Iraqi Journal for Computer Science and Mathematics
Journal Homepage: http://journal.esj.edu.iq/index.php/IJCM
ISSN: 2788-7421 p-ISSN: 2958-0544

Improvement of Network Reliability by Hybridization of the Penalty Technique Based on Metaheuristic Algorithms

Roaa Aziz fadhil[1,*], Zahir Abdul Haddi Hassan [2]

[1]Department of Mathematics,College of Education for Pure Sciences University of Babylon
[2]Professor, Department of Mathematics,College of Education for Pure Sciences University of Babylon

*Corresponding Author:Roaa Aziz fadhil

DOI: https://doi.org/10.52866/ijcsm.2024.05.01.007
Received July 2023 ; Accepted November 2023 ; Available online January 2024

ABSTRACT:
In this study, a network created by converting the Petri net into a network for shutdown system, by calculating a network's nonlinear reliability polynomial by splitting the network into two, each having a source node and an terminal node. This the procedure converts the system into two parallel-series block diagrams that are connected in a series. The metaheuristic algorithm using the honey badger algorithm (HBA) and Dwarf Mongoose Optimization Algorithm (DMO) have been used to improve the problem for the shutdown network hybridizing these algorithms employing the penalty method, we will obtained the (PFMHBA, PFMDMO) algorithms. Thereafter, we compared the results of the use of the hybridization technique with the results of the process that does not use this technique. The objective of this comparison was to ascertain whether the use of hybridization makes the results increasing reliable, reducing cost, and shortening the duration of implementation.

Keywords: Network reliability, penalty method, honey badger algorithm, dwarf mongoose optimization algorithm

1. INTRODUCTION

Metaheuristic algorithms respond differently when addressing certain tasks. Algorithms are developed in such a way that a high-performance algorithm that solves particular categories for problems because no algorithm can solve every problem in the best possible way, so as to find optimal solutions to certain complicated and major optimization issues [41], or help determine whether a given algorithm is the best one to solve network reliability problems. The possibility that each pair of nodes can communicate with one another is known as all-terminal network reliability, or overall network reliability. The primary design faces difficulties in terms of determining which set of links can be utilized for a given collection of nodes to either maximize reliability or reduce cost. Designing reliable networks is adversely affected by NP-hard design problems [1]. The evaluation of network reliability [5, 23] was carried out using a range of tools and techniques. Because network are based on graph theory, we used, from among the existing approaches used to assess the dependability of networks, the graph-based algorithms that, in turn, use either minimal path sets or minimal cut sets [7, 11, 13]. Any nonlinear reliability optimization problem may be solved using the Generalized Reduced Gradient (GRG) method. Two reliability issues are resolved using the Luus and Jaakola method and sequential unconstrained minimization technique, and the outcomes are compared [12]. The first problem maximizes system reliability while minimizing system cost, the second issue is to reduce system reliability while minimizes system cost [12, 14, 19], performance indicators and optimization methods. The development of reliability allocation methods have advanced significantly over the past few decades [2, 6], and reliability-enhancing choices. A general strategy for distributing network reliability that may be used on any network with identical or mismatched components was suggested by Aggarwal and Shashwati. The idea is built on the numerical analytical method [15]. Real-world engineering systems almost always have many competing objectives, hence

*Corresponding author:roaaazizfadhil@gmail.com,roaa.fadhil.pure256@student.uobabylon.edu.iq

multi-objective optimization is becoming increasingly popular, leading some providers to Create the Pareto optimum set of perfect solutions [16]. The optimization problem is often referred to as a many-objective optimization problem when there are several objectives [17, 39].

The Nondominant Gene Sorting Algorithm, or NSGA II, was created. Using an object-oriented methodology, residents were assigned to a set of objects, and each individual was assigned to a corresponding object. A Pareto process based on individual congestion ranks and levels was used, and the recommended method with multi-objective optimization was used to assign system reliability. The multi-objective concept optimization technique is based on the ability to calculate the maximum permissible variables of the electric vehicle speedbox perception[38]. Multi-objective system optimization through adaptive particle swarm optimization (ADAP-PSO) is required. In order to optimize locally and ensure diversification in the exploration of the search space, this approach uses the article component penalty function. The multi-objective problem is replaced by a single-objective problem, [20]. The issue is described as an optimization challenge for improving overall system reliability under resource constraints. Using a penalty function and interval coefficients, an advanced GA was used to address the problem of unconstrained integer programming [22]. In this research, we emphasize improving the numerical results obtained for both the Honey Badger and Dwarf Mongoose optimization algorithms by using a method to solve the multiobjective optimization problems of the shutdown network. This paper is organized as follows. In Section 2, we introduce the ground truth primitives for the Dwarf Mongoose optimization algorithm. In Section 3, In section 3, the shutdown system converted into an complex network and the reliability polynomial for it has been obtained. In Section 4, The use of algorithms to increase network reliability is covered in Section 4. Section 5 the results of the study, and the last section for conclusions.

2. BACKGROUND

In order to calculate accurate approximations to solutions to optimization problems that are challenging or impossible to solve with other optimization techniques like linear programming and nonlinear programming, many metaheuristic and evolutionary algorithms, frequently inspired by natural systems, are used. Meta-heuristic and evolutionary algorithms are widely applicable, independent-of-the-problem methods for resolving a wide range of difficult engineering problems. To solve complicated engineering optimization problems [41]. Due to the importance of these algorithms, we highlight them by presenting some of them, as well as trying to improve them by conducting a hybridization process for optimizing network reliability problems.

2.1 NETWORKS RELIABILITY

The ability of a system to run as intended, faultlessly, in an operational environment, for a certain amount of time, is the generally accepted definition of "reliability", in the engineering community. A more complete definition of reliability is that it is the science of anticipating, assessing, avoiding, and mitigating problems throughout time [44]; any system that can be visualized as a collection of tiny circles (nodes) connected by lines (edges) is referred to as a network. A broad discipline, networks deal with a variety of real-world systems in areas including industry, communication, software engineering, ..., etc. [45].

Definition 1. *[42, 43] The minimal path set connects the source and sink nodes as long as it does not contain any cycles. The minimal path set cannot be reduced, as it has no redundant elements, and removing any of the edges from the path means that the source and sink nodes are no longer connected.*

2.2 THE HONEY BADGER ALGORITHM

Honey badger (HB) foraging behavior is mimicked by the honey badger algorithm (HBA). The HB looks for food sources either by tracking the honey guide bird or by sniffing and burrowing. The first situation is referred to as the "digging mode", while the second section is referred to as "honey mode". It employs its keen sense of smell to locate the prey. When the prey comes, it scans the area around the prey to determine the ideal location for burrowing and capturing it. In the concluding scenario, the honey badger locates the beehive correctly in the final scenario by following the honey guide bird's lead [4, 24, 25].

2.2.1. Mathematical Model

These subsections provide an explanation of the HBA's mathematical models [24, 25]. It combines the stages of exploration and exploitation. Theoretically, global optimization is done using the Honey Badger Algorithm (HBA).The population of potential solutions (X) in the HBA is represented, mathematically as:

$$X = \begin{bmatrix} \varkappa_{11} & \varkappa_{12} & \varkappa_{13} & \dots & \varkappa_{1D} \\ \varkappa_{21} & \varkappa_{22} & \varkappa_{23} & \dots & \varkappa_{2D} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varkappa_{n1} & \varkappa_{n2} & \varkappa_{n3} & \dots & \varkappa_{nD}, \end{bmatrix},$$

By its position, each badger in the population is represented $\varkappa_i = \varkappa_i^1, \varkappa_i^2, ..., \varkappa_i^D$, where D indicates the number of variables, and $i = 1, 2, 3, ..., n$, where n is the population size.

- Step 1. Initialization phase: Equation (1) can be used to initialise the respective positions of honey badgers with n population, in detail, the HBA algorithm is as follows

$$\varkappa_i = \ell b_i + \varrho_1 \varkappa (ub_i - \ell b_i), \varrho_1 \in [0,1], \quad (1)$$

  where $\varkappa_i$ is the *ith* honey badger position and corresponds to a candidate solution in a population of size n, while ub_i, ℓb_i, denote the upper and lower bounds of the search space, respectively.

- Step 2. Determining Intensity (I): The two main parameters that impact intensity are the prey concentration strength and the separation between it and the $i-th$ honey badger. The prey's scent intensity is I_i, when the scent is faint, the prey travels slowly and vice versa. It displays the potency of the prey's scent. Inverse Square Law (ISL) offers it.

$$I_i = \varrho_2 \times \frac{S}{(4\pi d_i^2)}, \varrho_2 \in [0,1], \quad (2)$$

$$S = (\varkappa_i - \varkappa_{(i+1)})^2, \quad (3)$$

$$d_i = \varkappa_{prey} - \varkappa_i, \quad (4)$$

  where the source's power or concentration distance is denoted by the symbol S, d_i is what separates the *ith* badger from its victim.

- Step 3.Update density factor: Timing-dependent randomization, which is managed by the density factor (α), ensures a seamless transition from exploration to exploitation. to lessen randomization over time, use Equation (5), which updates the decreasing factor α with each iteration.

$$\alpha = \mathcal{C} \times exp(\frac{-t}{t_{max}}), \quad (5)$$

  where ($\mathcal{C}$) is an integer constant greater than 1 (the value of 2 as the default), and (t_{max}) represents the maximum number of iterations.

- Step 4. Leaving the local optimum: Use this stage along with the next two to exit local optima regions. The suggested approach uses a flag *digamma* in this case to modify search direction, giving agents a high chance to exhaustively explore the search space.

- Step 5. The HBA position update procedure ($\varkappa_{new}$) features two unique phases, the "digging phase" and the "honey phase," as was already explained. The following provides an explanation.

  1. Digging phase: Equation (6) Honey badgers act in a manner resembling a cardioid shape.

$$\varkappa_{new} = \varkappa_{prey} + F \times \beta \times I \times \varkappa_{prey} + F \times \varrho_3 \times \alpha \times d_i \times |cos(2\pi\varrho_4)[1 - cos(2\pi\varrho_5)]|, \quad (6)$$

  where $\varkappa_{prey}$ represents global optimum, or the best position of the prey as of yet. $\beta \geq 1$ indicates the honey badger's capacity to find food (default value is 6) ϱ_3,ϱ_4, and ϱ_5 are three different generated random numbers inside the range $[0,1]$. The search direction is controlled by the flag F, which is defined by (7)

$$F = \begin{cases} 1 & , if \ \varrho_6 \leq 0.5 \\ -1 & otherwise, \end{cases} \quad (7)$$

  where $\varrho_6 \in [0,1]$.

2. Equation (8) describes how the honey badger is guided to the hive by the guide bird:

$$\varkappa_{new} = \varkappa_{prey} + F \times \varrho_7 \times \alpha \times d_i, \tag{8}$$

where $r_7 \in [0,1]$, the prey position is represented by x_{prey}, the badger's most recent location is represented by $\varkappa_{new}$, α and F and are determined using equations (5) and (7), respectively. Equation (8) can be used to demonstrate that a badger conducts a search close to the prey position $\varkappa_{prey}$ that has already been identified based on the spatial awareness of d_i,in this stage,time-varying search behaviour α has an impact on the search. A honey badger may also find F disturbance.

2.3 DWARF MONGOOSE OPTIMIZATION ALGORITHM

Dwarf Mongoose Optimization Algorithm (DMO) is an algorithm with a population-based metaheuristic that drew its inspiration from the foraging and social behavior of dwarf mongooses. It uses a mathematical model to address optimization issues and starts with random initialization, focusing on the global best optimal solution. [26, 27].

2.3.1. The DMO mathematical model

The candidate population of the mongoose is initialized as the first step in the DMO solution. This population is produced stochastically between the lower and upper boundaries of a certain task.

$$X = \begin{bmatrix} \varkappa_{11} & \varkappa_{12} & \varkappa_{13} & \dots & \varkappa_{1d} \\ \varkappa_{21} & \varkappa_{22} & \varkappa_{23} & \dots & \varkappa_{2d} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \varkappa_{n1} & \varkappa_{n2} & \varkappa_{n3} & \dots & \varkappa_{nd} \end{bmatrix}, \tag{9}$$

where X stands for the population of candidates that are running for office d, which represents the problem dimension, was made, n, which represents the population size, the DMO begins by setting the initial value to a group solution using the following Equation (10)

$$\varkappa_{i,j} = \ell_j + rand \times (u_j - \ell_j), \tag{10}$$

where $rand$ is a random number from [0, 1]. The u_j and ℓ_j limits define the search domain. The Alpha Group, Scouts, and Babysitters are the three groups that make up the DMO. Each team uses a unique strategy to capture the food. The following is used to represent the details of these groups:

1. Alpha Group: After the population has been formed, the fitness of each solution is determined. The alpha female G_α is identified using Equation (11), which establishes the fitness probability value for each population.

$$G_\alpha = \frac{fit_i}{\sum_{i=1}^{n} fit_i}, \tag{11}$$

The $n - bs$, where bs is the quantity of nannies. Peep denotes that the family stays on course thanks to the dominant female's vocalization. Every mongoose that rests in the initial sleeping mound is set phi, which is situated they all spend the night. The DMO uses Equation (12) to choose an applicant for a food position.

$$\varkappa_{i+1} = \varkappa_i + phi \times peep, \tag{12}$$

There is a uniform distribution for the value phi, which ranges from $[-1,1]$. Equation(13) outputs the sleeping mound after each repetition.

$$sm_i = \frac{fit_{i+1} - fit_i}{max(|fit_{i+1}, fit_i|)}, \tag{13}$$

Equation (14) gives the average number of sleeping mounds that have been found.

$$\varphi = \frac{\sum_{i=1}^{n} fit_i}{n}. \tag{14}$$

2. Scout Group: When the conditions for criteria exchange babysitting are met. The program moves on to the scouting stage, when it analyzes the next food source or sleeping mound, after the prerequisite for a daycare swap is satisfied. Scouts look for the next sleeping mound to ensure exploration because mongooses are known to shun old ones. In our model, which combines foraging and reconnaissance, the direction he walks depends on whether it is successful in locating a new sleeping mound. The family will come to a fresh sleeping mound if they stray far enough. Additionally. Equation (15) is a representation of the scout mongoose.

$$\varkappa_{i+1} = \begin{cases} \varkappa_i - cf \times phi \times rand \times [\varkappa_i - \vec{\mu}] & , if \varphi_{i+1} > \varphi_i \\ \varkappa_i + cf \times \varphi \times rand \times [\varkappa_i - \vec{\mu}] & , otherwise, \end{cases} \tag{15}$$

where $rand \in [0.1]$. Equation (16) to compute (cf) value and Equation (17) is used to calculate value μ.

$$cf = (1 - \frac{iter}{max_{iter}})^{2 \times \frac{iter}{max_{iter}}}, \tag{16}$$

$$\vec{\mu} = \frac{\sum_{i=1}^{n} \varkappa_i \times sm_i}{\varkappa_i}. \tag{17}$$

3. Babysitters Group: Inferior group members who look after the young are frequently cycled out of the group on a regular basis, allowing the alpha female to lead the other group members on daily foraging forays. Babysitters influence the algorithm, depending on the percentage selected, by diminishing the size of the population as a whole; their predominance is proportional to the population's size. By lowering the number of people in the population by the percentage of babysitters, we might be able to recreate this group. The scouting and food source data previously held by the family members replacing them are reset by the babysitter exchange parameter.

2.4 PENALTY FUNCTION METHOD (PFM)

Optimization problems might be written in as follows

$$\begin{aligned} & min \ or \ max \quad f(\varkappa) \\ & subject \ to \quad h_j(\varkappa) = 0 \quad , j = 1, 2, ..., l \\ & \qquad\qquad g_i(\varkappa) \geq, i = 1, 2, ..., m \\ & \qquad\qquad \varkappa \in R^n \end{aligned} \tag{18}$$

Constrained optimization problems are transformed into unconstrained optimization problems using penalty methods [28], with penalty terms is written as equation (19).

$$F(\varkappa) = f(\varkappa) + p(\varkappa, \kappa, \vartheta), \tag{19}$$

The penalty function p is usually of the form

$$p(\varkappa, \kappa, \vartheta) = \sum_{j=1}^{l} \kappa_j h_j^2(\varkappa) + \sum_{i=1}^{m} \vartheta_i g_i(\varkappa)^2, \tag{20}$$

The parameters κ_j and ϑ_i are called penalty parameters and the function

$$g_i(\varkappa) = max\{0, g_i(\varkappa)\}, \tag{21}$$

And Equation (20) is the quadratic penalty. It is possible to create penalty functions that are precise in the sense that, for a given value of the penalty parameter, the solution to the penalty problem returns the exact solution to the original problem. These functions eliminate the need to work through an endless series of penalty issues in order to arrive at the right answer. However, the fact that a new difficulty introduced by these penalty functions is that they are non differentiable, and where the exact Penalty function is defined by [29, 30].

$$F(\varkappa) = f(\varkappa) + \sum_{j}^{l} \kappa_j |h_j(\varkappa)| + \sum_{i}^{m} \vartheta_i g_i(\varkappa), \tag{22}$$

One simple solution for problems with inequality constraints is to turn a constrained optimization problem into an unconstrained one, which can be accomplished by imposing a penalty for every violation of a constraint known as an exterior penalty method [31, 32] terms is written as Equation (23).

$$F(\varkappa) = f(\varkappa) + \vartheta_k \sum_{i=1}^{m} (g_i(\varkappa))^q, \tag{23}$$

where exponent q is a non-negative constant and ϑ_k is a positive penalty parameter.

3. CASE STUDY OF NETWORK

We use the network of the shut down system [33] in the Figure (1), conversion In this case, the Petri Nets (PN) is transformed into a network where the nodes stand in for the locations, and a single edge replaces the transitions and their connecting arcs. [34]. We obtain the network represented in the Figure (2).

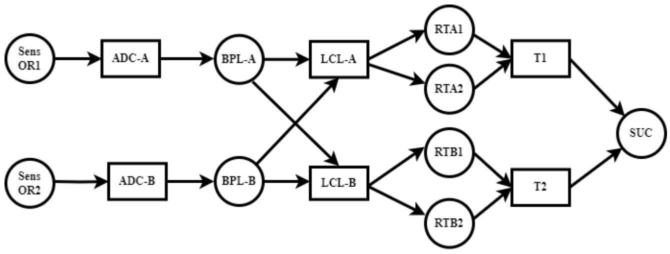


FIGURE 1. **Petri net of shutdown system**

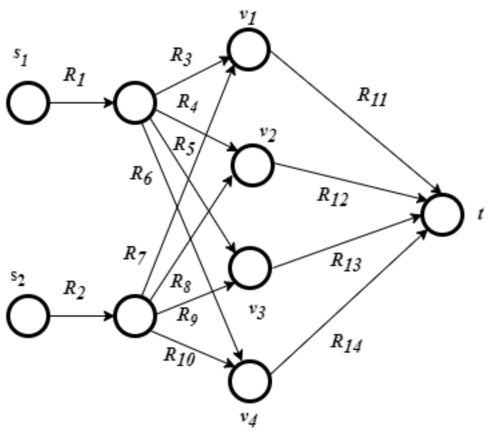
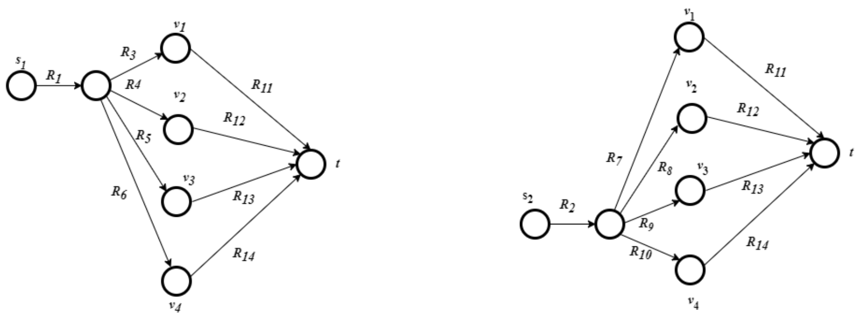


FIGURE 2. **Shutdown network**

3.1 RELIABILITY POLYNOMIAL OF THE SHUTDOWN NETWORK

Most of the methods for computation are polynomial to networks that contain one source vertex and one terminal vertex [3, 35], While the shoutdown network contains two source vertices and one terminal, to obtain the network equation, we will suggest that we divide the network into two sub networks, each containing one source, as shown in Figure(3).



(a) **Shoutdown sub network 1** (b) **Shoutdown sub network 2**

FIGURE 3. Shoutdown sub network

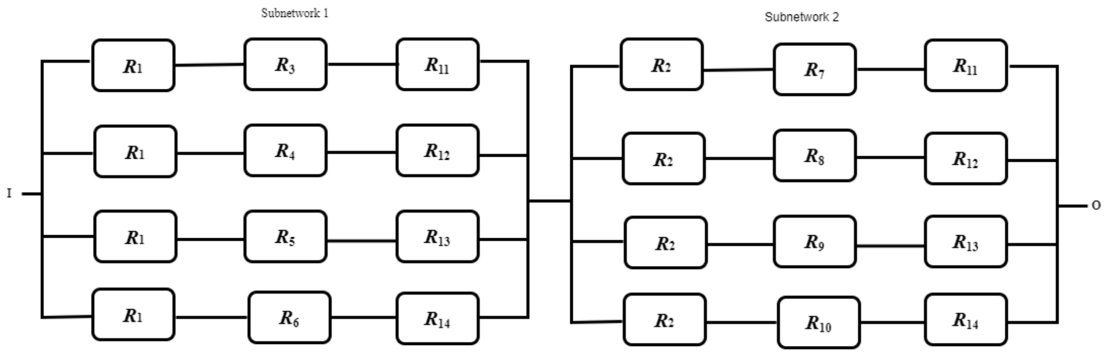


FIGURE 4. Shoutdown block diagram

The minimal path sets for the first sub network are

$$\{R_1,R_3,R_{11}\},\{R_1,R_4,R_{12}\},\{R_1,R_5,R_{13}\},\{R_1,R_6,R_{14}\}.$$

and the second sub network are

$$\{R_2,R_7,R_{11}\},\{R_2,R_8,R_{12}\},\{R_2,R_9,R_{13}\},\{R_2,R_{10},R_{14}\}.$$

We find polynomial of the shutdown network by represented parallel-series system of minimal path sets [36]. As two sub networks work together [33], we will link them in a series as shown in Figure 4

$$\begin{aligned}\mathfrak{R}_{sys_1} = {} & R_1R_3R_{11}+R_1R_4R_{12}+R_1R_5R_{13}+R_1R_6R_{14}-R_1R_3R_4R_{11}R_{12}-R_1R_3R_5R_{11}R_{13}-R_1R_3R_6R_{11}R_{14}- \\ & R_1R_4R_5R_{12}R_{13}-R_1R_4R_6R_{12}R_{14}-R_1R_5R_6R_{13}R_{14}+R_1R_3R_4R_5R_{11}R_{12}R_{13}+R_1R_3R_4R_6R_{11}R_{12}R_{14} \\ & +R_1R_3R_5R_6R_{11}R_{13}R_{14}+R_1R_4R_5R_6R_{12}R_{13}R_{14}-R_1R_3R_4R_5R_6R_{11}R_{12}R_{13}R_{14}.\end{aligned} \tag{24}$$

and

$$\begin{aligned}\mathfrak{R}_{sys_2} = {} & R_2R_7R_{11}+R_2R_8R_{12}+R_2R_9R_{13}+R_2R_{10}R_{14}-R_2R_7R_8R_{11}R_{12}-R_2R_7R_9R_{11}R_{13}-R_2R_7R_{10}R_{11}R_{14}- \\ & R_2R_8R_9R_{12}R_{13}-R_2R_8R_{10}R_{12}R_{14}-R_2R_9R_{10}R_{13}R_{14}+R_2R_7R_8R_9R_{11}R_{12}R_{13}+R_2R_7R_8R_{10}R_{11}R_{12}R_{14} \\ & +R_2R_7R_9R_{10}R_{11}R_{13}R_{14}+R_2R_8R_9R_{10}R_{12}R_{13}R_{14}-R_2R_7R_8R_9R_{10}R_{11}R_{12}R_{13}R_{14}.\end{aligned} \tag{25}$$

By Equations (24) and (25) get

$$\mathfrak{R}_{sys} = \mathfrak{R}_{sys_1}.\mathfrak{R}_{sys_2}. \tag{26}$$

4. MULTI-OBJECTIVE SYSTEM RELIABILITY OPTIMIZATION

The following formula can be used to express the multi-objective problem (MOP), the mathematical model for multi-objective system reliability allocation by using objective function is written as follows [20]:

$$\begin{aligned} & max \quad \Re_{sys}(R_i), \\ & min \quad C_s(R_i) \\ & \text{subject to :} \qquad R_{min} \le \Re_{sys} \le R_{max} \\ & \qquad a \le R_i \le b \text{ , } for \quad i = 1, ..., n, \quad a, b \in [0.5, 1] \end{aligned} \tag{27}$$

where $\Re_{sys}$ is the overall system reliability, C_s is the system cost, R_{min} is its minimum allowable value, R_{max} is its maximum allowable value. R_i is the $i-th$ component reliability. In order to resolve the multi-objective system reliability optimization problem, search for Pareto-optimal solutions by the Weighted Sum Method (WSM)[37] has been used. The single-objective function (SOP) is created by converting the objective functions of Equation (27) as follows :

$$minf(R_i) = \omega_1 C_s - \omega_2 \Re_{sys} + \psi(R_i), \tag{28}$$

where ω_1 and ω_2 is the weight vector, such that $\omega_1 + \omega_2 = 1$, where $\psi(R_i)$ is the penalty function computed as follows

$$\psi(R_i) = \vartheta_1 max\{0, R_{max} - \Re_{sys}\} + \vartheta_2 max\{0, \Re_{sys} - R_{min}\}, \tag{29}$$

where ϑ_1, ϑ_2 are the penalty factor. We present the mathematical models to case studies for system configuration. Using (HBA,DMO) algorithms and calculating them in two ways one by using the penalty method with algorithms (PFMHBA, PFMDMO), and another without using the it, with the original algorithms (Place constraints within the algorithm implementation steps), used number of iterations = 1000. The findings were rounded to four decimal places in order to determine a network best reliability values, and compare results values of R_i, $\Re_{sys}$ and total cost C_s. Take $\omega_1 = \omega_2 = 0.5$, and use parameters for HBA algorithm are $\beta = 6$, $c = 2$ and for DMO algorithm are bs=4, number of scouts=2, peep=4, performed on the MATLAB 2018.

4.1 MATHEMATICAL MODEL I

$$\begin{aligned} & max \quad \Re_{sys}(R_i), \quad for \quad i = 1, ..., 14 \\ & min \quad C_s(R_i) = \sum_{i=1}^{14} a_i (tan(\frac{\pi}{2}) R_i)^{\kappa_i}, \quad for \quad i = 1, ..., 14 \\ & \text{subject to : } 0.95 \le \Re_{sys} \le 0.9999 \\ & \qquad 0.75 \le R_i \le 0.9999, \quad for \quad i = 1, ..., 14 \end{aligned} \tag{30}$$

maximize the reliability network $\Re_{sys}$ represented in Equation (26), and $a_i = 0.003$, $\kappa_i = 2$ for all i.

Table 1. Comparison for results values of algorithms for model I

Components No	algorithm			
	HBA	PFMHBA	DMO	PFMDMO
Component R_1	0.9445	0.9557	0.9491	0.9554
Component R_2	0.9447	0.9557	0.9453	0.9557
Component R_3	0.8828	0.9089	0.8723	0.9088
Component R_4	0.7705	0.7914	0.8065	0.7857
Component R_5	0.8781	0.9054	0.8932	0.9048
Component R_6	0.808	0.821	0.7836	0.8152
Component R_7	0.8811	0.9073	0.8823	0.9059
Component R_8	0.7517	0.7639	0.8173	0.7773
Component R_9	0.8774	0.9046	0.8392	0.9071
Component R_{10}	0.8836	0.9072	0.8489	0.9098
Component R_{11}	0.8887	0.9118	0.8948	0.917
Component R_{12}	0.8002	0.8165	0.8334	0.8123
Component R_{13}	0.7500	0.7500	0.7501	0.7501
Component R_{14}	0.8853	0.9098	0.8766	0.9132
$\Re_{sys}$	0.9500	0.9932	0.9502	0.9941
C_s	1.5076	2.3617	1.5643	2.3883
time-run	8.977251	2.736140	407.616973	20.021530

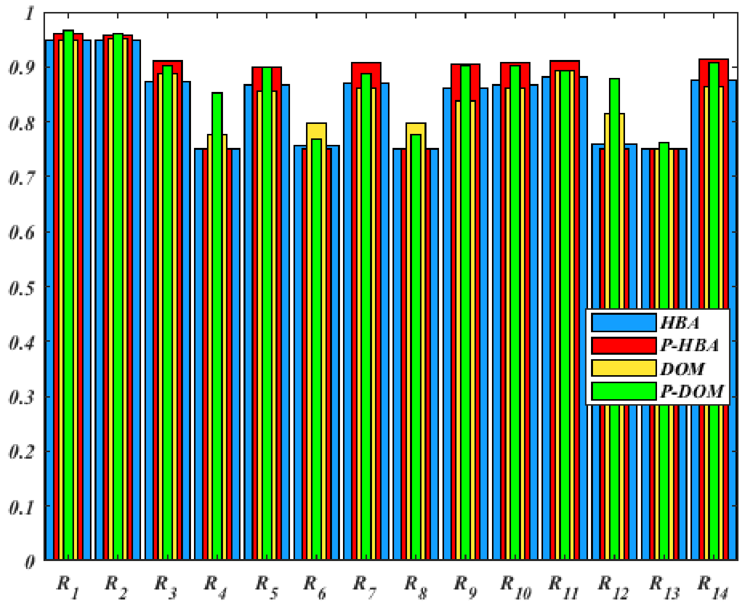


FIGURE 5. Comparison results values R_i of algorithms for model I

As illustrated in the Table (1) and the Figure (5), we have the results for values PFMHBA and PFMDMO algorithms, results have improved reliability shutdown network, and values of $\Re_{sys}$ were $(0.9500, 0.9502)$ by using (HBA,DMO) algorithms respectively while values were($0.9932, 0.9941$) by(PFMHBA , PFMDMO) algorithms. Values R_i of component between $0.7500 \leq R_i \leq 0.9445$ and $0.7500 \leq R_i \leq 0.9557$ for the (HBA,PFMHBA) algorithms respectively. Values R_i of component between $0.7501 \leq R_i \leq 0.9491$ and $0.7501 \leq R_i \leq 0.9554$ for the (DMO,PFMDMO) algorithms respectively. Total cost for best values reliability $C_s = 2.3617$ and 2.3883 for (PFMHBA , PFMDMO) algorithms. Execution time for PFMHBA and PFMDMO algorithms were less than execution time (HBA,DMO) algorithms where the time was 8.977251 and 407.616973 for (HBA,DMO) algorithm, while the time was 2.736140 for PFMHBA algorithm and 20.021530 for the PFMDMO algorithm.

4.2 MATHEMATICAL MODEL II

$$\begin{aligned}
&max \quad \Re_{sys}(R_i), \quad for \quad i = 1, ..., 14 \\
&min \quad C_s(R_i) = \sum_{i=1}^{14} a_i \exp(\frac{b}{1-R_i}), \quad for \quad i = 1, ..., 14 \\
&\text{subject to} : 0.95 \leq \Re_{sys} \leq 0.9999 \\
&\qquad\qquad 0.75 \leq R_i \leq 0.9999, \quad for \quad i = 1, ..., 14
\end{aligned} \tag{31}$$

and $\Re_{sys}$ represented in Equation (26), $a_i = 0.1$ and $b = 0.03$ for all i.

Table 2. Comparison for results values of algorithms for model I

Components No	algorithm			
	HBA	PFMHBA	DMO	PFMDMO
Component R_1	0.9499	0.9614	0.9485	0.9499
Component R_2	0.9499	0.9588	0.9519	0.9499
Component R_3	0.8745	0.9124	0.8884	0.8745
Component R_4	0.7500	0.7500	0.7782	0.7500
Component R_5	0.8663	0.8998	0.8569	0.8663
Component R_6	0.7556	0.7500	0.7978	0.7556
Component R_7	0.869	0.9074	0.8601	0.8690
Component R_8	0.7500	0.7500	0.7968	0.7500
Component R_9	0.8613	0.9042	0.8391	0.8613
Component R_{10}	0.8677	0.9093	0.8625	0.8677
Component R_{11}	0.8829	0.9115	0.8938	0.8829
Component R_{12}	0.7597	0.7500	0.8149	0.7597
Component R_{13}	0.7500	0.7500	0.7501	0.7636
Component R_{14}	0.8769	0.9134	0.8643	0.9080
$\Re_{sys}$	0.9500	0.9999	0.9513	0.9999
C_s	1.8128	1.9602	1.8225	1.9894
time-run	7.095848	3.114306	532.490625	22.722261

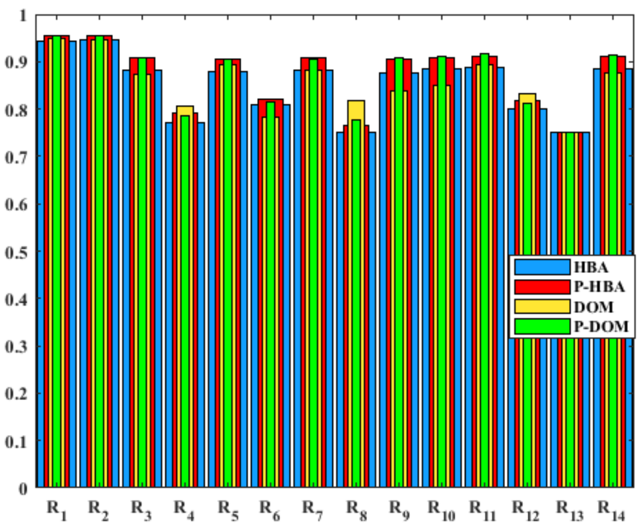
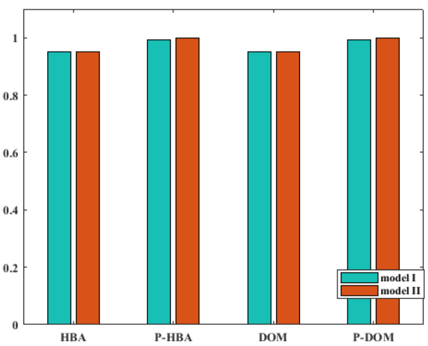


FIGURE 6. Comparison results values R_i of algorithms for model II

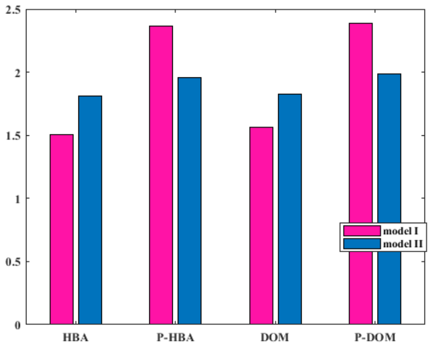
As illustrated in the Table (2) and the Figure (6), the results for values PFMHBA and PFMDMO algorithms. Results have improved reliability shutdown network; values of $\Re_{sys}$ were (0.9500, 0.9513) by using (HBA,DMO) algorithms respectively, while values were(0.9999) by(PFMHBA , PFMDMO) algorithms. Values R_i of component between $0.7500 \leq R_i \leq 0.9499$ and $0.7500 \leq R_i \leq 0.9614$ for the (HBA, PFMHBA) algorithms respectively. Values R_i of component between $0.7500 \leq R_i \leq 0.9499$ and $0.7501 \leq R_i \leq 0.9519$ for the (DMO,PFMDMO) algorithms respectively. Total cost for best values reliability $C_s = 1.9602$ and 1.9894 for (PFMHBA , PFMDMO) algorithms. Execution time for PFMHBA and PFMDMO algorithms were less than execution time for (HBA,DMO) algorithms where the time was 7.095848 and 532.490625 for (HBA,DMO) algorithm, while the time was 3.114306 for PFMHBA algorithm and 22.722261 for the PFMDMO algorithm.

5. MAIN RESULTS

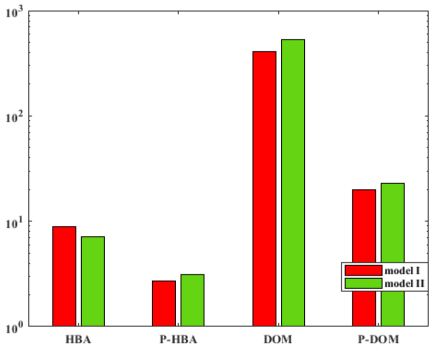
Compare results of algorithms for models I,II. Tables (1,2), and the Figure (7a) illustrates (PFMHBA and PFMDMO) algorithms improve results system reliability $\Re_{sys}$ for models. The best results of system reliability $\Re_{sys} = 0.9999$ for (PFMHBA and PFMDMO) for model II, is shown in the Figure (7c). PFMHBA and PFMDMO) algorithms able to reduce the execution time, best time to implement these algorithms is PFMHBA. The Figure (7b) shows that the lowest cost value of the models for best $\Re_{sys}$ was for the PFMHBA of the model II.



(a) Values R_i



(b) Values C_s



(c) Time of implementation

FIGURE 7. **Comparison results algorithms for two models**

6. CONCLUSION

This paper shows the directed network for shutdown system as in Figure (2). It has been created by converting the Petri net (PN), and the nonlinear reliability polynomial (26) for this network, which is simplified by splitting the network into two sub-networks as shown in Figures (3,4), which represent between two parallel- series systems. In order to study the improvement of the reliability of shutdown system, we applied (HBA,PFMHBA, DMO, and PFMDMO) algorithms to solve of multi-objective nonlinear problems. Reliability optimization problem has been addressed by using the WSM to the MOP single-objective problems SOP. The constraints handling was done via PF method. Two mathematical models I, II were used to calculate reliability and cost. The results shown by the comparison between the two cases (by using PF and without it) made it clear to us that PFMHBA is the best technique to improve the reliability of a given system than other techniques as shown in Tables (1,2). The second advantage of this technique includes the execution time of the algorithm, as shown in Figure (7c). From the point of view of industries related to aerospace, medicine, military industries, ... etc. The issue of cost increase can be considered a secondary factor in the case of increasing reliability, and this ensures that the comparison results are practically acceptable.

REFERENCES

[1] B. Dengiz, F. Altiparmak, and A. E. Smith, "Local search genetic algorithm for optimal design of reliable networks," *IEEE transactions on Evolutionary Computation*, vol. 1, no. 3, pp. 179–188, 1997.

[2] H. K. Sulaiman, F. H. Ali, and Z. A. H. Hassan, "Computational models for allocation and optimization of reliability for ross network," in *AIP Conference Proceedings*, vol. 2591, AIP Publishing, 2023.

[3] W. Kuo and M. J. Zuo, *Optimal reliability modeling: principles and applications*. John Wiley & Sons, 2003.

[4] R. A. Fadhil and Z. A. H. Hassan, "A hybrid honey-badger intelligence algorithm with nelder-mead method and its application for reliability optimization," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 4s, pp. 136–145, 2023.

[5] M. O. Ball, C. J. Colbourn, and J. S. Provan, "Network reliability," *Handbooks in operations research and management science*, vol. 7, 1995.

[6] S. A. Abed, M. S. Fiadh, and A. H. Ali, "Reliability allocation and optimization problem for waste treatment plant (wtp)," *Eurasian Research Bulletin*, vol. 5, pp. 6–13, 2022.

[7] S.-G. Chen and Y.-K. Lin, "Search for all minimal paths in a general large flow network," *IEEE Transactions on Reliability*, vol. 61, no. 4, pp. 949–956, 2012.

[8] K. Housni, "An efficient algorithm for enumerating all minimal paths of a graph," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 1, 2019.

[9] G. Jasmon and K. Foong, "A method for evaluating all the minimal cuts of a graph," *IEEE transactions on reliability*, vol. 36, no. 5, pp. 539–545, 1987.

[10] M. O. Locks, *Inverting and minimalizing path sets and cut sets*, vol. 27. IEEE, 1978.

[11] M.-L. Rebaiaia and D. Ait-Kadi, "A new technique for generating minimal cut sets in nontrivial network," *AASRI Procedia*, vol. 5, pp. 67–76, 2013.

[12] G. V. Reklaitis, A. Ravindran, and K. M. Ragsdell, *Engineering optimization: Methods and applications*. Wiley New York, 1983.

[13] R. A. Fadhil and Z. A. H. Hassan, "Simple method to extract minimal cut sets of a direct network," *Journal of Interdisciplinary Mathematics*, vol. 26, no. 5, pp. 903–908, 2023.

[14] S. A. Abed, A. H. Ali, O. A. Mohamad, and M. Aljanabi, "Reliability allocation and optimisation by using kuhn-tucker and geometric programming for series-parallel system," *International Journal of Computer Aided Engineering and Technology*, vol. 16, no. 4, pp. 488–496, 2022.

[15] K. Aggarwal and S. Guha, "Reliability allocation in a general system with non-identical components—a practical approach," *Microelectronics Reliability*, vol. 33, no. 8, pp. 1089–1093, 1993.

[16] R. T. Marler and J. S. Arora, "Survey of multi-objective optimization methods for engineering," *Structural and multidisciplinary optimization*, vol. 26, no. 6, pp. 369–395, 2004.

[17] P. J. Fleming, R. C. Purshouse, and R. J. Lygoe, "Many-objective optimization: An engineering design perspective," in *International conference on evolutionary multi-criterion optimization*, pp. 14–32, Springer, 2005.

[18] W. Li, Q. Zeng, L. Shen, and Z. B. Zhang, "A multi-objective optimization method for system reliability allocation," in *Advanced Materials Research*, vol. 860, pp. 2766–2773, Trans Tech Publ, 2014.

[19] G. Abdullah and Z. A. H. Hassan, "Utilize an ant colony algorithm to assign reliability and minimize costs for the complex system," in *AIP Conference Proceedings*, vol. 2591, p. 050014, AIP Publishing LLC, 2023.

[20] M. A. Mellal and E. Zio, "An adaptive particle swarm optimization method for multi-objective system reliability optimization," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 233, no. 6, pp. 990–1001, 2019.

[21] D. W. Coit and A. E. Smith, "Penalty guided genetic search for reliability design optimization," *Computers & industrial engineering*, vol. 30, no. 4, pp. 895–904, 1996.

[22] R. K. Gupta, A. K. Bhunia, and D. Roy, "A ga based penalty function technique for solving constrained redundancy allocation problem of series system with interval valued reliability of components," *Journal of Computational and Applied Mathematics*, vol. 232, no. 2, pp. 275–284, 2009.

[23] G. Levitin *et al.*, *The universal generating function in reliability analysis and optimization*, vol. 6. Springer, 2005.

[24] F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, and W. Al-Atabany, "Honey badger algorithm: New metaheuristic algorithm for solving optimization problems," *Mathematics and Computers in Simulation*, vol. 192, pp. 84–110, 2022.

[25] J. Ma, S. Yu, and W. Cheng, "Composite fault diagnosis of rolling bearing based on chaotic honey badger algorithm optimizing vmd and elm," *Machines*, vol. 10, no. 6, p. 469, 2022.

[26] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Dwarf mongoose optimization algorithm," *Computer methods in applied mechanics and engineering*, vol. 391, p. 114570, 2022.

[27] K. Mehmood, N. I. Chaudhary, Z. A. Khan, K. M. Cheema, M. A. Z. Raja, A. H. Milyani, and A. A. Azhari, "Dwarf mongoose optimization metaheuristics for autoregressive exogenous model identification," *Mathematics*, vol. 10, no. 20, p. 3821, 2022.

[28] K.-H. Chang, *Design theory and methods using CAD/CAE: The computer aided engineering design series*. Academic Press, 2014.

[29] S. Koziel and X.-S. Yang, *Computational optimization, methods and algorithms*, vol. 356. Springer, 2011.

[30] R. A. Sarker and C. S. Newton, *Optimization modelling: a practical approach*. CRC press, 2007.

[31] N. Andréasson, A. Evgrafov, and M. Patriksson, *An introduction to continuous optimization: foundations and fundamental algorithms*. Courier Dover Publications, 2020.

[32] S. Nayak, *Fundamentals of Optimization Techniques with Algorithms*. Academic Press, 2020.

[33] D. Zhang, K. Zhang, L. Wang, and Q. Hong, "Reliability modeling and analysis of reactor protect system based on petri net," in *Journal of Physics: Conference Series*, vol. 1754, p. 012059, IOP Publishing, 2021.

[34] A. S. Staines, "Rewriting petri nets as directed graphs," *International Journal of Computers*, vol. 5, no. 2, pp. 289–297, 2011.

[35] J. Li and W. Liu, *Lifeline Engineering Systems*. Springer, 2021.

[36] I. Gertsbakh and Y. Shpungin, *Network reliability: a lecture course*. Springer, 2020.

[37] F. S. Lobato and V. Steffen Jr, *Multi-objective optimization problems: concepts and self-adaptive parameters with mathematical and engineering applications*. Springer, 2017.

[38] X. Xu, X. Chen, Z. Liu, J. Yang, Y. Xu, Y. Zhang, and Y. Gao, "Multi-objective reliability-based design optimization for the reducer housing of electric vehicles," *Engineering Optimization*, pp. 1–17, 2021.
[39] C. Udriste, I. Tevy, and S. A. Abed, "Problems on multivariate reliability polynomial," *Atti della Accademia Peloritana dei Pericolanti-Classe di Scienze Fisiche, Matematiche e Naturali*, vol. 95, no. 2, p. 7, 2017.
[40] F. H. Abd Alsharify and Z. A. Haddi Hassan, "Bat and grey wolf algorithms to optimize complex network reliability," in *AIP Conference Proceedings*, vol. 2591, AIP Publishing, 2023.
[41] O. Bozorg-Haddad, M. Solgi, and H. A. Loáiciga, *Meta-heuristic and evolutionary algorithms for engineering optimization*. John Wiley & Sons, 2017.
[42] G. F. M. De Souza, *Thermal power plant performance analysis*. Springer, 2012.
[43] H. Kumamoto, *Satisfying safety goals by probabilistic risk assessment*. Springer Science & Business Media, 2007.
[44] F. De Carlo, "Reliability and maintainability in operations management," *Operations Management*, vol. 1, p. 32, 2013.
[45] F. Beichelt and P. Tittmann, *Reliability and maintenance: networks and systems*. Crc Press, 2012.