine lower levels of autonomous cars is also contained.

Zhang et al. (2023) suggested a new general reliability model enhancement model called k-out-of-n: G subsystems based on a continuous-time Markov chain and a mixed redundancy approach. RAP and RRAP optimizations were performed using a pseudo-parallel genetic algorithm (PPGA). Negi et al. (2021) optimized the complex system's reliability and cost using GWO.

3. Methodology.

This segment gives the details about the proposed techniques algorithm.

3.1. Moth Flame Optimization Algorithm (MFO).

The behaviour of moths drawn to light sources inspired the development of the Moth-Flame Optimisation (MFO) algorithm, a metaheuristic optimization technique inspired by nature. Mirjalili (2015) put forth the proposal. Swarm intelligence methods include MFO, which is made to solve optimization problems by imitating moth behavior in the wild. Here is a more concise mathematical representation of the MFO algorithm:

I. Initialization. Set a population of moths' initial places at random:

$$M = \{x_1, x_2, \dots\dots\dots.\ x_n\},$$

where x_i represents the ith Moth's location within the search area.

Evaluate the fitness of each Moth in the population: $f(x_i)$.

II. Main Loop. Set the maximum iterations: *max_iterations*.

Prepare the iteration counter: *iteration* = 0. Initialize global best solution (brightest flame):

$$best\_solution = argmin\ (f(x_i)).$$

III. While Loop.

While *iteration* < *max_iterations*).

Sort Moths based on their fitness (brightness): $M = sort\ (M, f(x_i))$.

Update the global best solution (brightest flame): $best\_solution = argmin\ (f(x_i))$.

*IV. For Each Moth.*For each Moth x_i in the population:

Calculate light intensity (fitness) for the current position: $I(x_i) = f(x_i)$.

Calculate the distance to the brightest flame:

$d\ (x_i, best\_solution) = \|\ x_i - best\_solution\|$.

Calculate the movement vector towards the brightest flame: $move\_vector =$

$(x_i - best\_solution)\ /\ d(\|\ x_i - best\_solution) + epsilon))$,

where *epsilon* is a small positive constant to avoid division by zero.

Apply a random perturbation to the Moth's position for exploration:

$rand\_vector = random\_vector\ ()$.

Update the position of the Moth:

$x_i = x_i + step\_size * move\_vector + exploration\ step\_size\ rand\_vector$,

where, *step_size* and *exploration step*_size are control parameters.

Evaluate the fitness of the new position:

$f\_new = f(x_i)$.

If *f_new* is better than the previous fitness, update x_i and $f(x_i)$.

V. Optional Mechanisms.

Increment the iteration counter:

$$iteration = iteration + 1.$$

End of While Loop.

VI. Result.

Return the best solution found: *best_solution*.

3.2. Whale Optimization Algorithm (WOA).

WOA is a bio-based optimization algorithm introduced by Mirjalili (2016). The WOA method draws inspiration from the social behaviour of humpback whales and hunting strategies, which work together to catch prey.

The main idea behind the WOA is to mimic the hunting behaviour of Whales to obtain optimal resolutions to optimization problems**.**

Certainly, here's a concise mathematical representation of the WOA:

I. Initialization. Initialize a population of Whales:

$$X = \{x_1, x_2, \ldots\ldots\ldots x_n\},$$

where x_i depicts the ith Whale's location within the search area.

Define the search space boundaries: x_{min} and x_{max}.

II. Evaluation Objective Function.

Evaluate the fitness of every one Whale's position:

$$f(x_i) \text{ for } i = 1, 2, \ldots\ldots N.$$

III. Main Optimization Loop.

While a termination requirement is not satisfied.

(I). Exploration Phase

Select a leader Whale with the best fitness: $x_{leader} = min\ (f(x_i))$.

Update the positions of the follower Whales using the encircling equation:

$$D = |\, x_{leader} - x_i \,|,$$

A = 2. r_1- 1 (where r_1 - random number belongs to [0, 1]),

C = 2. r_2 (where r_2 - another random number from [0, 1]),

$b = 1$ (constant),

$p = (A\ .\ D\ .\ e \wedge b\ .\ A.\ cos\ (C\ .\ D))$.

Update the position of the ith follower Whale: $x_i = x_{leader} - p$.

(II). Exploitation Phase

Update the positions of the follower Whales using the spiral equation:

$D = |\, x_{leader} - x_i \,|$,

A = 2. r_1- 1 (where $r_1$is a random number between 0 and 1),

C = 2. r_2 (where r_2 is another random number between 0 and 1),

$b = 1$ (constant),

$p = (A\ .\ D\ .\ e \wedge b\ .\ A.\ cos\ (C\ .\ D))$.

Update the position of the ith follower Whale:

$x_i = x_{leader} - p$.

IV. Boundary Check.

Ensure that the positions of Whales remain within the defined search space boundaries:

x_i = clip (x_{min}, x_{max}, x_i).

V. *Update ideal Result.*

After each iteration, update the ideal result found so far: $x_{best} = min\ (f(x_i)$.

This mathematical representation summarizes the key components and equations of the WOA. This provides a structured overview of the main operations of the algorithms.

3.3. Coati – Optimization Algorithm (COA).

Dehghani et al. (2021) presented a novel bio-inspired metaheuristic optimization technique in 2023 called the Coati-Optimization Algorithm (COA). In this metaheuristic, the population-based COA method considers the Coatis members of the population. The choice variables' values are determined by where each coati is in the search area. As such, the Coatis point of view offers a potential resolution to this issue in the COA. The mathematical representation of the COA is:

I. Initialization.

$$X_i: = x_{ij} = lb_j + r\ (ub_j - lb_j),$$

where X_i= the position of the ith Coati in the search area*,* x_{ij} = value of the jth decision variable, n, m = Coati number and decision variable number respectively, r = random real number from [0, 1], lb_j, = ith decision variable's lower bound, ub_j = ith decision variable's upper bound.

II. The population matrix of Coatis.

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_i \\ \vdots \\ X_N \end{bmatrix}_{N\times m} = \begin{bmatrix} x_{1,1} \cdots x_{1,j} \cdots x_{1,m} \\ \vdots \ \cdots \vdots \ \cdots \vdots \\ x_{i,1} \cdots x_{i,j} \cdots x_{i,m} \\ \vdots \ \cdots \vdots \ \cdots \vdots \\ x_{N,1} \cdots x_{N,j} \cdots x_{N,m} \end{bmatrix}_{N\times m}$$

Whenever feasible, several values for the objective function of the issue are evaluated.

Solutions are placed in decision variables:

$$F = \begin{bmatrix} F_1 \\ \vdots \\ F_i \\ \vdots \\ F_N \end{bmatrix}_{N\times 1} = \begin{bmatrix} F(X_1) \\ \vdots \\ F(X_i) \\ \vdots \\ F(X_N) \end{bmatrix}_{N\times 1}$$

F is obtained the objective function's vector, and Fi is obtained the objective function's value of the determined using the ith coati.

III. Updating the position of COA.

Here update on the new positions of Coats using the exploration and exploitation phase

(I) Exploration phase (Hunting and attacking strategy on *Iguana*)

$$X_i^{P_1} : X_i^{P_1} = X_{i,j} + r.(Iguana_j - I.X_{i,j}),$$
$$i = 1, 2, \ldots\ldots\ldots\ldots \left\lfloor \frac{N}{2} \right\rfloor,$$
$$j = 1, 2, \ldots\ldots. m.$$

Once the *Iguana* touches down, its location within the search region is randomly chosen.

Based on this random position, ground-based Coats move into the virtual search space:

$$Iguana^G : Iguana_j^G = lb_j + r.(ub_j + lb_j),$$
$$X_i^{P_1} : X_{i,j}^{P_1} = \begin{Bmatrix} x_{i,j} + r.(Iguana_j^G - I.x_{i,j}) & F_{Iguana^G} < F_i \\ x_{i,j} + r.(I.x_{i,j} - Iguana_j^G), & else, \end{Bmatrix},$$

$$for \quad i = \left\lfloor \frac{N}{2} \right\rfloor + 1, \left\lfloor \frac{N}{2} \right\rfloor + 2, \ldots\ldots N \quad and \quad j = 1, 2, \ldots\ldots m.$$

Updated new position-

$$X_{i,j}^{P_1} = \begin{Bmatrix} x_i^{P_1}, & F_i^{P_1} < F_i, \\ x_i, & else. \end{Bmatrix},$$

Here, $X_{i,j}^{P_1}$ = New position for the ith Coat,

$F_i^{P_1}$ = Objective function value,

r= Random-real number from [0, 1],

Iguana = *Iguana*'s position in search space,

I = Randomly integer number between {1, 2},

$Iguana^G$ = Position of the *Iguana* on the ground,

F_{Iguana^G} = Value of objective function,

$\lfloor . \rfloor$ = Floor function.

(II). Exploitation phase

Relative to each Coati's location, a random position is produced using the equation below:

$$lb_j^{local} = \frac{lb_j}{t},$$
$$ub_j^{local} = \frac{ub_j}{t}, \ where \quad t = 1, 2, \ldots\ldots. T.$$
$$X_i^{P_2} : X_{i,j}^{P_2} + (1-2r).(lb_j^{local} + r.(ub_j^{local} - lb_j^{local})),$$
$$where \ \ i = 1, 2, \ldots\ldots N, \quad j = 1, 2, \ldots\ldots\ldots m.$$

For new position:

$$X_i = \begin{Bmatrix} x_i^{P_2}, & F_i^{P_2} < F_i, \\ x_i, & else, \end{Bmatrix},$$

Here, $X_i^{P_2}$ = New position for the ith Coat,

$F_i^{P_2}$ = Objective function value,

r= Random real number from [0, 1],

t= Iteration counter'

$lb_j^{local}, ub_j^{local}$ = the jth decision variable's lower local and higher local, respectively,

lb_j, ub_j = Decision variable's lower bound and upper bound, respectively.

IV. Update the best solution found.

V. End COA algorithm.

3.4. Gazelle Optimization Algorithm (GOA).

The GOA is a natural-based optimization method introduced by Agushaka in 2023.

The mathematical representation of the GOA is given as:

I. Initialization.

Initialize a population of Gazelle in matrix form,

$$X = \begin{bmatrix} x_{1,1} \cdots x_{1,j} \cdots x_{1,m} \\ \vdots \quad \cdots \vdots \quad \cdots \vdots \\ x_{i,1} \cdots x_{i,j} \cdots x_{i,m} \\ \vdots \quad \cdots \vdots \quad \cdots \vdots \\ x_{N,1} \cdots x_{N,j} \cdots x_{N,m} \end{bmatrix}_{N \times m}$$

X = current candidate position generated by randomly using the below equation,

$$X_{i,j} = rand.(ub_j - lb_j) + lb_j,$$

where, ub_j, lb_j = Upper and lower bound, $rand$= random number, $X_{i,j}$ = Position of the jth dimension of the ith population.

II. Compute the fitness function of Gezelle.

III. Construct Elite Gazelle matrix.

The strongest Gazelles are selected as top Gazelles to build an Elite matrix for selecting the next step for the Gazelles, and the best outcomes are established after each iteration:

$$Elite = \begin{bmatrix} x'_{1,1} \cdots x'_{1,j} \cdots x'_{1,m} \\ \vdots \cdots \vdots \cdots \vdots \\ x'_{i,1} \cdots x'_{i,j} \cdots x'_{i,m} \\ \vdots \cdots \vdots \cdots \vdots \\ x'_{N,1} \cdots x'_{N,j} \cdots x_{N,m} \end{bmatrix}_{N\times m}$$

where, $x'_{i,j}$ = Top Gazelle vector.

IV. Start the main loop of GOA.

Here update on the new position using exploitation and exploration phases.

(I) Exploitation Phase.

Update the position of Gazelles,

$$\overrightarrow{Gazelle_{l+1}} = \overrightarrow{Gazelle_l} + s.\vec{R}*.\overrightarrow{R_B}.*$$
$$(\overrightarrow{Elite_l} - \overrightarrow{R_B}.*\overrightarrow{Gazelle_l}),$$

where, $\overrightarrow{Gazelle_l}$ = Solution at the current iteration,

$\overrightarrow{Gazelle_{l+1}}$ = Solution of the next iteration,

$\overrightarrow{R_B}$ = Vector containing random number,

R= Vector of a uniform random number in [0, 1],

s=Grazing speed of the Gazelles.

(II). Exploration Phase

Update the position of Gazelles in the exploration phase,

$$\overrightarrow{Gazelle_{l+1}} = \overrightarrow{Gazelle_l} + S.\mu.\vec{R}*.\overrightarrow{R_L}.*$$
$$(\overrightarrow{Elite_l} - \overrightarrow{R_L}.*\overrightarrow{Gazelle_l}),$$

where, S = Top speed, $\overrightarrow{R_L}$= Random number's vector based on Levy distributions.

How the Predator Chased the Gazelle:

$$\overrightarrow{Gazelle_{l+1}} = \overrightarrow{Gazelle_l} + S.\mu.CF.\vec{R}*.\overrightarrow{R_L}$$
$$.*(\overrightarrow{Elite_l} - \overrightarrow{R_L}.*\overrightarrow{Gazelle_l}),$$

where, $CF = (1-\frac{iter}{Max\_iter})^{(2\frac{iter}{Max\_iter})}$

This stands for the parameter that regulates the predator's movement.

V. Elite update.

VI. Applying PSR effect and update.

$$\overrightarrow{Gazelle_{l+1}} = \begin{cases} \overrightarrow{Gazelle_l} + CF[\vec{lb} + \vec{R}*.(\vec{ub}-\vec{lb})].*\vec{U} & if\ r \le PSRs \\ \overrightarrow{Gazelle_l} + [PSRs(1-r)+r](\overrightarrow{Gazelle_{r1}} - \overrightarrow{Gazelle_{r2}}) & else \end{cases}$$

VII. Update the best solution.

VIII. End GOA.

3.5. Dragonfly Algorithm (DA).

The DA is an optimization algorithm inspired by nature that takes its cues from the way dragonflies hunt. Mirjalili made the initial suggestion for it in 2016. The program mimics the foraging behaviour of dragonflies in pursuit of prey, which helps it tackle optimization challenges. The following are the main tenets and characteristics of DA. Here are the key features and principles of the DA:

I. Initialization.

Initialize a population of Dragonflies:

D = $\{d_1, d_2, \ldots\ldots d_i\}$, where d_i represents the position of the ith Dragonfly in the search space.

Define the search space boundaries: *d_min* and *d_max*.

II. Main Function Evaluation.

Evaluate the fitness of each Dragonfly's position: $f(d_i)$ for (i = 1, 2,……..N.

III. Main Optimization Loop.

When a criterion for termination (such as the maximum number of iterations or a convergence criterion) is not satisfied:

(I) Exploration Phase (Swarming Behaviour). Randomly select a leader Dragonfly from the population as the "group leader".

Update the positions of the other Dragonflies (followers) using the swarming equation, which encourages exploration,

Update the position of the ith follower Dragonfly:

$$d_i = d_i + step_i.(r1.v + r2.(d_{leader} - d_i))$$

*r1 and r*2 are random numbers from [0, 1],

v is random vector,

$c1$ and $c2$ are control parameters,

$step_i$ = step size.

(II). Exploitation Phase (Hunting Behaviour). Update the position of the leader dragonfly based on hunting behaviour to exploit promising areas.

(III). Using an equation derived from dragonfly hunting behaviour, modify the leader's position.

IV. Boundary Check.

Ensure that the dragonfly placements stay inside the designated search space boundaries:

$$d_i = clip(d_i, d_{\max}, d_{\min}).$$

V. Update Best Solution.

After each iteration, update the best solution found so far:

$$d_{best} = \arg\min(f(d_i)).$$

VI. End of DA algorithm.

4. Modelling the Framework and Objective Function for Complex Systems.

This section presents the mathematical formulation and block diagrams of the objective function, that is, a complex system with a mixed configuration. This type of system is neither a series nor a parallel configuration. There are two main objectives, CBS and LSSSC, to verify the efficiency of all five recent metaheuristic algorithms.

4.1. Complex Bridge System (CBS).

Systems with redundant units and non-pure series configurations are challenging to solve. Fig. 1 system (Tillman et al.,1970; Kumar et al., 2023) displays a block representation of the complex bridge.

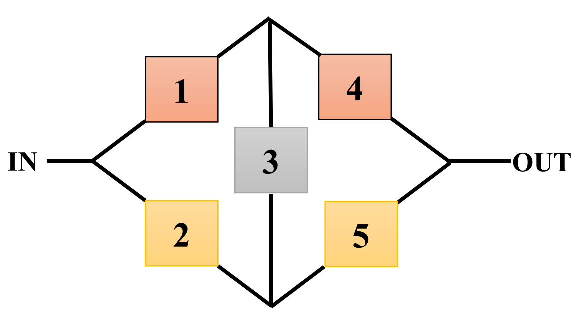

Fig. 1. Block Diagram of CBS.

Source: based on Tillman et al. (1970), Kumar et al. (2023).

This system comprises five parts, each with a component reliability of R_l (l = 1, 2, 3, 4, 5). The CBS is arranged into two subsystems: the first, which consists of components 1 and 4 connected in series, and the second, which consists of components 2 and 5.

Component 3 was placed between these two subsystems, which are joined in parallel. *Rrel* is a nonlinear reliability function of the variable R_l (l = 1, 2, 3, 4, 5).

The reliability of the system (R_{rel}) is the likelihood that the system will succeed, whereas the system cost ($C_{\cos t}$) is the system's cost. The main aim of this problem is to reduce system costs. Here is the mathematical representation of the R_{rel} and $C_{\cos t}$:

$$R_{rel} = R_1R_4 + R_2R_5 + R_2R_3R_4 + R_1R_3R_5$$
$$+2R_1R_2R_3R_4R_5 - R_1R_2R_4R_5 - R_1R_2R_3R_4$$
$$-R_2R_3R_4R_5 - R_1R_2R_3R_5 - R_1R_3R_4R_5.$$

$$C_{\cos t} = \sum_{l=1}^{5} d_l \exp[\frac{b_l}{(1-R_l)}].$$

The following mathematical equations have been used to optimize the overall system cost with nonlinear constraints.

Minimize $C_{\cos t}$, subjected to:

$$0 \le R_l \le 1, \quad l = 1,2,3,4,5$$
$$0.99 \le R_{rel} \le 1,$$
$$d_l = 1, \ and \ b_l = 0.0003 \quad for\ l = 1,2,3,4,5.$$

4.2. The Life Support System in Space Capsule (LSSSC).

To protect astronauts from the harshness of space, a physical habitat for space travel must be developed and studied. LSSSC (Tillman et al.,1970; Kumar et al., 2023) must also be regenerative to satisfy the requirements for human survival. The LSSSC is divided into four parts, each of which has a reliability rating of R_l (l = 1, 2, 3, 4), as shown in Fig. 2 (block diagram).

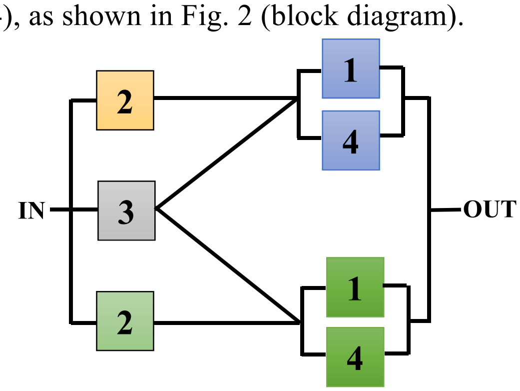

Fig. 2. Block Diagram of LSSSC.

Source: based on Tillman et al. (1970), Kumar et al. (2023).

The system includes two redundant subsystems, each consisting of components 1 and 4, and the system requires only one path to function effectively.

Two identical routes are created in a series-parallel configuration by connecting redundant subsystems in series with component 2. A third path and backup for the two was designated by component 3.

Parallel component 4 serves as the backup of component 1. There are two equivalent paths, each with component 2 following the stages containing parts 1 and 4.

These two parallel, equal paths provide guaranteed results if one is successful.

Let R_{rel} is system's reliability be the probability of the system's success and C_{cost} is the cost of the system. R_{rel} and C_{cost} can be mathematically represented as:

$$R_{rel} = 1 - R_3[(1-R_1).(1-R_4)]^2 -$$
$$(1-R_3)[1-R_2\{1-(1-R_1).(1-R_4)\}]^2.$$
$$C_{\cos t} = 2L_1R_1^{\alpha_1} + 2L_2R_2^{\alpha_2} + L_3R_3^{\alpha_3} + 2L_4R_4^{\alpha_4},$$

Where, L_1=100, L_2=100, L_3=200, L_4=150 and α_l=0.6, l=1, 2, 3, 4.

The mathematical form of the objective function, that is, reducing the overall system cost with nonlinear constraints, is provided as follows:

Minimize $C_{\cos t}$,

subjected to

$$0 \le R_l \le 1, \qquad l = 1,2,3,4,5$$
$$0.99 \le R_{rel} \le 1,$$

where, R_l = Reliability of lth component.

5. Results.

This section discusses and explains the outcomes that have been accomplished.

5.1. Setup for Simulation.

This subsection examines the efficacy of the proposed five algorithms and explains the optimized overall cost and reliability parameters of the CBS and LSSSC systems.

To examine the validity of COA [24], DA [26], MFO [17], GOA [25], and WOA [19]. compared these metaheuristic algorithms, and to prevent repetition, at least 10 different runs were carried out.

On MATLAB Version 2021a, the algorithms have been performed. The following is a list of the settings of the algorithm components for the cost and reliability parameters (Table 1).

Table 1. Value of the Parameters.

Parameter	Settings
No. of iterations	350
Lower boundary limit	0.50
Upper boundary limit	0.99
No. of variables for CBS	5
No. of variables for LSSSC	4
Search agent no	10

5.2. Analysis of the Cost Using the Techniques for Complex Systems.

This subsection provides the convergence behaviour of all comparative methodologies and effectively shows the performance curves.

The CBS system curve (Fig. 3) shows the overall cost analysis using the suggested strategies by varying the number of iterations.

When compared to each of the five proposed methodologies, this evaluation aids in identifying lower-cost convergence.

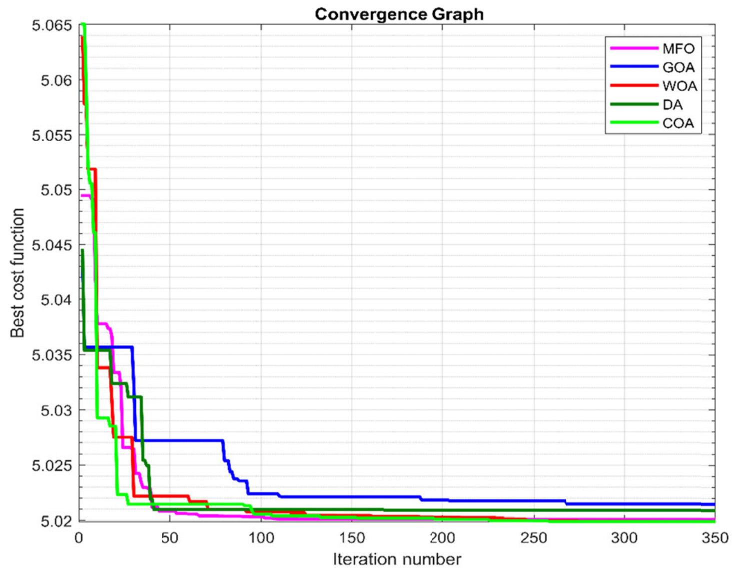

Fig. 3. CBS Convergence Graph for All Approaches (COA, WOA, GOA, MFO, DA).

From the observation of the graph, the authors found that COA is the best algorithm for CBS's cost and reliability parameters as compared to other existing approaches, and also provided a faster solution. The suggested methods provide the most cost-effective solutions for LSSSC, as shown in the convergence graph below (Fig. 4). Analysis of all five approaches (COA, GOA, DA, MFO, and WOA) provided the best solution for the overall system's cost and reliability parameters.

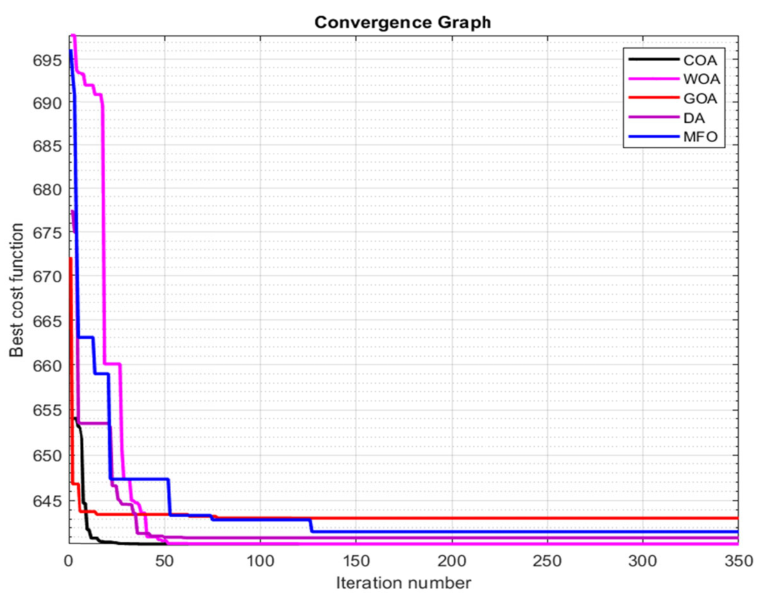

Fig. 4. LSSSC Convergence Graph for All Approaches (COA, WOA, GOA, MFO, DA).

5.3. Performance Analyses of Proposed Methods for Complex Systems.

This subsection evaluates the presentation of all approaches using the Friedman men ranking test and CPU time.

Friedman rank test (Derrac et al., 2011) is an effective non-parametric statistical technique for evaluating differences between several groups or treatments when the data may not match the assumptions of a parametric test.

A key advantage of the Friedman test is its versatility. In addition to the standard statistical analysis, which includes the best, mean, worst, and Standard Deviation (SD), this test was used to assess the significance of the data. This non-parametric test is also employed to rank the algorithms for the complex system costs that have been examined.

The Friedman test's null hypothesis, H_0 (p-value > 5%), denotes that there was no discernible difference between the algorithms under comparison.

The counter-hypothesis H_1, which is true for all 10 runs, indicates a significant variation between the compared five algorithms.

Each algorithm is assigned a rank based on how well it performs in this test. The best algorithms were those that used small ranks.

The outcomes obtained by this test are displayed in Table 2 and the corresponding Fig. 5 for the COA, GOA, WOA, DA, and MFO algorithms. Furthermore, Fig. 6 shows the CPU time of all approaches.

Table 2. Friedman Ranking Test.

Algorithms	Mean Rank	Sum Rank
COA	1.4	1
GOA	3.2	3
MFO	2.9	3
DA	5.1	5
WOA	4	4

A comparison of the greatest outcome found by COA for complex systems using the four approaches is presented in Table 3 and Table 4. The optimal solution is highlighted in the tables below.

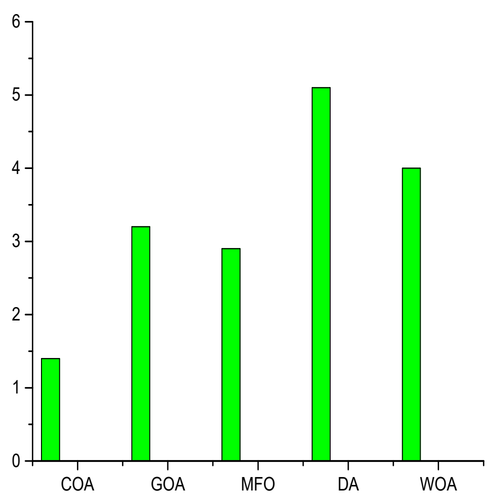

Fig. 5. Friedman Ranking Graph.

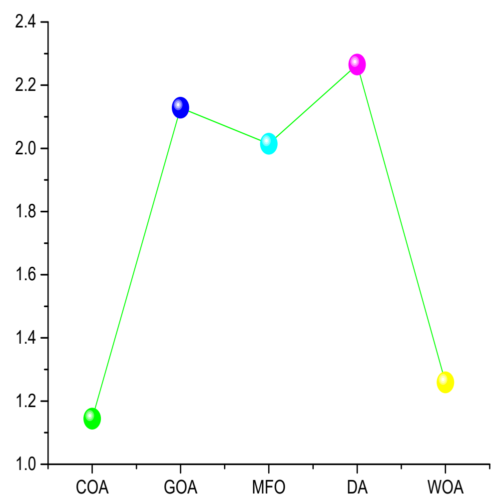

Figure 6. CPU Timing Graph.

Table 3. Comparison of the Best Outcomes of CBS with Different Metaheuristic Algorithms.

Algorithms	C_{cost}	R_{rel}	R_1	R_2	R_3	R_4	R_5
COA	**5.01991833**	0.9900008	0.931853	0.931577	0.93185	0.93048	0.9323445
GOA	5.02146954	0.9900000	0.933816	0.932887	0.782265	0.937000	0.936507
MFO	5.02010132	0.9900010	0.935625	0.935625	0.791589	0.936582	0.935621
DA	5.02262918	0.9900000	0.934080	0.923082	0.833828	0.926066	0.934562
WOA	5.02933931	0.99000005	0.922895	0.919269	0.563333	0.981652	0.933852

Table 4. Comparison of the Best Outcomes of LSSSC with Different Metaheuristic Algorithms.

Algorithms	C_{cost}	R_{rel}	R_1	R_2	R_3	R_4
COA	**641.7895895**	0.990000	0.784045	0.995263	0.558563	0.952548
GOA	643.8673604	0.990000	0.556987	0.556982	0.806395	0.992586
MFO	641.8125865	0.990000	0.551235	0.902062	0.550458	0.901523
DA	641.7925632	0.990000	0.556932	0.995862	0.552526	0.787042
WOA	641.8532132	0.990000	0.788254	0.988752	0.559632	0.708052

5.4. Statistical Analysis of Algorithms.

The maximum, minimum, mean, and standard deviation of five algorithms are listed in Table 5 and Table 6 for both complexes.

According to the statistical data, COA was found to be the most appropriate and effective parameter optimization strategy.

Table 5. Statistical Outcomes of All Proposed Approaches for CBS.

Algorithm	Minimum (Best)	Maximum (Worst)	Mean	SD
COA	5.01991833	5.02925865	5.02010135	0.000301452
WOA	5.02146954	5.03025896	5.02022070	0.000254563
DA	5.02010132	5.02814562	5.02212315	0.004525896
MFO	5.02256291	5.03012545	5.02123524	0.000350586
GOA	5.02933931	5.02325156	5.02214562	0.000256348

Table 6. Statistical Outcomes of All Proposed Approaches for LSSSC.

Algorithm	Minimum (Best)	Maximum (Worst)	Mean	SD
COA	641.7895895	671.611779	644.256586	1.562548
WOA	643.8673604	677.341961	648.254548	4.056893
DA	641.8125865	672.109578	647.586226	2.569874
MFO	641.7925632	671.625265	650.596352	1.895426
GOA	641.8532132	654.258963	648.895146	3.256816

6. Conclusions.

The objective of this study is to present five novel approaches for reducing the overall complex system cost. The author considers two systems: CBS and LSSSC, for example, a complex system. Lastly, the model's performance was evaluated and contrasted with each suggested optimization technique. In addition, the authors analysed the comparative solution and observed that the COA provided the best minimum solution for the overall complex system as compared to the other four (GOA, DA, MFO, and WOA) approaches. The statistical results and Friedman ranking test demonstrated that COA has the first rank and minimum SD, which indicates that COA is the best algorithm.

In summary, this study suggests novel strategies for reducing the cost of complex systems.

The effectiveness of the COA as an algorithm for cost and reliability parameter optimization was demonstrated. The results of this study show that the suggested COA procedure is more successful than the other procedures described in this article. The following are ideas for future research:

• Develop more models that can be used in real time.

• Extending the model to multi-objective metaheuristic approaches.

• Take a hybrid metaheuristic and heuristic method for more complex systems.

REFERENCES

Abd Alsharify, F. H., & Hassan, Z. A. H. (2022). Optimization of complex system reliability: Bat algorithm-based approach. International Journal of Health Sciences, 6, S1. https://doi.org/10.53730/ijhs.v6nS1.8637

Abdullah, G., & Hassan, Z. A. H. (2020). Using of particle swarm optimization (PSO) to addressed reliability allocation of complex network. Journal of Physics: Conference Series, 1664 (1), 012125. https://doi.org/10.1088/1742-6596/1664/1/012125

Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2022). Dwarf mongoose optimization algorithm. Computer Methods in Applied Mechanics and Engineering, 391, 114570. https://doi.org/10.1016/j.cma.2022.114570

Agushaka, J. O., Ezugwu, A. E., & Abualigah, L. (2023). Gazelle optimization algorithm: a novel nature-inspired metaheuristic optimizer. Neural Computing and Applications, 35(5), 4099-4131. https://doi.org/10.1007/s00521-022-07854-6

Beji, N., Jarboui, B., Eddaly, M., & Chabchoub, H. (2010). A hybrid particle swarm optimization algorithm for the redundancy allocation problem. Journal of Computational Science, 1(3), 159-167. https://doi.org/10.1016/j.jocs.2010.06.001

Bhunia, A. K., Duary, A., & 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, 2(3), 185-212. https://doi.org/10.33889/ijmems.2017.2.3-016

Choudhary, S., Ram, M., Goyal, N., & Saini, S. (2023). Reliability and cost optimization of series–parallel system with metaheuristic algorithm. International Journal of System Assurance Engineering and Management, 1-11. https://doi.org/10.1007/s13198-023-01905-4

Coit, D. W., & Zio, E. (2019). The evolution of system reliability optimization. Reliability Engineering & System Safety, 192, 106259. https://doi.org/10.1016/j.ress.2018.09.008

Dahiya, B. P., Rani, S., & Singh, P. (2019). A hybrid artificial 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, 4(2), 471. https://doi.org/10.33889/IJMEMS.2019.4.2-039

Dao, T. K., Pan, T. S., & Pan, J. S. (2016, November). A multi-objective optimal mobile robot path planning based on whale optimization algorithm. In 2016 IEEE 13th international conference on signal processing (ICSP) (pp. 337-342). IEEE. https://doi.org/10.1109/ICSP.2016.7877851

Dehghani, M., Hubálovský, Š., & Trojovský, P. (2021). Cat and mouse-based optimizer: A new nature-inspired optimization algorithm. Sensors, 21(15), 5214. https://doi.org/10.3390/s21155214

Dehghani, M., Montazeri, Z., Trojovská, E., & Trojovský, P. (2023). Coati Optimization Algorithm: A new bio-inspired metaheuristic algorithm for solving optimization problems. Knowledge-Based Systems, 259, 110011. https://doi.org/10.1016/j.knosys.2022.110011

Derrac, J., García, S., Molina, D., & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm and Evolutionary Computation, 1(1), 3-18. https://doi.org/10.1016/j.swevo.2011.02.002

Durgadevi, A., & Shanmugavadivoo, N. (2023). Availability Capacity Evaluation and Reliability Assessment of Integrated Systems Using Metaheuristic Algorithm. Computer Systems Science & Engineering, 44(3). https://doi.org/10.32604/csse.2023.026810

Gandhi, B. R., & 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, 16, 105-117. https://doi.org/10.1007/978-3-030-26458-1_7

Garg, H. (2021). Bi-objective reliability-cost interactive optimization model for series-parallel system. International Journal of Mathematical, Engineering and Management Sciences, 6(5), 1331.https://doi.org/10.33889/IJMEMS.2021.6.5.080

Hwang, C. L., Tillman, F. A., & Lee, M. H. (1981). System-Reliability Evaluation Techniques for Complex/Large Systems —A Review. IEEE Transactions on Reliability, 30(5), 416-423. https://doi.org/10.1109/TR.1981.5221152

Kaveh, A., Talatahari, S., & Khodadadi, N. (2020). Stochastic paint optimizer: theory and application in civil engineering. Engineering with Computers, 1-32. https://doi.org/10.1007/s00366-020-01179-5

Khodadadi, N., Azizi, M., Talatahari, S., & Sareh, P. (2021). Multi-objective crystal structure algorithm (MOCryStAl): Introduction and performance evaluation. IEEE Access, 9, 117795-117812. https://doi.org/10.1109/ACCESS.2021.3106487

Khorshidi, H. A., & Nikfalazar, S. (2015). Comparing two meta-heuristic approaches for solving complex system reliability optimization. Applied and Computational Mathematics, 4(2-1), 1-6. https://doi.org/10.11648/j.acm.s.2015040201.11

Kumar, A., Pant, S., & Singh, S. B. (2017). Reliability optimization of complex systems using a cuckoo search algorithm. In Mathematical Concepts and Applications in Mechanical Engineering and Mechatronics (pp. 94-110). IGI global. https://doi.org/10.4018/978-1-5225-1639-2

Kumar, A., Pant, S., Singh, M. K., Chaube, S., Ram, M., & Kumar, A. (2023). Modified Wild Horse Optimizer for Constrained System Reliability Optimization. Axioms, 12(7), 693. https://doi.org/10.3390/axioms12070693

Mahdavi-Nasab, N., Abouei Ardakan, M., & Mohammadi, M. (2020). Water cycle algorithm for solving the reliability-redundancy allocation problem with a choice of redundancy strategies. Communications in Statistics-Theory and Methods, 49(11), 2728-2748. https://doi.org/10.1080/03610926.2019.1580741

Mettas, A. (2000, January). Reliability allocation and optimization for complex systems. In Annual reliability and maintainability symposium. 2000 Proceedings. International symposium on product quality and integrity (Cat. No. 00CH37055) (pp. 216-221). IEEE. https://doi.org/10.1109/RAMS.2000.816310

Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems, 89, 228-249. https://doi.org/10.1016/j.knosys.2015.07.006

Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural computing and applications, 27, 1053-1073. https://doi.org/10.1007/s00521-015-1920-1

Mirjalili, S., & Lewis, A. (2016). The whale optimization algorithm. Advances in engineering software, 95, 51-67. https://doi.org/10.1016/j.advengsoft.2016.01.008

Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey wolf optimizer. Advances in Engineering Software, 69, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Nath, R., & 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 & System Safety, 244, 109835. https://doi.org/10.1016/j.ress.2023.109835

Negi, G., Kumar, A., Pant, S., & Ram, M. (2021). Optimization of complex system reliability using hybrid grey wolf optimizer. Decision Making: Applications in Management and Engineering, 4(2), 241-256. https://doi.org/10.31181/dmame210402241n

Pahuja, G. L. (2020). Solving reliability redundancy allocation problem using grey wolf optimization algorithm. Journal of Physics: Conference Series, 1706(1), 012155. https://doi.org/10.1088/1742-6596/1706/1/012155

Ravi, V. (2004). Optimization of complex system reliability by a modified great deluge algorithm. Asia-Pacific Journal of Operational Research, 21(04), 487-497. https://doi.org/10.1109/rams.2000.816310

Rocco, C. M., Miller, A. J., Moreno, J. A., & Carrasquero, N. (2000, January). 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 (pp. 210-215). IEEE. https://doi.org/10.1109/RAMS.2000.816309

Sahoo, S. K., Saha, A. K., Ezugwu, A. E., Agushaka, J. O., Abuhaija, B., Alsoud, A. R., & Abualigah, L. (2023). Moth flame optimization: theory, modifications, hybridizations, and applications. Archives of Computational Methods in Engineering, 30(1), 391-426. https://doi.org/10.1007/s11831-022-09801-z

Tillman, F. A., Hwang, C. L., Fan, L. T., & Lai, K. C. (1970). Optimal reliability of a complex system. IEEE Transactions on Reliability, 19(3), 95-100. https://doi.org/10.1109/TR.1970.5216413

Umamaheswari, E., Ganesan, S., Abirami, M., & Subramanian, S. (2018). Reliability/risk centered cost-effective preventive maintenance planning of generating units. International Journal of Quality & Reliability Management, 35(9), 2052-2079. https://doi.org/10.1108/IJQRM-03-2017-0039

Wang, F., Araújo, D. F., & Li, Y. F. (2023). Reliability assessment of autonomous vehicles based on the safety control structure. Journal of Risk and Reliability, 237(2), 389-404. https://doi.org/10.1177/1748006X211069705

Wei, Y., & Liu, S. (2023). Reliability analysis of series and parallel systems with heterogeneous components under random shock environment. Computers & Industrial Engineering, 179, 109214. https://doi.org/10.1016/j.cie.2023.109214

Zhang, J., Lv, H., & 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 & System Safety, 229, 108843.https://doi.org/10.1016/j.ress.2022.108843