axioms

MDPI

Article

Modified Wild Horse Optimizer for Constrained System Reliability Optimization

Anuj Kumar [1], Sangeeta Pant [2,*], Manoj K. Singh [3], Shshank Chaube [1], Mangey Ram [4,*] and Akshay Kumar [5]

[1] Department of Mathematics, University of Petroleum and Energy Studies, Dehradun 248007, India; anuj4march@gmail.com (A.K.); chaube.shshank@gmail.com (S.C.)
[2] School of Engineering and Computing, Dev Bhoomi Uttarakhand University, Dehradun 248007, India
[3] School of Computer Science, University of Petroleum and Energy Studies, Dehradun 248007, India; manojkumar.erg@gmail.com
[4] Department of Mathematics, Computer Science and Engineering, Graphic Era Deemed to be University, Dehradun 248002, India
[5] Department of Mathematics, Graphic Era Hill University, Dehradun 248002, India; akshaykr1001@gmail.com
* Correspondence: pant.sangeet@gmail.com (S.P.); mangeyram@geu.ac.in (M.R.);
Tel.: +91-135-2642727 or +91-135-2642729 (ext. 256) (M.R.)

Abstract: The last few decades have witnessed advancements in intelligent metaheuristic approaches and system reliability optimization. The huge progress in metaheuristic approaches can be viewed as the main motivator behind further refinement in the system reliability optimization process. Researchers have intensively studied system reliability optimization problems (SROPs) to obtain the optimal system design with several constraints in order to optimize the overall system reliability. This article proposes a modified wild horse optimizer (MWHO) for SROPs and investigates the reliability allocation of two complex SROPs, namely, complex bridge system (CBS) and life support system in space capsule (LSSSC), with the help of the same process. The effectiveness of this framework based on MWHO is demonstrated by comparing the results obtained with the results available in the literature. The proposed MWHO algorithm shows better efficiency, as it provides superior solutions to SROPs.

Keywords: reliability optimization; metaheuristics; modified wild horse optimizer; system cost; system reliability

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

Academic Editor: Valery Y. Glizer

Received: 17 April 2023
Revised: 2 July 2023
Accepted: 4 July 2023
Published: 16 July 2023

1. Introduction

The crucial role of reliability optimization in 21st century industry is the reason for the extensive involvement of various researchers, industry experts and decision makers (DM) in it. All stakeholders, ranging from automobile industries, transportation systems, and the military to food industries, have some stake in this concept's success, as the combination of reliability and the associated cost of their products has a significant influence on customer satisfaction. Thus, to remain competitive in today's world, the basic goal of associated reliability engineers is to improve the overall reliability of the product and its components, as well as maintaining production of the product at a competitive cost [1]. Reliability can be viewed as the probability that a system works uninterrupted for a specific period of time. Reliability is also defined as the probability that a product, system, or service will perform its intended function adequately for a specified period of time, or at least operate in a defined environment without failure [2–4]. This definition encourages system engineers to develop a reliable and cost-effective product, which, in turn, increases the complexities and creates a complex system design process and, hence, a complex SROP. Generally, SROPs can be distinguished into three classes, namely, reliability allocation problems (ReAP), redundancy allocation problems (RAP) and reliability–redundancy allocation problems

(RRAP) [5]. In all of these approaches, researchers aim to achieve optimal system reliability and cost under certain constraints based on available resources.

The computational complexity associated with SROPs has prompted researchers to solve these problems using the various approaches available in the literature. These approaches can be broadly categorized into two categories, namely, heuristic approaches and metaheuristic approaches [6]. Metaheuristic approaches have an edge over heuristics, as they have a derivative-free mechanism and are quite simple and flexible, even when dealing with highly complex non-linear optimization problems like SROPs. Metaheuristics also have a superior ability to avoid local extrema. SROPs have been proven to be NP-hard in nature, and their computational complexity increases dramatically as the scale of system configuration increases [7,8]. According to the "No Free Launch (NFL)" theorem [9], there exist no metaheuristics available that can solve all optimization problems with the same efficiency. Alternatively, it may be true that, while a particular algorithm offers better solutions to some optimization problems, it may not offer better solutions for some other problems, hence its failure to resolve them. Thus, no metaheuristic approach is perfect. Therefore, NFL-based motivation provokes researchers to develop new algorithms or upgrade some original metaheuristics to solve a wider range of complex problems like SROPs.

Recently, works based on various recent metaheuristic approaches have been presented by many authors, as is reported in Table 1, which discusses approaches used specifically for SROPs.

Table 1. Literature review of the studies that have used metaheuristic approaches for SROPs.

Literature Reviewed	SROP Type	Optimization Technique Used
Modibbo et al. [10]	ReAP	Simulation algorithm (SA)
Ouzineb et al. [11]	RAP	Tabu search (TS)
Beji et al. [12]	RAP	Hybrid particle swarm optimizer (HPSO)
Kumar et al. [13]	ReAP	Gray wolf optimizer (GWO)
Hsieh and You [14]	RRAP	Artificial immune search algorithm (AISA)
Wu et al. [15]	RRAP	Improved PSO (IPSO)
Zou et al. [16]	RRAP	PSO and harmony search algorithm (HSA)
Hsieh and Yeah [17]	RAP	Bee colony algorithm (BCA)
Lins and Droguett [18]	RAP	Genetic algorithm (GA)
Wang and Li [19]	RRAP	HAS
Wang and Li [20]	RAP	HPSO and local search algorithm (LSA)
Pourdarvish and Ramezani [21]	RAP	Memetic algorithm (MA)
Valian and Valian [22]	RRAP	Cuckoo search algorithm (CSA)
Valian et al. [23]	RAP	Improved CSA (ICSA)
Afonso et al. [24]	RRAP	Modified imperialist competitive algorithm (MICA)
Ardakan and Hamadani [25]	RAP	GA
Yeh [26]	RAP	Orthogonal simplified swarm optimization (OSSO)
Kumar et al. [27]	ReAP	CSA and GWO
Huang [28]	RRAP	Particle-based SSO

Table 1. *Cont.*

Literature Reviewed	SROP Type	Optimization Technique Used
He et al. [29]	RRAP	Novel artificial fish swarm optimization (NAFSO)
Ardakan et al. [30]	RRAP	Non-sorting GA II (NSGA II)
Zhuang and Li [31]	RAP	Stochastic order technique (SOT)
Garg [32]	RRAP	CSA
Mellal and Zio [33]	RAP	Penalty-guided stochastic fractal search algorithm (PSFSA)
Abouei et al. [34]	RRAP	Modified GA (MGA)
Gholinezhad and Hamadani [35]	RAP	GA
Kim & Kim [36]	RAP	Continuous-time Markov chain technique (CMCT)
Kim [37]	RAP	Matrix-analytic technique (MAT)
Garg [38]	ReAP	GSA (gravitational search algorithm)- GA
Al-Azzoni and Iqbal [39]	ReAP	GA and ACO
Kumar et al. [40]	ReAP	GWO
Negi et al. [41]	ReAP	Hybrid GWO (HGWO)

Hence, based on the discussion, it can be concluded that SROPs are a hot topic for researchers and scientists due to their association with high computational complexity. In this article, the authors propose a framework based on the modified version of very recent metaheuristic named wild horse optimizer, which mimics the social and herding behaviour of wild horses in their natural habitat, for the solution of SROPs.

The rest of the article is organized as follows: Section 2 describes the modified wild horse optimization (MWHO) algorithm. Section 3 elaborates on the mathematical formulation of SROPs i.e., formulation of complex bridge system (CBS) and life support system in space capsule (LSSSC). Section 4 illustrates the results obtained by the MWHO algorithm for SROPs discussed in Section 3. Finally, the conclusions and future scope are drawn in Section 5.

2. Modified Wild Horse Optimization (MWHO) Algorithm

Naruei and Keynia [42] developed a wild horse optimizer (WHO) which mathematically mimics the social life behaviour of wild horses and is able to effectively handle various recently developed complex test problems like CEC2017 and CEC 2019, on which several metaheuristics perform poorly. Wild horses live in a groups and follow their leader, the stallion horse. Foals and mares follow the stallion in their day-to-day activities like grazing, breeding, pursuing, etc. We modified the original WHO for SROPs application (Figure 1). Here, the major difference is in the optimizing reliability parameters in the form of vectors and matrices. We discuss here, the major steps associated with MWHO for SROPs.

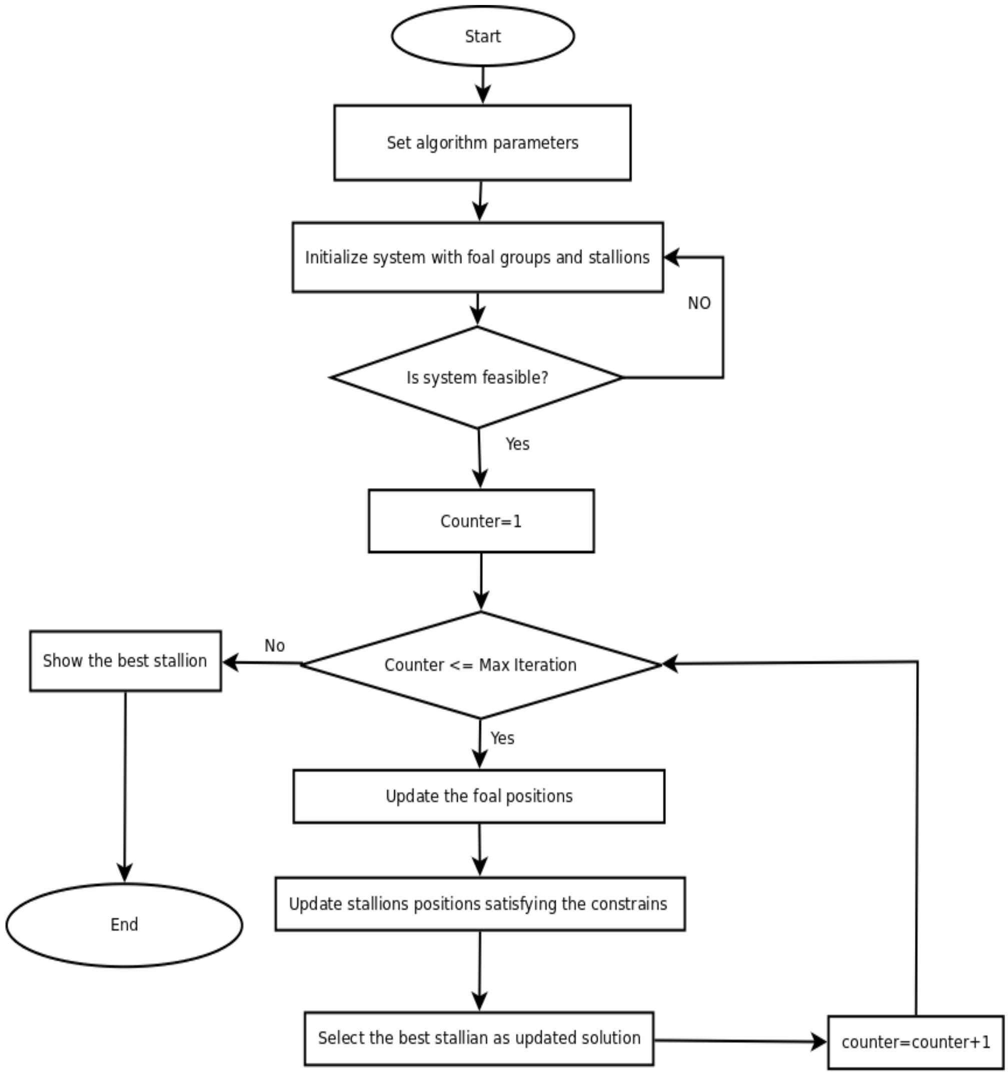

Figure 1. Flow chart of MWHO.

2.1. Initialization, Group Construction and Stallion Selection

Initially, the population of horses is divided into several groups. Let M denote the set of horses in the population. H is the number of subsets with each subset representing a group. The algorithm assigns a leader, i.e., a stallion, to every group. Hence, there are H stallions in the algorithm. The remaining $(M - H)$ population consists of foals and mares are further distributed among these H groups. Each stallion, foal and mare represent a matrix of size $p \times q$, where p is an upper bound in the number of the components in each of the q subsystems. The elements of the matrix represent the reliability of the components in each of the subsystems.

2.2. Grazing Behaviour of a Wild Horse

The majority of time is spent by foals and mares grazing in their group with a stallion at the centre of the grazing region [42]. Equation (1) simulates this grazing behaviour of wild horses.

$$\overline{Y}^{q}_{p,H} = 2Z\cos\left(2\pi RAN \times Z\right) \times \left(Stallion^{q} - Y^{q}_{p,H}\right) + Stallion^{q} \tag{1}$$

where $\overline{Y}^{q}_{p,H}$ denotes the current position of the qth subset member in the pth subset. $Stallion^{q}$ is the state variable associated with the stallion in subset q. Z is computed using Equation (2), which represents the adaptive nature of wild horses. RAN represents a random number following uniform distribution in the range $[-2, 2]$. Finally, $\overline{Y}^{q}_{p,H}$ gives the updated values of the state variables associated with the subset member during grazing. The term $2Z\cos(2\pi RAN \times Z) \times \left(Stallion^{q} - Y^{q}_{p,H}\right)$ in Equation (1) is a scaling of the vector, $\left(Stallion^{q} - Y^{q}_{p,H}\right)$, which determines the direction distance between the stallion and the member. Adding $Stallion^{q}$ to this term i.e., the right side of the equation, represents the repositioning of the member along the vector joining the position of the stallion and the member. In other words, Equation (1) provides the new position of the member depending on the sign of the cosine term away from or towards the stallion, and thus the equation also determines the force between the stallion and the member. If the force is positive, the stallion pulls the member towards itself; if not, it repels the member away. The variables in Equation (1) are matrices of size $p \times q$ and the reliability of each of the components in the subsystem are updated according to Equation (1).

$$Q = \vec{S}_1 < TDR; IDX = (Q == 0); Z = S_2 \Theta IDX + \vec{S}_3 \Theta (\backsim IDX) \tag{2}$$

Here, Q denotes a matrix with elements either 0 or 1. The elements in the matrices $\vec{S}_1$ and $\vec{S}_3$ follow uniform distribution in the range $[0, 1]$. S_2 is, again, a random number following uniform distribution in the range $[0, 1]$. The indices of the random matrix $\vec{S}_1$ masked with IDX following the positions where Q is zero in Equation (2) (The operator == in the equation is a logical operator aligned with the == operator in MATLAB or Python programming language). The operator $\backsim$; represents the negation i.e., if x = 0 (FALSE) then ~x will return 1 (TRUE). The operator Θ represents elementwise multiplication; in the expression $S_2\Theta IDX$, both S_2 and IDX are numbers; therefore, the elementwise multiplication reduces to normal multiplication between two numbers. In the expression $Z = S_2 \Theta IDX + \vec{S}_3 \Theta (\backsim IDX)$, the operands of the operator + are a number on the left and a matrix on the right. The number on the left operand transforms itself into a matrix of the same size as the right operand, and all the elements of the matrix as the left operand. TDR represents a dynamic parameter lying between [0, 1], which updates itself in accordance with Equation (3) during the execution process of the algorithm.

$$TDR = 1 - iter\left(\frac{1}{iter_{max}}\right) \tag{3}$$

where iter denotes an iteration counter with $iter_{max}$ as the upper bound on the number of iterations. As discussed earlier, in the expression $\vec{S}_1 < TDR$, the right operand (TDR) transforms itself into a matrix of the same size as $\vec{S}_1$ and the operator $<$ acts as elementwise comparison operator. The position of the member is updated by using Equation (1), which is modulated by Equation (2) in conjunction with Equation (3). The value of TDR reduces as the iteration progresses, which leads to a reduction in the number of 1s in Q (Equation (2)), which further leads to a increase in the number of 1s in IDX. The increase in 1s in IDX bounds the variations among elements of Z, leading all the members to move at the same scale towards $iter_{max}$ iterations. Therefore, the optimization algorithm does not behave abnormally and does not deviate from convergence in the last few iterations.

2.3. Breeding Behaviour of a Horse

Wild horses have developed a mechanism so that fathers cannot mate with their daughters or siblings, as the foals split from the group before reaching puberty. Equation (4) represents this behavior depending upon a Crossover operator [42].

$$Y^{i}_{G_k} = Crossover\left(Y^{j}_{G_p}, Y^{Z}_{G_q}\right); k \neq p \neq q, i = j = end, Crossover = Mea \tag{4}$$

where $Y^{i}_{G_k}$ denotes the state space of the ith horse in the kth group who splits from the subset and leaves its place for a horse with parents who parted from groups p and q in addition to having reached puberty. $Y^{i}_{G_p}$ is the state space of the jth foal from the subset p, which leaves the group after reaching the puberty, mates with the horse Z with the state space $Y^{Z}_{G_q}$, which then splits from group q. In Equation (4), the state of the vacated ith horse in the kth group is updated with the mean value of the position of the jth foal from the subset p and horse Z from subset q. In addition, Equation (4) also suggest that the crossover takes place between elements from two distinct groups.

2.4. Leadership Behaviour

The leadership and competitive behaviour of stallions when moving towards water sources is represented by Equation (5) [42].

$$\bar{S}_{G_p} = \begin{cases} 2Z\cos(2\pi RAN \times Z) \times \left(WH - S_{G_p}\right) + WH, & \text{if } S_3 > 0.5 \quad (5a) \\ 2Z\cos(2\pi RAN \times Z) \times \left(WH - S_{G_p}\right) - WH, & \text{if } S_3 \leq 0.5 \quad (5b) \end{cases}$$

where $\bar{S}_{G_p}$, WH and S_{G_p} denote the next position in state space of the leader in the pth group, the state of the water hole and the current state of the leading element of the pth subset, respectively. Equation (5a,b) behave in a similar way to Equation (2); however, here, the role of the stallion is taken by the global optimal and the role of the members is swapped for the stallions. Therefore, the global optimal position searched by the algorithm does not change abruptly from the convergence.

2.5. Leader Selection and Exchange

Finally, the leader is chosen based on their fitness or cost (Equation (6)) [42]. The leader's position, along with the relevant member, will be modified by using Equation (6).

$$S_{G_p} = \begin{cases} Y_{G_p}, \text{if} \cos t\left(Y_{G_p}\right) < \cos t\left(S_{G_p}\right) \\ S_{G_p}, \text{if} \cos t\left(Y_{G_p}\right) > \cos t\left(S_{G_p}\right) \end{cases} \quad (6)$$

The original WHO algorithm [42] can be applied across a wide range of optimization problems. However, to apply the WHO for SROPs, certain modifications must be made. These include the change of one variable $Y^{q}_{p,G}$ to matrix $Y^{q}_{p,G}$ of size $p \times q$. We discuss here two classes of modifications, which are further required to use the proposed MWHO efficiently for SROPs. The first class of modification pertains to the application of constraints, which is not considered in the original WHO algorithm [42]. However, as the SROPs in question involve constraints (i.e., the constraints of overall system reliability ($R_{Sys.}$) and components reliabilities (R_k)), conditional steps are introduced to the WHO algorithm to accommodate these constraints. During the update process, these conditions are assessed against the constraints, and only if the constraints are met, the new positions of the horses are updated. The second class of modifications relates to the initialization process in MWHO algorithms.

While the horse population is initialized with random values, these values may not adhere to the constraints, causing the horses to diverge or converge at a slow rate. To address this, each horse is compelled to start in the feasible region by iterating over multiple generations until a feasible solution is attained. This step is repeated for each member of the population. Figure 2 provides the pseudo-code for MWHO.

```
1. Initialize algorithm parameters
2. Initialize the Horse population
3. Evaluate the fitness of the population
4. If a horse does not satisfy the constraints,
   replace it with another horse which satisfies the constraints
5. Form foal groups and a stallions
6. Find the best horse
7. While (iteration <= Maxiter)
8. Calculate TDR
9. for each stallion
10.  for each foal in a group
11.   If (rand>PC)
12.  Compute the position of a foal using equation 1
13.  end
14.  else
15.  Compute the position of a foal using equation 4
16.  end
17.  if new position satisfies the constraints
18.  Update the position of the foal
19.  end
20.  if rand > 0.5
21. Compute the position of the stallion using equation 5a
22. end
23. else
24. Compute the position of the stallion using equation 5b
25. if the new position satisfies the constraints
26. update the position of the stallion
27. end
28.  If the cost of the updated stallion is less than the old stallion
29. Replace the stallion with the updated stallion
30. end
31. Find foal with optimal cost
32. if the cost of the selected foal is lower than the cost of the stallion
33. Exchange foal and stallion using equation 6
34. end
35. end
36. Update optimal position of objective function
37. end
```

Figure 2. Pseudo-code of the MWHO.

3. Problem Description

To check the applicability and efficiency of the MWHO, two SROPs and one engineering optimization problem (EOP) are considered here. The block diagram of the SROPs and EOP considered is depicted in Figures 3–5.

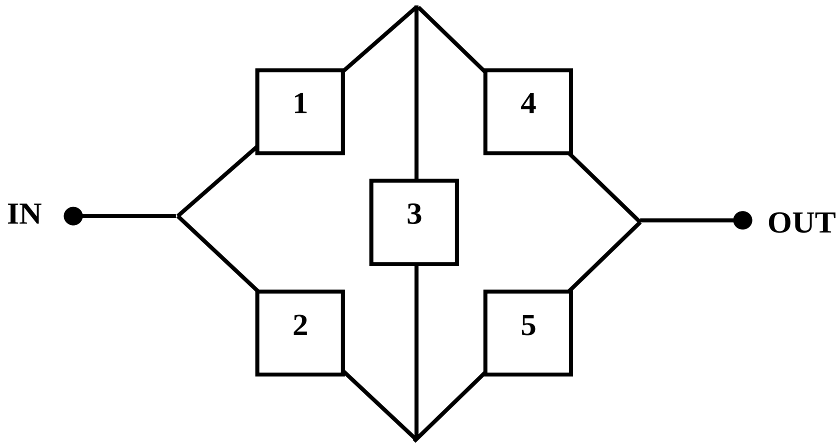

Figure 3. Block diagram of SROP 1.

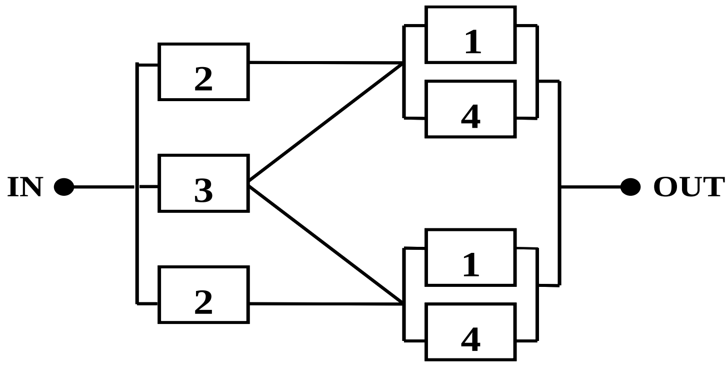

Figure 4. Block diagram of SROP 2.

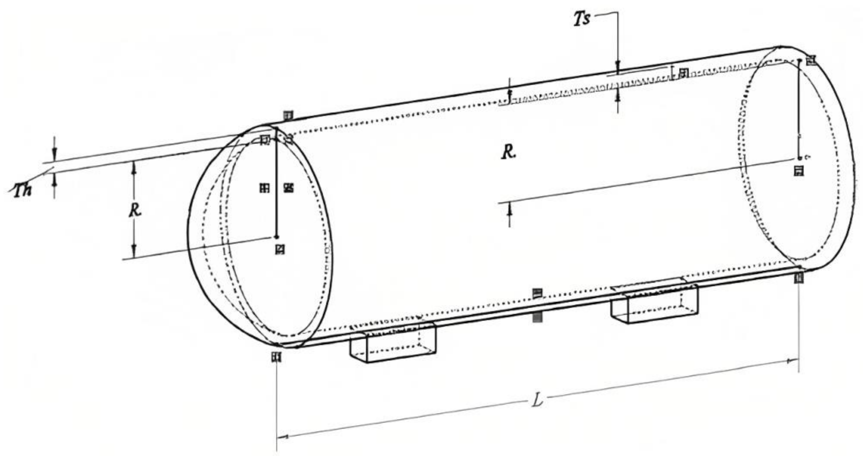

Figure 5. Schematic of PVD.

3.1. Statement of the Optimization Problems

A general constraint optimization problem, which is non-linear in nature, can be defined as

$$\text{Optimize } g\left(\vec{x}\right), \qquad \vec{x} = (x_1,\ x_2,\ \ldots,\ x_n) \in \mathbb{R}^n$$

$$\text{Where } f_j\left(\vec{x}\right) \leq 0 \quad for\ j = 1,\ 2,\ 3, \ldots,\ p\ \ and$$

$$h_j\left(\vec{x}\right) = 0 \quad for\ j = p+1, \ldots,\ m$$

A typical implementation consists of three fundamental elements: a collection of variables, a fitness function for optimization (either maximizing or minimizing), and a set of constraints that define the permissible range of values for the variables. The objective is to determine the optimal values for the variables that optimize the fitness (cost) function while adhering to the given constraints.

In SROP1 and SROP2, the objective is to minimize the associated system cost ($C_{Sys.}$), subject to the constraints of overall system reliability ($R_{Sys.}$) and component reliabilities (R_k). In EOP, the objective again is to minimize the total cost ($C_{Sys.}$) consisting of material, forming and welding of a cylindrical vessel subjected to various constraints formed by the decision variables, namely, thickness of the shell (T_S), thickness of the head (T_h), inner radius (R), length of the cylindrical section without considering the head (L).

3.2. SROP 1: Complex Bridge System (CBS)

Solving a system that has a redundant unit and is not in a pure series configuration poses significant difficulty. The complex bridge system problem, depicted in Figure 3, is a prime example. This system consists of five components, each possessing a component reliability value of $R_k(k = 1......5)$. The complex bridge system is composed of two subsystems: the first subsystem involves components 1 and 4 connected in series, while the second subsystem comprises components 2 and 5 in series. These two subsystems are connected in a parallel configuration, with component 3 inserted in between. Figure 3 depicts the block diagram of CBS.

$$\begin{aligned} R_{Sys.} = {} & R_1R_4 + R_2R_5 + R_2R_3R_4 + R_1R_3R_5 + 2R_1R_2R_3R_4R_5 - R_1R_2R_4R_5 \\ & - R_1R_2R_3R_4 - R_2R_3R_4R_5 - R_1R_2R_3R_5 - R_1R_3R_4R_5 \end{aligned} \tag{7}$$

$$C_{Sys.} = \sum_{k=1}^{5} c_k \exp\left[\frac{d_k}{(1-R_k)}\right] \tag{8}$$

The mathematical formulation of SROP 1, with the objective of minimizing overall system cost with nonlinear constraint, is given below [13]:

$$\text{Minimize} C_{Sys.}$$

subjected to

$$0 \leq R_k \leq 1, k = 1,\ 2,\ 3,\ 4,\ 5$$

$$0.99 \leq R_{sys.} \leq 1,$$

$$c_k = 1 \text{ and } d_k = 0.0003,\ \text{for } k = 1,\ 2,\ 3,\ 4,\ 5$$

3.3. SROP 2: Life Support System in Space Capsule (LSSSC)

The creation and analysis of physical habitat for space exploration is crucial to shield the astronaut from the harshness of space. Additionally, the LSSSC must be regenerative, providing essential elements for human survival. Figure 4 illustrates the block diagram of the LSSSC, which consists of four components, each with a reliability value of $R_k(k = 1......4)$. The system requires a single path for successful operation and contains two redundant subsystems, with each subsystem comprising components 1 and 4. Both redundant subsystems are in a series with component 2, forming two identical paths in a series-parallel arrangement. Component 3 serves as a third path and backup for the pair. Component 1 is backed up by a parallel component 4, and two identical paths are created, each having component 2 in a series with the stage consisting of components 1 and

4. These two paths operate in parallel, so that if one of them functions correctly, the output is guaranteed.

$$R_{Sys.} = 1 - R_3[(1 - R_1)(1 - R_4)]^2 - (1 - R_3)[1 - R_2\{1 - (1 - R_1)(1 - R_4)\}]^2 \tag{9}$$

$$C_{Sys.} = 2I_1R_1^{\alpha_1} + 2I_2R_2^{\alpha_2} + I_3R_3^{\alpha_3} + 2I_4R_4^{\alpha_4} \tag{10}$$

where, $I_1 = 100$, $I_2 = 100$, $I_3 = 200$, $I_4 = 150$ and $\alpha_k = 0.6, k = 1, 2, 3, 4$.

The mathematical formulation of SROP 2, with the objective of minimizing overall system cost with nonlinear constraint, is given below [13]:

$$\text{Minimize} C_{Sys.}$$

subjected to

$$0.5 \le R_k \le 1 \qquad k = 1, 2, 3, 4$$

$$0.9 \le R_{Sys.} \le 1$$

where R_k is the kth component's reliability.

3.4. Engineering Optimization Problem (EOP): Pressure Vessel Design (PVD)

The objective of this EOP, named PVD, is to minimize the total cost ($C_{Sys.}$) consisting of material, forming and welding of a cylindrical vessel, as presented in Figure 5. Both ends of the vessel are capped, and the head has a hemispherical shape. This EOP consists of four decision variables, namely, thickness of the shell (T_S), thickness of the head (T_h), inner radius (R) and length of the cylindrical section without considering the head (L) [38].

The mathematical formulation of this EOP is as follows:

$$\text{Minimize} C_{Sys.} = 0.6224T_sRL + 1.7781T_hR^2 + 3.1661T_s^2L + 19.84T_s^2R \tag{11}$$

subjected to

$$-T_s + 0.0193R \le 0$$

$$-R + 0.0193R \le 0$$

$$-\pi R^2L - \frac{4}{3}\pi R^3 + 1{,}296{,}000 \le 0$$

$$L - 240 \le 0$$

$$\text{Variable range} \quad 0 \le T_s,\ T_h \le 99 \quad 10 \le\ R,\ L \le 200$$

4. Results and Discussion

To evaluate the performance of the MWHO algorithm on SROPs, the proposed MWHO algorithm was implemented on the two SROPs describe in Section 3 using GNU Octave version 6.4.0 on a personal computer with the following performance: 11th Gen Intel Core I5-1135G7, 2.4 GHz * 8 and 16 GB of RAM. To obtain the best working combination of parameters for MWHO, a trial-and-error methodology was used.

The proposed algorithm was applied to these SROPs with the following parameters: For SROP 1: number of stallions, 40; crossover percentage, 20; number of iterations, 1000; and for SROP 2: number of stallions, 40; crossover percentage, 20; number of iterations, 500; and for EOP (PVD): number of stallions, 40; crossover percentage, 20; number of iterations, 100. The MWHO algorithm was executed independently for ten runs for each SROP and

EOP (PVD). The best, worst, mean and standard deviation values for each run obtained by MWHO algorithm are reported in Tables 2–4.

Figure 6 depicts the convergence curve obtained by MWHO in 10 different runs for SROP1 and SROP2.

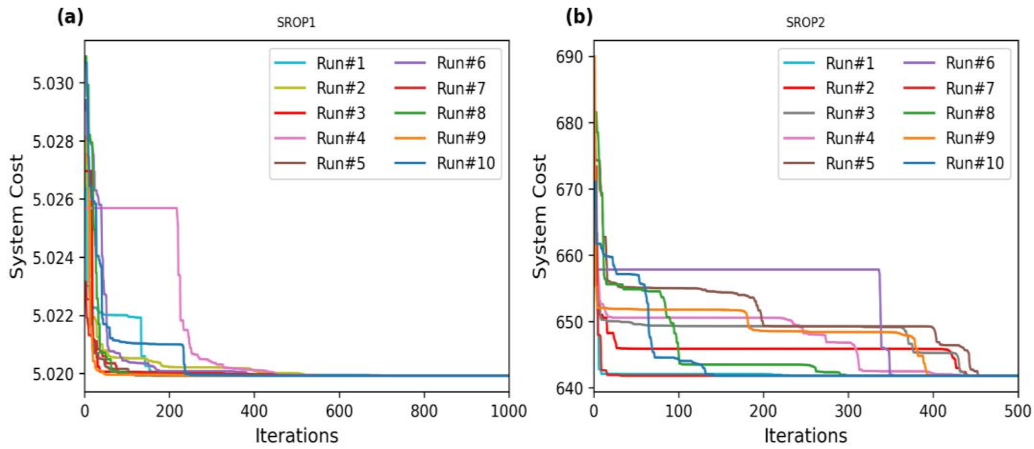

Figure 6. Convergence curve obtained by MWHO in 10 different runs for SROPs (first 500 iterations). (**a**) SROP1 (**b**) SROP2.

As shown in Figure 6, the system cost converges towards optimal value approximately after 600 iterations across the 10 runs in SROP1 (Figure 6a). On the other hand, the system cost converges towards optimal value approximately after 460 iterations in SROP2 (Figure 6b). Figures 7 and 8 present the convergence curve of the best run for SROP 1 and SROP 2. For EOP (PVD), system cost converges towards optimal value after approximately 25 iterations (Figure 9a). It is also observed that the initial values taken by the optimization algorithm have little effect on the convergence towards the optimal points. However, some initial values can lead to reaching the optimal point faster than some other initial values of the parameters. As shown in Figures 7, 8 and 9b, the best run, defined as the run providing minimum cost, converges towards the optimal point faster than most of the runs. We do not conclude that faster convergence means the best solution, as in Figure 6a, run #9 shows a lower system cost compared to the best run (run #3). However, a faster convergence does have an advantage, as the search particles can descend towards the optimal point faster and save computational time.

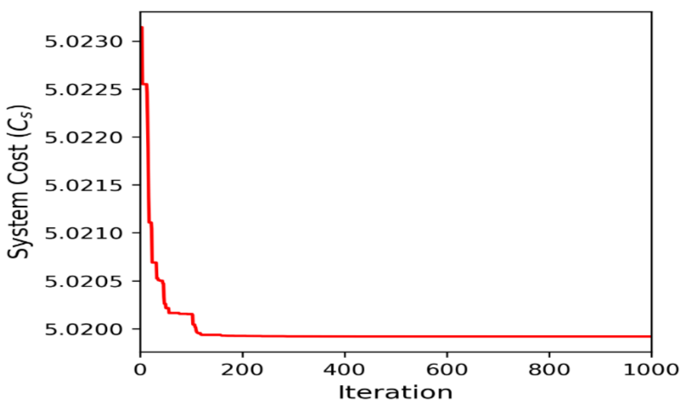

Figure 7. Convergence curve of SROP 1.

Table 2. Results of ten MWHO runs for SROP 1.

Parameter	$C_{Sys.}$	R_1	R_2	R_3	R_4	R_5	Rsys.	Minimum (Best)	Maximum (Worst)	Mean	Standard Deviation
Run1	5.0199282842	0.9360195664	0.9337034472	0.7981921280	0.9371237883	0.9320169548	0.9900000000	5.0199282842	5.0309085098	5.0201892919	0.0013487858
Run 2	5.0199280124	0.9355263294	0.9334803358	0.7822209813	0.9343527087	0.9371561818	0.9900000000	5.0199280124	5.0230545661	5.0200296731	0.0003101977
Run 3	5.0199184060	0.9349331779	0.9348248186	0.7913341473	0.9353969594	0.9344941166	0.9900000000	5.0199184060	5.0231438177	5.0200024039	0.0003699874
Run 4	5.0199242231	0.9340927010	0.9339336516	0.7986320543	0.9343808422	0.9365264065	0.9900000000	5.0199242231	5.0274058684	5.0202714576	0.0008451419
Run 5	5.0199278748	0.9340675052	0.9347335655	0.8031105659	0.9327614533	0.9368741416	0.9900000000	5.0199278748	5.0269632952	5.0200872892	0.0009139384
Run 6	5.0199261851	0.9327023268	0.9363847777	0.7910815478	0.9334170090	0.9370704737	0.9900000000	5.0199261851	5.0256903672	5.0213169039	0.0023479312
Run 7	5.0199270145	0.9345176482	0.9339328057	0.7935430577	0.9333222283	0.9376149951	0.9900000000	5.0199270145	5.0275383672	5.0200320369	0.0007645991
Run 8	5.0199260865	0.9336921006	0.9370912052	0.7980113819	0.9331846134	0.9349835171	0.9900000000	5.0199260865	5.0281582787	5.0202224804	0.0008470497
Run 9	5.0199261773	0.9367178093	0.9337026258	0.7966995010	0.9361658666	0.9324564062	0.9900000000	5.0199261773	5.0294137197	5.0203531091	0.0014706063
Run 10	5.0199194160	0.9351343121	0.9349204035	0.7930024286	0.9356598048	0.9337626176	0.9900000000	5.0199194160	5.0306942338	5.0204067134	0.0013594992

Table 3. Results of ten MWHO runs for SROP 2.

Parameter	$C_{Sys.}$	R_1	R_2	R_3	R_4	Rsys.	Minimum (Best)	Maximum (Worst)	Mean	Standard Deviation
Run 1	641.8235682227	0.5000000000	0.8389200456	0.5000000000	0.5000000548	0.9000000000	641.8235682227	671.0551405545	644.2835421315	5.5640172063
Run 2	641.8235623261	0.5000000000	0.8389201009	0.5000000000	0.5000000000	0.9000000000	641.8235623261	671.9281071267	642.1114292308	1.8683118611
Run 3	641.8235623261	0.5000000000	0.8389201009	0.5000000000	0.5000000000	0.9000000000	641.8235623261	681.5619277020	645.2677115347	6.3491794550
Run 4	641.8235623263	0.5000000000	0.8389201009	0.5000000000	0.5000000000	0.9000000000	641.8235623263	667.5127214183	645.5943686963	2.2888766297
Run 5	641.8235623262	0.5000000000	0.8389201009	0.5000000000	0.5000000000	0.9000000000	641.8235623262	681.9891680944	648.0256628239	3.4253433355
Run 6	641.8354495645	0.5004169091	0.8384997756	0.5000000000	0.5000000000	0.9000000000	641.8354495645	673.3826280527	642.1138823691	2.5553654839
Run 7	641.8789866199	0.5019360000	0.8369721277	0.5000000000	0.5000000000	0.9000000000	641.8789866199	689.9675908673	648.2237515217	4.1702652678
Run 8	641.8235623262	0.5000000000	0.8389201009	0.5000000000	0.5000000000	0.9000000000	641.8235623262	689.8633502043	647.1428418598	4.5464971105
Run 9	641.8235624518	0.5000000000	0.8389201018	0.5000000000	0.5000000000	0.9000000003	641.8235624518	687.8662298184	650.8092023998	5.4946777022
Run 10	641.8449422722	0.5007491730	0.8381651189	0.5000000000	0.5000000000	0.9000000000	641.8449422722	657.8461365447	652.7289293272	7.3862098280

Table 4. Results of ten MWHO runs for the EOP (PVD).

Parameter	$C_{Sys.}$	T_{S}	T_{h}	R	L	Minimum (Best)	Maximum (Worst)	Mean	Standard Deviation
Run 1	5885.332774	0.778168641	0.384649163	40.31961872	200	5885.332774	188,242.5454	6005.540996	2557.519377
Run 2	5923.354381	0.799805064	0.395344057	41.44067687	184.9614972	5923.354381	213,759.7422	6033.154867	3775.860149
Run 3	6003.185713	0.841774819	0.416089729	43.61527557	158.7055172	6003.185713	554,447.4169	6185.216133	6475.883025
Run 4	5885.332774	0.778168641	0.384649163	40.31961872	200	5885.332774	601,327.7629	6124.196893	9227.022274
Run 5	5885.332774	0.778168641	0.384649163	40.31961872	200	5885.332774	171,610.9702	6031.333811	2601.655011
Run 6	5887.683154	0.77954125	0.385327644	40.39073833	199.0123263	5887.683154	84,546.53112	5972.584245	2096.40849
Run 7	5955.328956	0.817134416	0.403909965	42.33857078	173.6836087	5955.328956	119,618.6572	6021.663479	1934.093259
Run 8	5885.332774	0.778168641	0.384649163	40.31961872	200	5885.332774	204,567.4258	6013.034189	3430.863708
Run 9	5900.162061	0.786749181	0.388890528	40.76420625	193.9022357	5900.162061	344,169.3199	6096.501464	4300.237632
Run 10	5885.332774	0.778168641	0.384649163	40.31961872	200	5885.332774	131,447.596	5991.645744	3183.610924

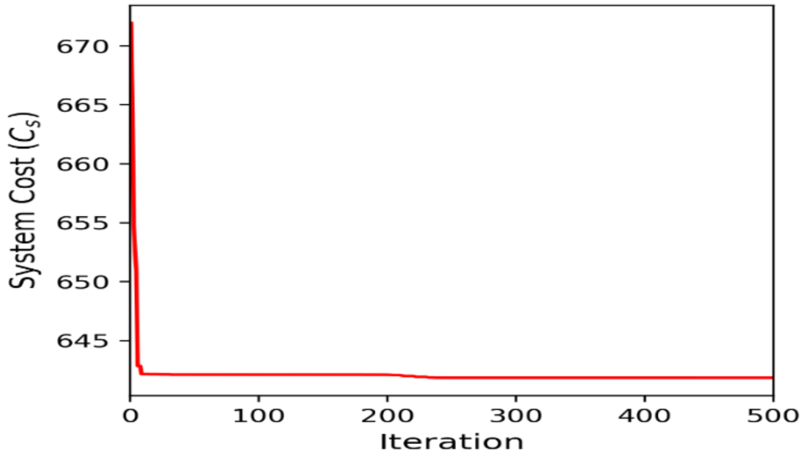

Figure 8. Convergence curve of SROP 2.

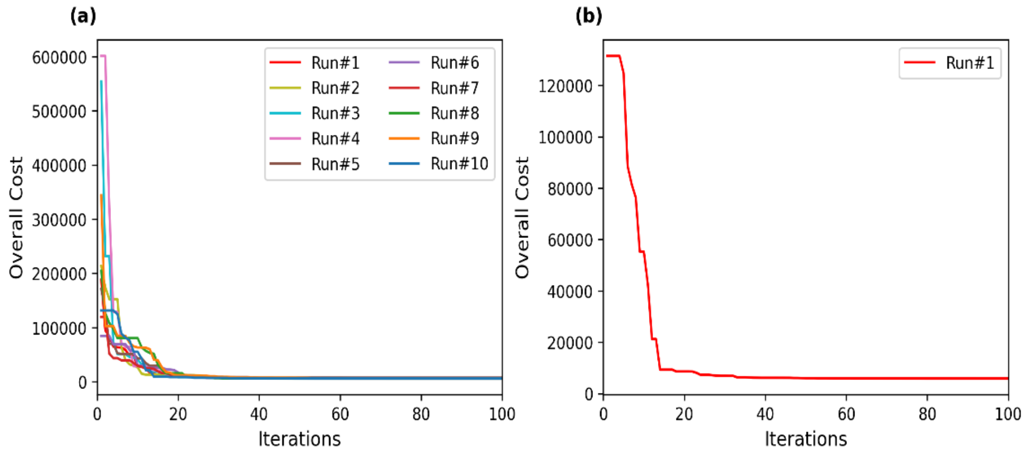

Figure 9. (**a**) Convergence curve obtained by MWHO in 10 different runs for EOP (PVD) (**b**) Convergence curve of EOP (PVD).

Tables 2–4 provide the statistical results of MWHO for SROP1, SROP2 and EOP (PVD), respectively. They reflect the minimum (best), maximum (worst), mean and standard deviation of the system cost for each run in fixed iterations. Based on the convergence curves (Figures 6–9) and statistical results presented in Tables 2–4, Run 3 has been identified as the best run for SROP1, Run 2 has been identified as the best run for SROP2, while Run 1 has been identified as the best run for EOP (PVD). Run 3 for SROP1 has a minimum system cost of 5.0199184060, with a mean cost of 5.0200024039 and a standard deviation of 0.0003699874, whereas Run 2 for SROP2 has a minimum system cost of 641.8235623261, with a mean cost of 642.1114292308 and a standard deviation of 1.8683118611. Run 1 for EOP (PVD) has a minimum system cost of 5885.332774, with a mean cost of 6005.540996 and a standard deviation of 2557.519377.

Comparison of the best results obtained by MWHO for SROP 1, SROP 2 and EOP (PVD) with other metaheuristics is presented in Tables 5–7, respectively.

Table 5. Comparison of the best results for SROP 1 obtained using different algorithms.

Parameter	$C_{Sys.}$	R_1	R_2	R_3	R_4	R_5	Rsys.	FE
MWHO	5.0199184060	0.9349331779	0.9348248186	0.7913341473	0.9353969594	0.9344941166	0.9900000000	40,000
PSO	5.019918	0.935028	0.791948	0.935005	0.934735	0.934821	0.990000	120,000
GWO	5.019900	0.934100	0.936350	0.791370	0.933880	0.935650	0.990028	9000
CSA	5.019980	0.935546	0.788534	0.941231	0.927708	0.934900	0.990000	60,000
ACO	5.019923	0.935073	0.798365	0.935804	0.934223	0.933869	0.990001	80,160

Table 6. Comparison of the best results for SROP 2 obtained using different algorithms.

Parameter	$C_{Sys.}$	R_1	R_2	R_3	R_4	Rsys.	FE
MWHO	641.8235623261	0.5000000000	0.8389201009	0.5000000000	0.5000000000	0.9000000000	20,000
PSO	641.823562	0.838920	0.500000	0.500000	0.500000	0.900000	2040
GWO	641.823600	0.500000	0.838920	0.500000	0.500000	0.900000	50,000
CSA	641.823563	0.838920	0.500000	0.500000	0.500000	0.900000	15,000
ACO	641.823562	0.838920	0.500000	0.500000	0.500000	0.900000	20,100

Table 7. Comparison of the best results for the EOP (PVD) obtained using different algorithms.

Parameter	$C_{Sys.}$	T_S	T_h	R	L	FE
MWHO	5885.332774	0.778168641	0.384649163	40.31961872	200	4000
PSO	6061.0777	0.812500	0.437500	42.091266	176.746500	-
GWO	6051.5639	0.812500	0.434500	42.089181	176.758731	-
GA	6288.7445	0.812500	0.434500	40.323900	200.000000	-
ACO	6059.0888	0.812500	0.437500	42.103624	176.572656	-

The results for SROP1 obtained by the MWHO algorithm, along with a few other results, are presented in Table 5. The table demonstrates that the MWHO yields significant improvements over solutions obtained using PSO, GWO, CSA and ACO. With only 40,000 function evaluations, MWHO achieved a minimum system cost of 5.0199184060 while maintaining system reliability of 0.9900000000.

The results for SROP2 obtained by the MWHO algorithm, along with a few other results, are presented in Table 6. The table demonstrates that MWHO provides very competitive results when compared to solutions obtained using PSO, GWO, CSA and ACO. With 20,000 function evaluations, MWHO achieved a minimum system cost of 641.8235623261 while maintaining system reliability of 0.9000000000.

The results for EOP (PVD) obtained by the MWHO algorithm, along with a few other results, are presented in Table 7. The table demonstrates that MWHO yields significant improvements when compared to solutions obtained using PSO, GWO, GA and ACO. With only 4000 function evaluations, MWHO achieved a minimum system cost of 5885.332774.

5. Conclusions and Future Scope

The objective of this article was to introduce a solution approach based on MWHO to deal with SROPs. The proposed MWHO, which is a modified version of WHO, has high performance on SROPs and can handle them with great ease. In addition, the comparative analysis with the results available in the literature dealing with the same SROPs shows that the MWHO algorithm has better efficiency, as it provides either superior or comparable solutions. For further study, our work will be devoted to the development of a hybrid multi-objective version of the MWHO algorithm to deal with multi-objective SROPs. Additionally, analysing the applicability, efficiency and stability of the MWHO on real life engineering cases such as Failure Mode Effects Analysis [43], Bayesian Network predictive analysis, and combining MWHO with some MCDM techniques may be expwereed [44,45].

Author Contributions: A.K. (Anuj Kumar): Conceptualization, Investigation, Methodology, Draft preparation. S.P.: Conceptualization, Investigation and drafting. M.K.S.: Programming, Investigation. S.C.: Conceptualization, Investigation and drafting. M.R.: Drafting, Investigation. A.K. (Akshay Kumar): Drafting, Investigation. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Charles, E.E. *An Introduction to Reliability and Maintainability Engineering*; McGraw Hill: New York, NY, USA, 1997.
2. Kumar, A.; Ram, M.; Pant, S.; Kumar, A. Industrial system performance under multistate failures with standby mode. In *Modeling and Simulation in Industrial Engineering*; Springer: Cham, Switzerland, 2018; pp. 85–100.
3. Yazdi, M.; Mohammadpour, J.; Li, H.; Huang, H.Z.; Zarei, E.; Pirbalouti, R.G.; Adumene, S. Fault tree analysis improvements: A bibliometric analysis and literature review. *Qual. Reliab. Eng. Int.* **2023**, *Early View*. [CrossRef]
4. Li, H.; Yazdi, M.; Huang, H.Z.; Huang, C.G.; Peng, W.; Nedjati, A.; Adesina, K.A. A fuzzy rough copula Bayesian network model for solving complex hospital service quality assessment. *Complex Intell. Syst.* **2023**, 1–27. [CrossRef]
5. Pant, S.; Kumar, A.; Ram, M. Reliability optimization: A particle swarm approach. In *Advances in Reliability and System Engineering*; Springer: Cham, Switzerland, 2017; pp. 163–187.
6. Kumar, A.; Pant, S.; Ram, M.; Yadav, O. (Eds.) *Meta-Heuristic Optimization Techniques: Applications in Engineering*; Walter de Gruyter GmbH & Co KG: Berlin, Germany, 2022; Volume 10.
7. Coit, D.W.; Smith, A.E. Reliability optimization of series-parallel systems using a genetic algorithm. *IEEE Trans. Reliab.* **1996**, *45*, 254–260. [CrossRef]
8. Liang, Y.-C.; Smith, A.E. An ant colony optimization algorithm for the redundancy allocation problem (RAP). *IEEE Trans. Reliab.* **2004**, *53*, 417–423. [CrossRef]
9. Wolpert, D.H.; Macready, W.G. No free lunch theorems for optimization. *IEEE Trans. Evol. Comput.* **1997**, *1*, 67–82. [CrossRef]
10. Modibbo, U.M.; Arshad, M.; Abdalghani, O.; Ali, I. Optimization and estimation in system reliability allocation problem. *Reliab. Eng. Syst. Saf.* **2021**, *212*, 107620. [CrossRef]
11. Ouzineb, M.; Nourelfath, M.; Gendreau, M. Tabu search for the redundancy allocation problem of homogenous series–parallel multi-state systems. *Reliab. Eng. Syst. Saf.* **2008**, *93*, 1257–1272. [CrossRef]
12. Beji, N.; Jarboui, B.; Eddaly, M.; Chabchoub, H. A hybrid particle swarm optimization algorithm for the redundancy allocation problem. *J. Comput. Sci.* **2010**, *1*, 159–167. [CrossRef]
13. Kumar, A.; Pant, S.; Ram, M. System reliability optimization using gray wolf optimizer algorithm. *Qual. Reliab. Eng. Int.* **2017**, *33*, 1327–1335. [CrossRef]
14. Hsieh, Y.-C.; You, P.-S. An effective immune based two-phase approach for the optimal reliability–redundancy allocation problem. *Appl. Math. Comput.* **2011**, *218*, 1297–1307. [CrossRef]
15. Wu, P.; Gao, L.; Zou, D.; Li, S. An improved particle swarm optimization algorithm for reliability problems. *ISA Trans.* **2011**, *50*, 71–81. [CrossRef]
16. Zou, D.; Gao, L.; Li, S.; Wu, J. An effective global harmony search algorithm for reliability problems. *Expert Syst. Appl.* **2011**, *38*, 4642–4648. [CrossRef]
17. Hsieh, T.-J.; Yeh, W.-C. Penalty guided bees search for redundancy allocation problems with a mix of components in series–parallel systems. *Comput. Oper. Res.* **2012**, *39*, 2688–2704. [CrossRef]
18. Lins, I.D.; Droguett, E.L. Redundancy allocation problems considering systems with imperfect repairs using multi-objective genetic algorithms and discrete event simulation. *Simul. Model. Pract. Theory* **2011**, *19*, 362–381. [CrossRef]
19. Wang, L.; Li, L.-P. A coevolutionary differential evolution with harmony search for reliability–redundancy optimization. *Expert Syst. Appl.* **2012**, *39*, 5271–5278. [CrossRef]
20. Wang, Y.; Li, L. Heterogeneous redundancy allocation for series-parallel multi-state systems using hybrid particle swarm optimization and local search. *IEEE Trans. Syst. Man Cybern. Part A Syst. Hum.* **2012**, *42*, 464–474. [CrossRef]
21. Pourdarvish, A.; Ramezani, Z. Cold standby redundancy allocation in a multi-level series system by memetic algorithm. *Int. J. Reliab. Qual. Saf. Eng.* **2013**, *20*, 1340007. [CrossRef]
22. Valian, E.; Valian, E. A cuckoo search algorithm by Lévy flights for solving reliability redundancy allocation problems. *Eng. Optim.* **2013**, *45*, 1273–1286. [CrossRef]
23. Valian, E.; Tavakoli, S.; Mohanna, S.; Haghi, A. Improved cuckoo search for reliability optimization problems. *Comput. Ind. Eng.* **2013**, *64*, 459–468. [CrossRef]
24. Afonso, L.D.; Mariani, V.C.; dos Santos Coelho, L. Modified imperialist competitive algorithm based on attraction and repulsion concepts for reliability-redundancy optimization. *Expert Syst. Appl.* **2013**, *40*, 3794–3802. [CrossRef]
25. Ardakan, M.A.; Hamadani, A.Z. Reliability optimization of series–parallel systems with mixed redundancy strategy in subsystems. *Reliab. Eng. Syst. Saf.* **2014**, *130*, 132–139. [CrossRef]
26. Yeh, W.-C. Orthogonal simplified swarm optimization for the series–parallel redundancy allocation problem with a mix of components. *Knowl. Based Syst.* **2014**, *64*, 1–12. [CrossRef]
27. Kumar, A.; Negi, G.; Pant, S.; Ram, M.; Dimri, S.C. Availability-Cost Optimization of Butter Oil Processing System by Using Nature Inspired Optimization Algorithms. *Reliab. Theory Appl.* **2021**, *64*, 188–200.
28. Huang, C.-L. A particle-based simplified swarm optimization algorithm for reliability redundancy allocation problems. *Reliab. Eng. Syst. Saf.* **2015**, *142*, 221–230. [CrossRef]
29. He, Q.; Hu, X.; Ren, H.; Zhang, H. A novel artificial fish swarm algorithm for solving large-scale reliability–redundancy application problem. *ISA Trans.* **2015**, *59*, 105–113. [CrossRef]
30. Ardakan, M.A.; Hamadani, A.Z.; Alinaghian, M. Optimizing bi-objective redundancy allocation problem with a mixed redundancy strategy. *ISA Trans.* **2015**, *55*, 116–128. [CrossRef]

31. Zhuang, J.; Li, X. Allocating redundancies to k-out-of-n systems with independent and heterogeneous components. *Commun. Stat.-Theory Methods* **2015**, *44*, 5109–5119. [CrossRef]
32. Garg, H. An approach for solving constrained reliability-redundancy allocation problems using cuckoo search algorithm. *Beni-Suef Univ. J. Basic Appl. Sci.* **2015**, *4*, 14–25. [CrossRef]
33. Mellal, M.A.; Zio, E. A penalty guided stochastic fractal search approach for system reliability optimization. *Reliab. Eng. Syst. Saf.* **2016**, *152*, 213–227. [CrossRef]
34. Abouei Ardakan, M.; Sima, M.; Zeinal Hamadani, A.; Coit, D.W. A novel strategy for redundant components in reliability-redundancy allocation problems. *IIE Trans.* **2016**, *48*, 1043–1057. [CrossRef]
35. Gholinezhad, H.; Hamadani, A.Z. A new model for the redundancy allocation problem with component mixing and mixed redundancy strategy. *Reliab. Eng. Syst. Saf.* **2017**, *164*, 66–73. [CrossRef]
36. Kim, H.; Kim, P. Reliability models for a nonrepairable system with heterogeneous components having a phase-type time-to-failure distribution. *Reliab. Eng. Syst. Saf.* **2017**, *159*, 37–46. [CrossRef]
37. Kim, H. Maximization of system reliability with the consideration of component sequencing. *Reliab. Eng. Syst. Saf.* **2018**, *170*, 64–72. [CrossRef]
38. Garg, H. A hybrid GSA-GA algorithm for constrained optimization problems. *Inf. Sci.* **2019**, *478*, 499–523. [CrossRef]
39. Al-Azzoni, I.; Iqbal, S. Meta-heuristics for solving the software component allocation problem. *IEEE Access* **2020**, *8*, 153067–153076. [CrossRef]
40. Kumar, A.; Pant, S.; Ram, M. Gray wolf optimizer approach to the reliability-cost optimization of residual heat removal system of a nuclear power plant safety system. *Qual. Reliab. Eng. Int.* **2019**, *35*, 2228–2239. [CrossRef]
41. Negi, G.; Kumar, A.; Pant, S.; Ram, M. Optimization of complex system reliability using hybrid grey wolf optimizer. *Decis. Mak. Appl. Manag. Eng.* **2021**, *4*, 241–256. [CrossRef]
42. Naruei, I.; Keynia, F. Wild horse optimizer: A new meta-heuristic algorithm for solving engineering optimization problems. *Eng. Comput.* **2021**, *38*, 3025–3056. [CrossRef]
43. Li, H.; Soares, C.G.; Huang, H.Z. Reliability analysis of a floating offshore wind turbine using Bayesian Networks. *Ocean Eng.* **2020**, *217*, 107827. [CrossRef]
44. Pant, S.; Garg, P.; Kumar, A.; Ram, M.; Kumar, A.; Sharma, H.K.; Klochkov, Y. AHP-based multi-criteria decision-making approach for monitoring health management practices in smart healthcare system. *Int. J. Syst. Assur. Eng. Manag.* **2023**, 1–12. [CrossRef]
45. Li, H.; Díaz, H.; Soares, C.G. A failure analysis of floating offshore wind turbines using AHP-FMEA methodology. *Ocean. Eng.* **2021**, *234*, 109261. [CrossRef]