Applied Soft Computing Journal 167 (2024) 112241

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A Giza Pyramids Construction metaheuristic approach based on upper bound calculation for solving the network reliability problem

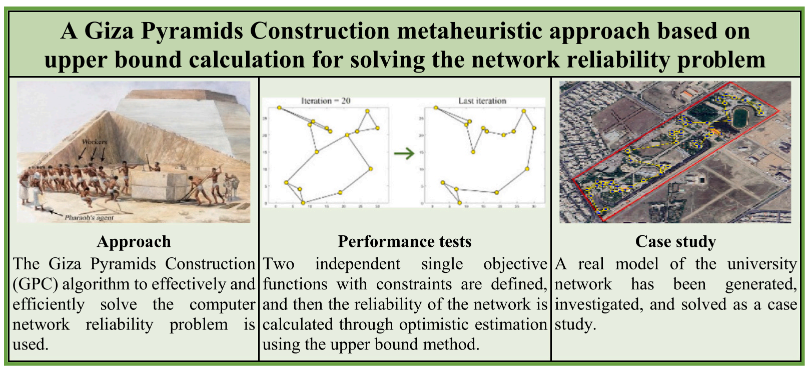
Sasan Harifi *, Amirmasoud Razavi, Melika Heydari Rad, Alireza Moradi

Department of Computer Engineering, Karaj Branch, Islamic Azad University, Karaj, Iran

HIGHLIGHTS

- This paper uses the novel Giza Pyramids Construction (GPC) algorithm to generate a network topologic with demand reliability and cost.
- The reliability of the network is calculated through optimistic estimation using the upper bound method.
- The approach is tested on randomly generated networks and compared to 10 other algorithms.
- A real model of the university network has been generated, investigated, and solved as a case study.

GRAPHICAL ABSTRACT



ARTICLE INFO

Keywords:
Optimization
Metaheuristic
Giza Pyramids Construction algorithm
Computer network reliability
Upper bound method

ABSTRACT

Network reliability optimization is an optimization problem that focuses on finding an optimal solution for a reliable network design. In network reliability optimization, the goal is to maximize network reliability so that the overall cost of the network is reduced at the same time. The discussion of the reliability of various systems in the field of industry and engineering is of great importance, that's why reliability optimization has received a lot of attention in recent decades. Since this problem is included in the category of NP-Hard problems, the use of soft computing methods will be highly effective in solving it. In this paper, an approach based on the Giza Pyramids Construction (GPC) metaheuristic algorithm is proposed to solve the network reliability problem. For this purpose, two independent single objective functions with constraints are defined, and then the reliability of the network is calculated through optimistic estimation using the upper bound method. In order to compare the performance, 12 types of diverse and complex network models have been generated and the proposed algorithm has been compared with 10 popular and state-of-the-art algorithms. Statistical analysis has been used to find significant differences in the performance of algorithms. Also, a real model of the university network has been generated, investigated, and solved as a case study. The results of experiments, statistical analysis, and observations show that the proposed algorithm has a better performance than other metaheuristic algorithms and the proposed approach in solving reliability is an effective and low-cost approach.

* Corresponding author.
E-mail address: s.harifi@kiau.ac.ir (S. Harifi).

https://doi.org/10.1016/j.asoc.2024.112241
Received 4 December 2023; Received in revised form 3 August 2024; Accepted 6 September 2024
Available online 14 September 2024

1. Introduction

Reliability is a basic feature in any modern technology system for the safe and correct operation of that system [1]. In other words, reliability is the probability of equipment or processes working correctly in a certain period and under specified and declared conditions. Reliability is a basic concept that is used in many fields to ensure correct functioning and stability of performance in various systems, processes, and technologies. The problem of reliability plays a very important role due to its requirement in engineering and industrial applications. In the fields of engineering, we can mention the use of reliability in mechanical engineering, electrical engineering, architectural engineering, agricultural engineering, software and network engineering, industrial engineering, chemical engineering, system engineering, nuclear engineering, robotics engineering, etc. It is also used in the medical, military, transportation, and aerospace industries.

As mentioned, one of the applications of reliability is the discussion of reliability in various types of networks. Reliability in the network means the probability of the system remaining operational and working properly after components and nodes or links fail independently with a certain probability [2]. A network is referred to as a set of Nodes and Links so that this set forwards network traffic. Reliability of the network is said to create a balance between the desire to lower the cost of building the network in operational mode and the ability of the desired network in an environment with the possibility of component failure [3].

Specifically, network reliability, in areas such as computer systems and networks, communication systems, information networks, Internet of Things, transportation and distribution systems, oil and gas systems and networks, and networks transmission and distribution of power is discussed. Also, in safety-critical network systems, reliability is an important issue. In the discussion of all-terminal reliability, areas such as ad-hoc wireless networks, grid computing networks, and telecommunications are investigated [4–6]. Fig. 1 shows the classification of network reliability applications. Although this category may not be unique and sometimes there may be overlap between some subcategories.

Computer network reliability refers to the ability of a computer network to function correctly and continuously, without failure or interruption, over a specified period. This is an important aspect of computer network design and operation because a computer network failure can lead to significant problems, data loss, and other negative consequences. From another point of view, computer network reliability refers to the availability of the network at the user level. Obviously, the higher the degree of reliability, the more reliable the network.

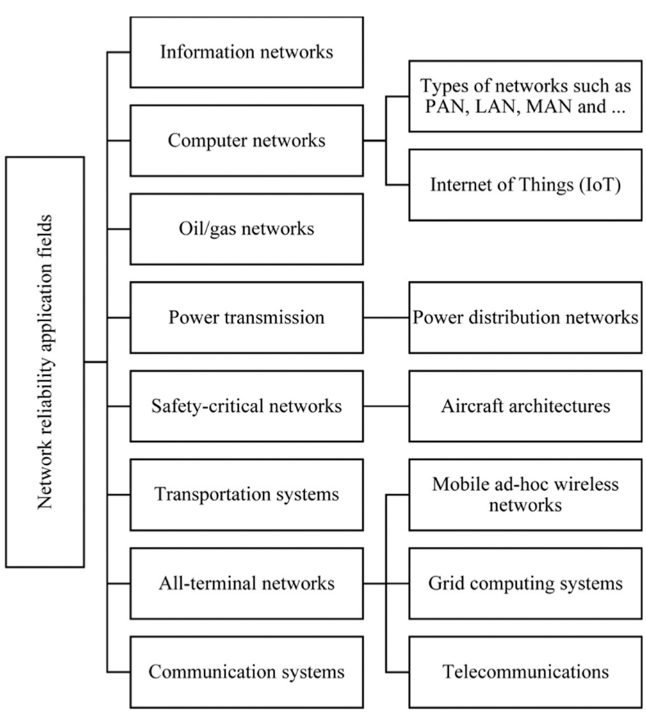


Fig. 1. Classification of network reliability applications.

Reliability optimization is an optimization problem that focuses on finding an optimal design solution. The objective function of this problem seeks to maximize reliability while minimizing system cost. In network reliability optimization, the objective function seeks to maximize network reliability while satisfying constraints such as total network cost [8]. Reliability optimization has received much attention in recent decades; because the discussion of the reliability of various systems in the field of industry and engineering is very important. Its purpose has generally been to improve the reliability of product components or the manufacturer's system. Also, having a reliable system, in general, is important for manufacturers and users because logistical decisions such as preventive maintenance, warranty policies, and spare parts management are made based on it [9,10].

Reliability optimization is used in various scientific and engineering fields to improve the efficiency and reliability of systems and processes by better managing random factors and uncertainty [8]. Network topology optimization is one of the key types of reliability optimization that is done through strategies such as adding redundancy through parallel paths and improving overall connectivity, optimizing redundant capacity by allocating backup components and capacity, optimizing network maintenance through preventive maintenance and scheduled maintenance checks, optimizing protocols and architectures to create failure-resistant networks, constrained optimization by adding reliability requirements to traditional models, simulation-based optimization to deal with possible failures, and so on. Another key type of reliability optimization is the use of specific techniques including the addition of redundant links/nodes, redundant capacity placement optimization, maintenance planning processes, joint cost-reliability optimization in capacity planning, and reliability-aware network protocols. The overall goal is to increase availability and guarantee connectivity in the face of disruptions by taking advantage of redundancy, timely repairs, resilience, and fault tolerance across the network [11].

There are many challenges in reliability optimization problems [12,13]. Uncertainty in data and parameters is one of the main challenges. In many cases, the exact values of the parameters are not fully known and there is a possibility of their change. This uncertainty can have a direct impact on the optimization results. Computational complexity can be introduced as another challenge in reliability optimization. Reliability optimization usually has complex calculations that require the use of complex optimization algorithms. Using appropriate algorithms to solve these problems requires the ability to deeply understand the problem and use powerful optimization techniques. Another challenge is balancing different goals. Because reliability optimization is usually done with multiple goals that may conflict with each other, we must seek to find a balance between these goals. For example, increased reliability may be accompanied by decreased efficiency or increased costs. To deal with the problem, we need careful analysis and rational decision-making to achieve a balance between goals. In addition, the large dimensions of the problems are also a challenge in reliability optimization. Usually, in real problems, the number of variables and constraints increases greatly, which makes the problem more complicated. This issue is challenged by the need to use advanced algorithms and techniques to solve large-scale problems. Finally, improvement and progress in reliability optimization require a precise and appropriate combination of different techniques, algorithms, and methods to manage the raised problems well and achieve effective solutions.

As mentioned earlier, many approaches have been proposed for the network reliability optimization problem. These approaches mostly solve the problem using artificial intelligence or Monte Carlo simulation.

Even, the use of neural networks to calculate reliability has been suggested in many studies. Also, the use of metaheuristic methods has been used many times. Methods such as Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) were used more than others. The network reliability problem is an NP-hard problem, so meta-heuristic approaches can be extremely effective to solve it.

In this paper, we used the Giza Pyramids Construction (GPC) algorithm to effectively and efficiently solve the computer network reliability problem. This algorithm is controlled by the movement of the workers to push the stone blocks on the ramp. For this purpose, we proposed a binary discrete version of the algorithm. The main contribution of this paper is to create diverse, different, and challenging sets of different computer networks and also to create a real model of a computer network, and use a new algorithm based on GPC to reduce computing costs and increase the speed of reaching the desired solutions. The goal of this study is to reach the optimal values to minimize the cost and also keep the reliability of the network high according to the defined rules and constraints.

The remainder of this paper is structured as follows: Section 2 introduces literature review. Section 3 describes problem description and formulation. Section 4 includes solution methodology. Section 5 is experimental results. Section 6 provides the statistical analysis. Section 7 presents application in solving a case study. Section 8 discusses limitations. Finally, Section 9 represents discussion and conclusions.

2. Literature review

In the literature, there are many studies in the field of network and reliability. Yang [14] proposed a method to analyze the reliability of computer networks in cloud computing. The author has proposed an analytical approach based on intelligent cloud computing, which includes choosing a suitable network platform and using a random key generator to generate session keys for data protection and reconstruction. The results of the experiments show that the proposed method has more reliability compared to the traditional methods. The reliability of the proposed method is very suitable, which shows its applicability and feasibility for analyzing the reliability of computer networks. According to the author, this research is a cornerstone for further research and application in this field. Moazzeni et al. [15] proposed a Reliable Distributed SDN (RDSDN) method to increase the reliability of Software-Defined Networks (SDN). The traditional centralized controller in SDNs is only one point of failure so when it fails, the whole system is disrupted. Therefore, the authors explored a distributed controller architecture as a solution. They introduced a network partitioning strategy with small subnets, each with a master controller and a slave controller for recovery purposes. They also presented a new formula to calculate the reliability rate of each subnet based on factors such as load, number and degree of nodes, and link loss rate. Their proposed RDSDN method detects controller failures and implements fast recovery schemes. The results of their experiments show that RDSDN improves failure recovery compared to existing methods in failure recovery time. They showed that performance, reliability, packet loss, latency, and network throughput are effectively improved by employing RDSDN.

Schäfer et al. [16] calculated reliability in non-series-parallel network systems that are common in critical safety engineering. The traditional method using the inclusion-exclusion principle leads to complex and computationally expensive calculations. They presented a new formula that reduces the computational cost significantly. The authors showed that the proposed formula has an exponential complexity with a linear power instead of a double exponential complexity. The results of their experiments showed that the proposed formula not only improves efficiency but also it enables the formulation of optimization problems for complex network systems based on accurate reliability calculations. Bai et al. [17] presented an improved algorithm for the reliability assessment of multi-mode networks using state space decomposition and minimal path vectors. The algorithm recursively decomposes the set of minimal path vectors so that it considers only qualified minimal path vectors from the previous set of uncertain states. They have also proposed an improved heuristic rule to select a suitable minimal path vector for decomposition. The efficiency of the algorithm has been evaluated through hypothetical networks and networks with known structures. The results showed that their proposed algorithm performs better than the existing algorithms based on state space decomposition methods and the recursive sum of disjoint products.

Kroese et al. [18] proposed a new approach based on the cross-entropy algorithm to solve the network planning problem. This problem is a combinatorial optimization problem that involves selecting the most reliable links in a network given a fixed budget. Their proposed approach can handle the limitations and noise introduced when estimating reliability through simulation. In their study, they review techniques for noise reduction and explain how the deterministic version of the cross-entropy algorithm can be modified to handle the noise situation. The results of their experiments show that the noisy cross-entropy algorithm can be as effective and reliable as the deterministic algorithm. This paper also proposed two techniques to speed up the convergence of the algorithm. These two techniques are the removal of possible redundant links and hybrid cross-entropy.

In some studies, authors used neural networks to calculate reliability. Solanki et al. [19] in their research focused on the problem of reliability in computer networks, especially in wireless networks. The authors believe that with the rapid expansion of communication networks, reliability has become a challenge. They also state that existing research in wired networks is insufficient to address the specific issues of wireless networks. To deal with the reliability problem, they introduced a new algorithm to calculate network reliability. This algorithm considers the flexibility of networks. The goal has been to improve the reliability of large and complex networks. Their proposed algorithm is more efficient than traditional methods such as Monte Carlo simulation and other reliability algorithms. In addition to the reliability calculation algorithm, they also proposed a method for reliability evaluation based on network reduction and Artificial Neural Networks (ANN). Their proposed approach speeds up the estimation process for networks of different sizes. The authors showed that their optimized ANN performs well in network reliability estimation for networks of a certain size. They also showed that optimized ANN can accurately estimate the reliability of the network for different sizes in the training set.

Davila-Frias et al. [20] proposed a framework to estimate the reliability of an all-terminal network by considering the exhaustion and failure probability of links and nodes over time. Mathematically, the reliability of all-terminal can be calculated as the product of the reliability of individual terminals and the probability that network paths are operational. This calculation involves considering different failure scenarios, such as the failure of individual terminals or the failure of specific connections in the network. Unlike the previous methods, the framework proposed by them uses Bayesian methods to estimate the reliability of links and nodes based on data exhaustion and loss. To speed up this estimation, a combination of Monte Carlo simulation and Deep Neural Networks (DNN) have been used, while DNN perform much better than traditional ANN. Their proposed framework examines the dynamic behavior of the network and provides appropriate estimates for network reliability. They claim that their approach is applicable to real-world topologies. Hassan et al. [21] proposed the use of an ANN as a predictive model to evaluate the reliability of flow networks in engineering problems. Assessing network reliability is a challenging task due to its computational complexity, especially when considering the allocation of components and their capacities. Their developed ANN model uses the maximum capacity of network components as input and predicts network reliability as output. They also recommended the integration of the ANN approach with existing optimization algorithms for efficient component allocation and computational reduction. They claim that their approach enables significant computational savings.

Kundu and Garg [22] introduced an improved neural network algorithm to solve the reliability-redundancy allocation problem with nonlinear resource constraints. They solved the problem of inappropriate exploitation of the neural network by using a novel logarithmic spiral search operator and the Teaching Learning-Based Optimization (TLBO) algorithm. The performance of their improved neural network algorithm has been evaluated on several reliability optimization problems and compared with other existing metaheuristic algorithms. The results of experiments show that the approach presented by them has performed better than other competing algorithms. They believe that their approach can be expanded. They also suggested that their approach can be suitable for improving reliability with other metaheuristic algorithms. The problem of calculating the reliability of the all-terminal network has been discussed by Srivaree-Ratana et al. [23]. The authors have proposed an alternative approach to network reliability estimation using ANN prediction models. ANN models are built, trained, and validated using network topologies, link reliability, and an upper bound on network reliability. They employed a hierarchical approach, with a general ANN screening all network topologies and a specialized ANN used for highly reliable network designs. The results of their investigation show that their proposed approach provides more accurate estimates compared to the upper-bound method, especially in worst-case scenarios. They also showed that their approach outperforms backtracking and Monte Carlo simulation methods in terms of computational efficiency.

Metaheuristic methods have been used in many researches. Lyu and Yin [24] investigated the issue of high reliability in Internet of Things (IoT) systems. They also focused on transmission and network reliability in complex environments. The authors proposed an Artificial Bee Colony (ABC) algorithm to analyze the shortest path for each cluster head node. This approach has improved data collection efficiency, energy balance, and network reliability, and extended the network life cycle. The simulation results performed by them show that their proposed algorithm reduces data transmission and energy consumption and increases the network life cycle. This study also analyzed the impact of communication link changes on network capacity under mobile nodes, with an emphasis on end-to-end reliability and node connectivity. An improved method for calculating node navigation and reliability is also proposed, and the importance of node connectivity for mobile network reliability is demonstrated. Garg [25] addressed the reliability redundancy assignment problems in series-parallel systems using the penalty-guided Biogeography Based Optimization (BBO) approach. The study aimed to determine the number of additional components and their corresponding reliability in each subsystem to maximize the overall reliability of the system. The proposed method incorporated a parameter-free penalty function to encourage the exploration of feasible and near-feasible regions while avoiding infeasible solutions. The author concluded that penalty-guided biogeography-based optimization is a promising tool for solving reliability-redundancy optimization problems, with potential applications in various systems and real decision-making scenarios.

One of the ways to achieve the stability of computer networks is to determine the optimal allocation of redundancy in such a way that the reliability of the system is maximized. For this purpose, Yeh and Fiondella [26] have proposed a redundancy optimization approach using the Simulated Annealing (SA) algorithm. Kumari and Sahana [27] combined Ant Colony Optimization (ACO) and Particle Swarm Optimization (PSO) algorithms to optimize multicast quality of service in MANET networks. They showed that their approach has acceptable performance in finding network paths and reliability. Guleria and Verma [28] tried to reduce the number of message transmissions in a wireless sensor network using a clustering-based ACO method. Their approach reduces the energy consumption of nodes and increases reliability. Chahal and Harit [29] used the PSO algorithm to select suitable paths in the VANET network. According to them, the more optimal the selected routes are, the more focus can be placed on reducing costs and increasing reliability in the network.

In other studies, Guo [30] proposed the use of the PSO algorithm in combination with a neural network to improve intrusion detection and increase the reliability of the network. Abdullah and Hassan [31] also proposed the PSO algorithm to optimize reliability in a complex network. Li et al. [32] used the PSO algorithm to optimize the reliability problem. Yeh et al. [33] addressed the challenge of solving network reliability optimization problems. While most existing methods focus on simple structured networks, modern networks are more complex, which makes it impossible to calculate the exact reliability function of the network using traditional analytical methods in a limited time. To overcome this problem, they proposed a new approach called MCS-PSO, which is a combination of Particle Swarm Optimization (PSO) with Monte Carlo Simulation (MCS). The objective of MCS-PSO is to minimize the cost while addressing the reliability constraints. The results of their experiments show that the MCS-PSO approach is more efficient and provides solutions closer to the exact solution compared to the previous methods. They believe that their approach can be extended to solve more complex reliability optimization problems in the future. Watcharasitthiwat and Wardkein [34] proposed the improved ACO algorithm to solve the network topology optimization problem so that the cost is low but the reliability is acceptable. They compared their proposed algorithm with GA and TS. The results of their experiments showed that their improved algorithm based on ACO has good quality solutions and computational efficiency compared to competing algorithms.

Genetic Algorithm (GA) has been used independently in many studies. Hamed et al. [35] investigated the robust design problem in capacitated flow networks. They considered computer networks and power transmission networks as capacitive flow networks. The goal was to determine the minimum allocation capacity for each edge to ensure the survival of the network even in case of edge failure. They mathematically formulated the robust design problem, which is an NP-hard problem, and proposed a GA to find the optimal solution. The algorithm successfully achieves maximum reliability. Ai et al. [36] used GA in optimizing topologies of computer communication networks based on maximizing reliability while adhering to cost constraints. The approach presented by them has shown acceptable performance in solving complex problems of computer work environments. Liu [37] believes that Re-Optimization of the current network is the best possible suggestion for most networks to reduce costs. He also showed that for network optimization, optimization based on GA and for multi-objective optimization can be more suitable than other approaches. Liu et al. [73] proposed an effective Traffic Signal Control (TSC) method based on genetic programming (GP) to change the phase of traffic signals in urban transportation networks. This method was presented to improve the efficiency of traffic flow in urban transportation networks.

To optimize the reliability of the network, Lin et al. [38] proposed the GA in such a way that the reliability is measured based on the minimum path and state-space analysis. The results of their experiments showed that the proposed algorithm can be executed in a reasonable time and has better computational efficiency. They compared their method with PSO, ACO, and Tabu Search (TS) algorithms and obtained the best solution with their method in a reasonable time. Ozkan et al. [39] proposed two GABB and SABB algorithms to design a reliable communication network with minimum cost, which is a combination of GA with Branch and Bound (B&B) method and SA with B&B. According to them, these two algorithms have more advantages than the standard GA and SA algorithms. The results of the experiments conducted by them have also shown that the combination of metaheuristics with B&B is a very effective approach for network design with reliability or finding solutions for existing network problems. Du and Zhang [40] also investigated the GA to solve the reliability problem in the network. Huang [41] investigated the effect of improving network reliability using the GA algorithm. The author also compared his proposed GA algorithm with traditional methods. The results of the experiments showed that GA can have a more positive effect on improving network reliability. Xiong [42] also recently proposed the GA in a comprehensive evaluation of computer network reliable index systems.

In other researches, Won et al. [43] focused on the challenge of designing a cost-effective and reliable network for various engineering applications. The authors provided an overview of recent research efforts in dealing with the reliable network design problem and developed a new hybrid heuristic algorithm that combines the GA and local search ACO. The experiments conducted by them showed the effectiveness and efficiency of the proposed hybrid heuristic algorithm. Ramirez-Marquez and Rocco [44] presented an algorithm that can be easily applied to solve all-terminal network reliability assignment problems. Their algorithm is based on two main steps that use a probabilistic solution discovery approach and Monte Carlo simulation to generate near-optimal network designs. The approach presented by them has similarities with GA and ACO. Its main difference from GA is that the solutions do not evolve. In the algorithm, Solution Space areas are checked based on the suitability of the generated solutions, and as the algorithm continues, the checked areas offer better solutions. Liu and Iwamura [45] presented a GA based on random simulation as a solution to reduce the cost and reliability of the network. Dehghani and Dashti [46] also used the GA to calculate the reliability of the power distribution network. Lin and Yeh [47] investigated the use of GA to optimize the reliability of a computer network.

In other works, Adetunji et al. [48] presented a comprehensive review of metaheuristic techniques for optimal integration of electrical units in distribution networks considering different stages of the optimization process. Amohadi and Firuzabad [49] presented a new method to determine the optimal number and locations of automatic switches and circuit breakers in power distribution networks. For this, they defined an objective function in which investment costs, switch maintenance, and reliability coefficients are considered. They optimized this objective function using the PSO algorithm. Kahouli et al. [50] presented an optimal method to optimize the network configuration in a power distribution system to increase reliability and reduce power losses. For this, they used genetic algorithm (GA) and particle swarm optimization (PSO). The results of their experiments showed that the optimization of the distribution network topology using the PSO approach has significantly helped to improve reliability. Memari et al. [51] proposed the idea of using clustering algorithms to evaluate reliability in the network. They examined classical clustering algorithms such as k-means, fuzzy c-means, and k-medoids in their evaluation method. According to their experiments, using fuzzy c-means had better accuracy in network reliability evaluation. Shan et al. [52] proposed the use of a Deep Belief Network (DBN) simulation model to improve the reliability analysis performance of power distribution networks. They also optimized DBN parameters using the PSO algorithm. The results of their experiments show that their approach is accurate and efficient. Tolson et al. [53] presented an approach using the GA for reliability-based optimization of water distribution networks. Xu et al. [74] propose a framework that uses a neural network to predict or balance the network. The main goal was to balance and choose an effective route in the traffic network for self-driving cars. Ibraheem et al. [75] detected anomaly in encrypted network traffic using machine learning and compared the performance of different feature selection techniques in network traffic.

3. Problem description and formulation

3.1. Reliability

Reliability is considered in all branches of engineering. Because the job of an engineer is to create a system or a device that needs to operate correctly. It is also necessary to maximize the time and duration of correct operation. In this case, this system or device can be trusted. Trust can be quantified and conceptually expressed as reliability. Reliability can be defined as the correct operation of the system in a certain period so that certain conditions have been determined for it. A standard operating condition is considered for each system. The probability of the correct operation of the system is defined in standard conditions and is measured in a given time frame. This probability is exactly equivalent to the reliability of the system.

As stated, the probability of a device, system, or device functioning correctly in a certain period and under certain conditions can be considered as a definition of reliability. Mathematically, suppose the Probability Density Function (PDF) of failure is denoted by $f(t)$. Here $f(t)$ is denoted by t because it is relative to time. Although in some applications this variable can be distance instead of time or another variable can be used based on the application.

Consider Fig. 2. In Fig. 2, $f(t)$ of the probability of failure in the time interval $[t_1, t_2]$ is equal to the area of the highlighted part whose relation is $\int_{t_1}^{t_2} f(t)dt$. Corresponding to the probability distribution function $f(t)$, which is the PDF, there is a Cumulative Distribution Function $F(t)$, which is the CDF. The mathematical definition of CDF is,

$$F(t) = \int_{-\infty}^{t} f(\tau)d\tau \tag{1}$$

Since there is no negative time, the integral can also start from zero. In this way, if it is assumed that the random variable that represents t is T, the above equation is written as follows,

$$F(t) = \int_{0}^{t} f(\tau)d\tau \Longrightarrow \Pr\{T \leq t\} \tag{2}$$

where, as stated, T is a random variable and t is a sample. In this way, the above equation calculates the probability of failure from the beginning to the moment t. Now, the opposite is the probability of success, which is $R(t) = 1 - F(t)$. This equation can be written as follows,

$$R(t) = \int_{0}^{\infty} f(\tau)d\tau - \int_{0}^{t} f(\tau)dt = \int_{t}^{\infty} f(\tau)d\tau \tag{3}$$

System reliability is the probability that the system will fail after time t. So for the above equation, there is,

$$R(t) = \int_{t}^{\infty} f(\tau)d\tau \Longrightarrow \Pr\{T \geq t\} \tag{4}$$

But the above solution is conceptual. For many systems, it is not possible to evaluate the integral.

3.2. Reliability in computer network

There are important and widely used categories of problems that are included in the discussion of network reliability. Among these problems, we can refer to problems in the field of computers and designing computer networks, industries, management, telecommunications and designing telecommunication networks, power engineering and designing systems for power transmission or designing monitoring networks for these distribution systems.

In the discussion of network reliability, the goal is to design a network in which all nodes have direct or indirect communication with each other. In this discussion, in addition to the need to reduce the cost of network design, the reliability of the network should be increased. Of course, the reliability of the network can be defined as a constraint so that it is not less than a certain limit.

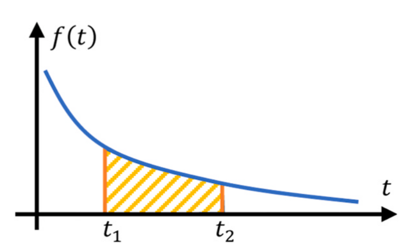


Fig. 2. Probability of failure in the period of time.

It can be said that the network reliability problem is a special and advanced type of the minimum spanning tree problem, which in addition to the connection between the nodes, in general, the reliability is also at an optimal level. In the minimum spanning tree problem, the goal is to design a tree so that all nodes are directly or indirectly connected. But in the problem of network reliability, the goal is not to design a tree, but the goal is that the network graph is connected first, which should be the default condition. Secondly, the graph must be two connected. It means that every node in the network is connected to at least two other nodes. In this way, if a communication edge of a particular node fails, there will be another edge that will keep the node connected to the network. Sometimes, a particular node may be connected to more than two other nodes, but since one of the goals is to reduce the cost, the additional connection should be examined to see if it is worth the cost or not.

Here, there is the decision variable denoted by x_{ij}. In this way, if i has a connection with j, so $x_{ij} = 1$, and if there is no connection, $x_{ij} = 0$. As it turns out, x_{ij} is a binary decision variable. It should also be considered that the lines are two-way namely $x_{ij} = x_{ji}$. While $x_{ii} = 0$ and $x_{jj} = 0$, this means that there is no connection from a node to itself. If the two-connectivity condition is also considered, there is,

$$\sum_{j} x_{ij} \geq 2 \tag{5}$$

The above explanations are structural constraints that have been created according to the stated concepts of the problem. But here it is necessary to raise the problem as an optimization problem. For this purpose, the cost of setting up the network or the objective function of this problem is defined as follows,

$$\min \sum_{i} \sum_{j \neq i} c_{ij} x_{ij} \tag{6}$$

where c is the cost of creating edge (i,j). That is, it connects node i to j. x is also an indicator that shows whether i is connected to j or not. There is also another objective function that maximizes reliability. This objective function is as follows,

$$\max R(x) \tag{7}$$

where R is independent of time and varies concerning each x. The definition of the problem and the definition of the above two objective functions can be considered a multi-objective optimization problem or can be defined as a single-objective with constraint. But in this paper, we want to convert it into an objective function with specified constraints. In other words, an epsilon constraint (ε-constraint) method is used. One of the effective methods in solving multi-objective optimization problems is the ε-constraint method, which, unlike the weighted sum method, can find non-dominated points in non-convex parts of the non-dominated boundary. Here, one of the objective functions is removed and a constraint is added to the problem instead. In this way, the problem becomes a single-objective problem. For this single objective problem, two types of objective functions can be defined and each of them can be solved independently. For the first type of objective function, there is,

$$\min \sum_{i} \sum_{j \neq i} c_{ij} x_{ij} \quad \text{s.t.} \quad R(x) \geq R_0 \tag{8}$$

where the minimization of the network setting up cost is considered as the objective function and the reliability maximization is considered as the constraint. For the second type of objective function, there is,

$$\max R(x) \quad s.t. \quad \sum_{i} \sum_{j \neq i} c_{ij} x_{ij} \leq c_0 \tag{9}$$

where reliability maximization is defined as an objective function and minimization of network setting up cost is defined as a constraint.

3.3. Calculating the reliability of a network

If G is considered a graph, then for this graph $G = (V,E)$. where V is the set of vertices and E is the set of edges. For this graph, the connectivity can be represented through the adjacency matrix. Consider Fig. 3 as an example. This Figure has five vertices and five edges. Suppose a message is to be sent from vertex number one to vertex number three. Now, according to the conditions, a path is chosen. In this way, the message cross through vertex number two. The entire path is marked in red color in Fig. 3. This crossing path is active or operational edges. Therefore, the active set is always a subset of all edges and is represented as $A \subseteq E$. For an active set, there is,

$$G = (V,E) \longrightarrow G' = (V,A) | A \subseteq E \tag{10}$$

It can be said that the reliability related to the active set is the probability of the correct operation of all the edges that are in set A. If p is the probability of the correct operation of the edges and $q = 1 - p$ is the probability of the incorrect operation of the edges, then for set A, there is,

$$R(G') = R(A) = \prod_{e \in A} p_e \times \prod_{e \notin A} q_e \tag{11}$$

where in this case, inactive edges in the entire graph are considered as unreliable edges. Consider that there are different active sets, which are called the set of all possible working states and denoted by Ω. In fact, Ω is the power set of E. Thus, the reliability of a graph, if it is considered a function of x, is equal to,

$$R(E) = R(x) = \sum_{A \in \Omega} R(A) \tag{12}$$

where $R(A)$ is the Eq. (11). As a result, Eq. (12) is the method of calculating the reliability of the network. Since the possible states for the set Ω are equal to $2^{|E|}$ is, as the network becomes larger, Ω grows exponentially. In this case, its complexity is $O(2^n)$. Therefore, using the Eq. (12) for calculation in large networks becomes very complicated. One solution is to use Monte Carlo simulation. In such a way that instead of checking all possible states, several states are checked and then an approximate reliability is obtained by averaging. Another solution is to optimistically estimate the reliability of a network through the upper bound method provided by Jan [54]. According to the upper bound method, the reliability of network G is obtained through the following relationship,

$$H(d_1, d_2, \ldots, d_n) = 1 - \left[\sum_{i=1}^{n} q^{d_i} \prod_{k=1}^{m_i} \left(1 - q^{d_k - 1}\right) \prod_{k=m_i+1}^{i-1} \left(1 - q^{d_k}\right) \right] \tag{13}$$

where d is the degree sequence and $H(d_1, d_2, \ldots, d_n)$ is the upper bound of the reliability of $R(G)$. Also, $m_i = \min(d_i, i-1)$ and $i = 1, 2, \ldots, n$. In this case, the complexity is $O(n^2)$. For this reason, it is more acceptable to use the Eq. (13) than the Eq. (12). Monte Carlo simulation is a stochastic

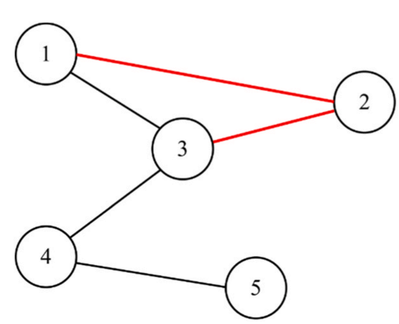


Fig. 3. An example of the selected path to send a message from node 1 to node 3 (marked in red color).

method that involves generating random samples from a probability distribution to estimate the reliability of a network. This method is useful when the network has a complex structure or when the probability distributions of the component failures are unknown. On the other hand, the upper bound method is a deterministic method that provides an upper bound on the reliability of a network. This method is useful when a quick estimate of the reliability is needed, but it may not provide an accurate estimate, especially for complex networks. In comparison, Monte Carlo simulation can provide a more accurate estimate of the reliability, but it can be computationally intensive. The upper bound method, on the other hand, is faster but may not provide an accurate estimate. According to the available computational resources and the use of the optimization method through soft computing, in this paper, the upper bound method is used to calculate network reliability.

4. Solution methodology

This section describes the Giza Pyramids Construction (GPC) algorithm as a solution methodology to solve reliability models. However, a full explanation of the algorithm with its details is available in the main GPC paper [55].

4.1. Standard GPC

Throughout history, the times when writing was used to record and narrate historical events are called ancient times. To understand that times, there are two main and important [56]. The first source is archaeologists, who try to obtain information about the life and culture of that period through the examination of physical artifacts and remnants of the past, such as objects, buildings, and sites, etc. The second source is textual sources that are used in this context. These sources include written texts that survive from antiquity, including writings, travelogues, official documents, poems, and stories, and generally surviving correspondences that give us detailed information about the events, culture, and societies of that time. Using these two sources gives us a more accurate picture of the ancient period and helps us to understand it more easily. Sources show that in ancient times, various restrictions could not prevent technological and engineering advances. A study of civilizations, their construction engineering methods, and technologies shows that these civilizations have always sought to find better and more efficient ways to live. They tried to find and develop new ways to improve the living conditions and development of their society by taking advantage of the available resources, advanced technologies, and knowledge. In general, humans throughout history have always wanted to move towards a better life and evolution by improving and revising their methods and technologies.

The existence of architectural remains from ancient times points to the fact that past civilizations have always tried to display their cultural values and technologies. The GPC algorithm drew inspiration from ancient times. In this algorithm, it is inspired by the method that was used in the process of building pyramids. This method involves moving stone blocks by workers on a ramp to build pyramids. Important and influential factors in the process of building pyramids include the management of workers, the use of ramps, slopes, and friction. In this algorithm, a worker who works in the best and most efficient manner can achieve the position of Pharaoh's special agent, and other workers are compared with this special agent. The special agent is chosen as a role model for other workers. There is always competition to achieve the special agent of the pharaoh, and every worker tries his best to achieve this position and become a role model for the rest of the workers. Also, workers who do not perform their duties in the best possible way may be substituted by another worker and lose their position. In this algorithm, there is a chance for all workers to reach the position of special agent. Even workers who seem weaker at first are more likely to achieve special agent because they can strengthen themselves and improve their skills. Fig. 4 represents the standard version of the GPC algorithm [57].

```
Pseudo-code of the standard Giza Pyramids Construction (GPC) Algorithm.
STEP 1:
  generate initial population of stone blocks or workers;
  generate position and cost of stone block or worker;
  determine best worker as Pharaoh's agent;
  STEP 2: for FirstIteration to MaxIteration do
    STEP 3: for i=1 to n do (all n stone blocks or workers)
      calculate amount of stone block displacement (Eq. 14);
      calculate amount of worker movement (Eq. 15);
      estimate new position (Eq. 16);
      investigate possibility of substituting workers (Eq. 17);
      determine new position and new cost;
      if new_cost < Pharaoh's agent cost then
          set new_cost as Pharaoh's agent cost;
      end if
    END STEP 3
    Sort solutions for next iteration;
  END STEP 2
END STEP 1
```

Fig. 4. Standard version of GPC algorithm.

At initial, the workers are spread out in an environment that has stone blocks. Each worker moves their respective block of stone, so both they and the stone block are moved. New solutions are obtained by moving workers in each iteration. At the end of each iteration, the worker with the best solution is selected as Pharaoh's special agent, and the rest of the workers are compared with him in the next iteration. If a worker is unable to move the stone block in the desirable shape (no desirable solution is achieved) it may be substituted with another worker. Here, considering that the workers push the stone on an inclined surface to reach the place of installation in the pyramid, the equations of motion on the inclined surface or ramp are used.

In this algorithm, assumptions have been determined for the movement of stone blocks, according to which, the blocks move upwards on the ramp with an initial speed of zero v_0. According to this hypothesis, after traversing a distance, the stone block stops. Fig. 5 shows the force acting on the stone block.

According to the conditions of the algorithm, a time-independent equation is required for movement with constant acceleration. As a result, the amount of distance that the stone block traverses on the ramp, is obtained through the following equation,

$$d = \frac{v_0^2}{2g(\sin\theta + \mu_k \cos\theta)} \tag{14}$$

where d represents the distance traversed by the stone block on the ramp, which is obtained after calculations through the above equation. v_0 is the initial speed of the stone block, which is determined using a random function as a random number between zero and one. g is gravity (it is 9.816). θ shows the angle of the slope of the ramp relative to the horizon. μ_k is the coefficient of kinetic friction between the stone block and the ramp, which is determined from a uniform random distribution between two predetermined minimum friction and maximum friction values.

To increase the worker's mastery while pushing the stone blocks, the worker performs certain movements. This can lead to better control,

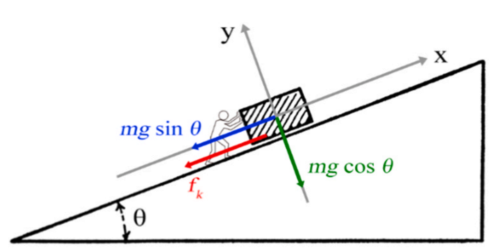


Fig. 5. Force acting on the stone block [55].

more mastery, and ultimately non-repetitive movements when pushing the stone block. Through the following equation, the amount of movement of the worker when pushing the block is shown,

$$x = \frac{v_0^2}{2g\sin\theta} \tag{15}$$

where this equation is the same as the Eq. (14), with the difference that friction is not considered in this equation and it has no effect on the equation, so it is not considered. Understanding this type of worker movement is shown in Fig. 6. Now, by combining these two equations namely Eq. (14) and Eq. (15), we get a new equation that gives us the amount of changes in the movement of the worker and the stone block. Based on the cumulative form of the algorithm that is,

$$\overrightarrow{p} = (\overrightarrow{p}_i + d) + x\overrightarrow{\epsilon_i} \tag{16}$$

where in the above equation, p_i indicates the current position. d represents the displacement of the stone block. x gives us the movement rate of the worker from the Eq. 15. ϵ_i is also a vector of random numbers that can be uniform, normal, or Lévy distribution.

In the standard GPC algorithm [55], with a probability of fifty percent, one worker is substituted by another worker, which balances exploration and exploitation operations. In the process of building the pyramids, tired workers were regularly substituted by fresh workers. Tired workers were practically useless in furthering construction objectives, but they could be useful elsewhere. In each iteration of the algorithm, workers are constantly substituted. Suppose that the vector $\Phi = (\varphi_1, \varphi_2, \ldots, \varphi_n)$ is the solutions obtained in STEP1 of the algorithm and also $\Psi = (\psi_1, \psi_2, \ldots, \psi_n)$ is the solutions obtained through Eq. (16) which indicates the position of the workers and the block. In the substitution step, some solutions of the vector Φ are substituted with Ψ. As stated, this operation happens in the original algorithm with a probability of fifty percent. Although in this paper this possibility is increased. Finally, a vector Z is constructed which is a combination of the features of two vectors Φ and Ψ. So, vector Z is $(\zeta_1, \zeta_2, \ldots, \zeta_n)$. The substitution operation is done through the following equation.

$$\zeta_k = \begin{cases} \psi_k, & \text{if } rand[0,1] \le 0.5 \\ \varphi_k, & \text{otherwise} \end{cases} \tag{17}$$

where In this equation, ψ is the new solution created by the previous equation and the original solution is φ, and ζ also shows the new solution. In this algorithm, all members of the population move and displace, and because the algorithm has a memory, the best ones are selected for the next iteration, according to the number of members of the population.

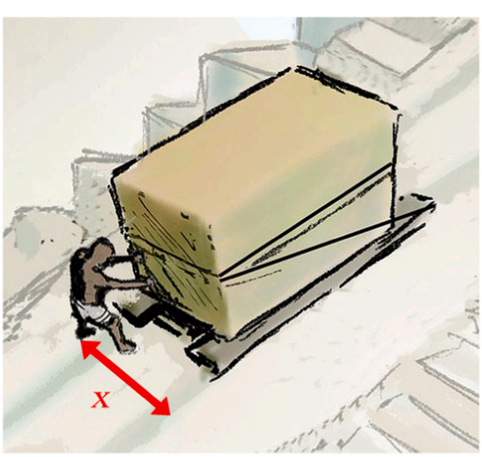


Fig. 6. Shaking of the worker along the x for better control and mastery of the stone.

4.2. Proposed modified binary GPC

As explained in Section 3, the decision variable of the raised problem in this paper is binary. In such a way that if there is a connection between the nodes, this variable is one and otherwise it is specified as zero. The standard GPC algorithm is presented for continuous problems. However, a binary version is also provided to solve the knapsack problem [58]. We applied the method used in [58] to binarize the algorithm. This is a simple and practical solution. In this way, the algorithm produces solutions continuously, but the final solution is rounded to zero and one and becomes a binary solution. For this purpose, Eq. (16) is changed to the following equation, so we have,

$$\overrightarrow{p} = \text{round}[(\overrightarrow{p}_i + d) + x\overrightarrow{\epsilon_i}] \tag{18}$$

where, the generated solutions are rounded into binary. In the standard GPC algorithm, the substitution operation is effective in establishing a balance between exploration and exploitation. The probability of substitution is assumed to be fifty percent by default. However, this probability has changed in different studies conducted by different researchers. Depending on the nature of the problem, the balance can be established by changing this probability. In this paper, we have assumed a replacement probability of eighty percent. Also, we changed the way of doing the substitution so that it works more effectively than the standard GPC. For this purpose, instead of complete substitution, some characteristics of the workers are substituted. For this operation, first, a mask is generated from the substitute candidate worker and the new worker. Then, based on the mask, a new worker with new characteristics is generated. This operation can be considered similar to binary uniform crossover in GA. Fig. 7 shows how to do the substitution.

Fig. 7 shows that a random mask is generated based on workers 1 and 2. Now, based on the mask, new workers 1 and 2 are generated. So for the new worker 1, if the bit mask is zero, the bit of worker 1 is taken, and if the bit mask is one, the bit of worker 2 is taken. In the case of new worker 2, it is exactly the opposite.

4.3. Representation scheme

As mentioned metaheuristics can be used to find the most reliable paths or configurations within a network. This is typically a complex optimization problem, as there may be many possible paths, and the reliability of each path can depend on various factors such as the condition of the connectivity and nodes, the path length, and so on. The choice of approach depends on various factors, such as the specific problem characteristics, the available computational resources, and the desired balance between exploration and exploitation.

The approach of using the GPC algorithm as a soft computing method

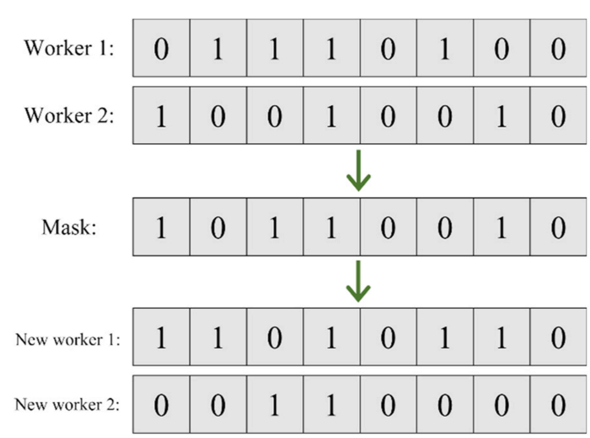


Fig. 7. An example of binary substitution similar to the binary uniform crossover operator.

to solve and calculate the reliability of the pre-generated network models used in this paper is as follows. First, the data related to a pre-generated model is loaded. Then one of the objective functions described in the previous section is considered. The initial parameters of the algorithm are set and the necessary variables are created. The variables include the population, the position of each worker, and its cost based on the model and the determined objective function. Then the main loop of the algorithm starts to optimize the solutions and records the optimal results. In this main loop, in each iteration, a new solution is created based on the described approach in subsections 4–1 and 4–2. Then the cost of the new solution is measured. In this way, connectivity and two-connectivity are checked and reliability is measured through the objective function in each iteration. The representation scheme of how to apply the algorithm to the models presented in this paper is shown in Fig. 8.

Also, a block representation [70] for the proposed method is shown in Fig. 9. A block representation, such as a block diagram, offers several advantages such as a high-level overview of the approach, simplification, decision-making in the approach, universal language, and so on. The method of applying the algorithm in this paper can be considered as a systematic approach so that it can be applied to other metaheuristics as well. This application approach is not specific to solving the network reliability optimization problem and can even be used to solve other problems.

5. Experimental results

In this section, the descriptions of the selected algorithms for comparison, parameter settings, network models, and how to create them for experiments, how to conduct experiments, and experimental results are described. It should be noted that comparative analysis of the GPC

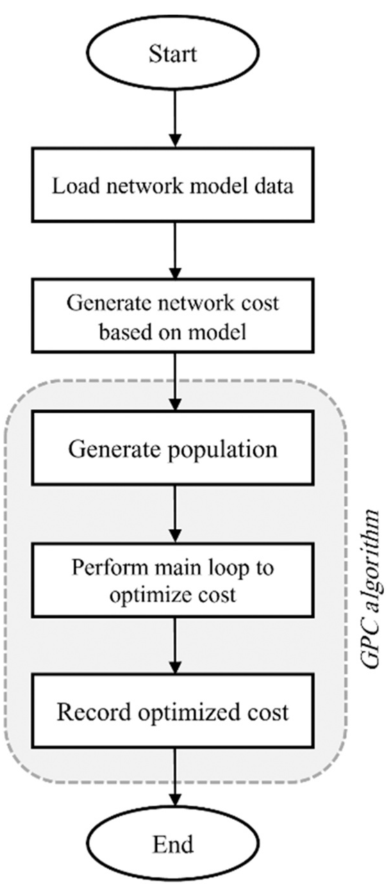


Fig. 8. Representation scheme of applying GPC on a network reliability model.

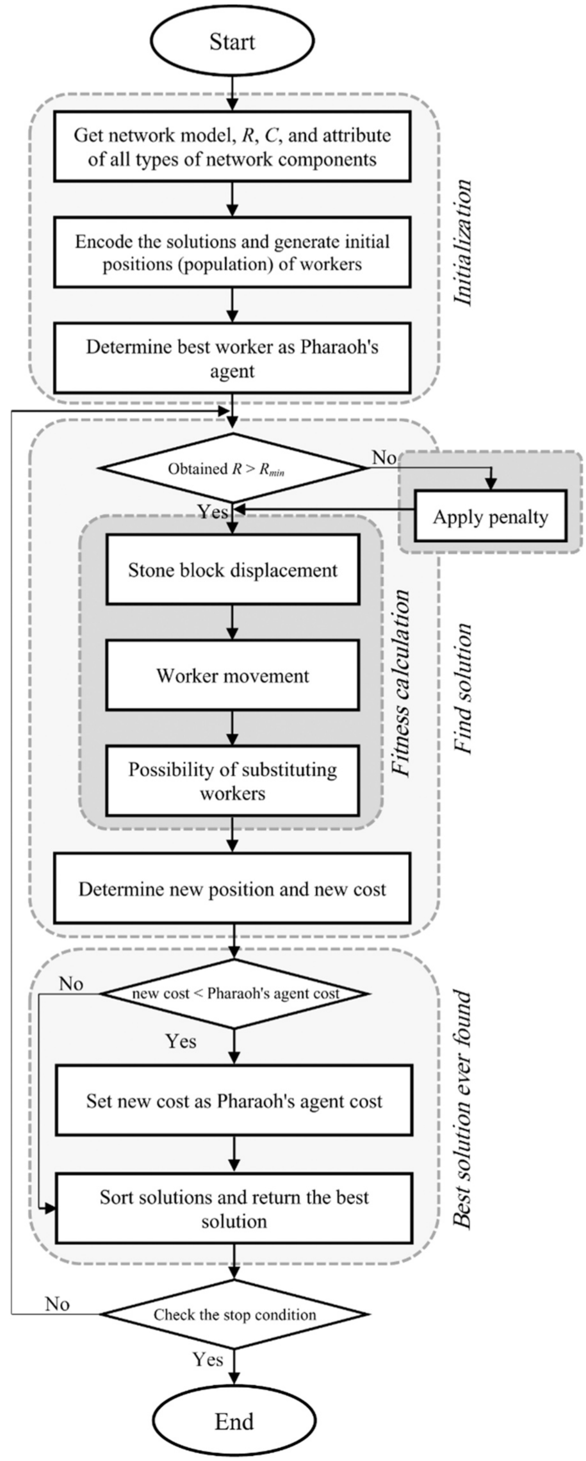


Fig. 9. Block representation of the proposed method.

performance on standard benchmark functions is done completely and in detail in [55].

5.1. Algorithms used for comparisons

In this paper, to evaluate the performance of the proposed algorithm, we selected ten popular and state-of-the-art metaheuristic algorithms. These algorithms include Artificial Bee Colony (ABC) [59], Ant Colony Optimization (ACO) [60], Biogeography Based Optimization (BBO) [61], Differential Evolution (DE) [62], Firefly Algorithm (FA) [63], Genetic Algorithm (GA) [64], Grey Wolf Optimizer (GWO) [65], Invasive Weed Optimization (IWO) [66], Simulated Annealing (SA) [67], and Teaching Learning Based Optimization (TLBO) [68]. We tried to select a variety of competing algorithms to cover metaheuristic classification. For example, we select the GA algorithm from the category of evolutionary algorithms. This algorithm has always been the choice of researchers in different experiments for solving different problems. SA algorithm is a trajectory-based algorithm. This algorithm is suitable for solving problems in which it is necessary to determine some kind of path. ACO algorithm is also a suitable algorithm for finding the path. This algorithm and the rest of the algorithms selected for comparisons are classified as nature-inspired algorithms.

5.2. Settings for algorithms

In this subsection, explanations are given about the details of the competing algorithms and how to perform the experiments. It should be mentioned that the binary version of the selected algorithms was used. To conduct fair experiments, some parameters of the algorithms, such as the number of the population, are set to be similar to each other. As the network model becomes larger, the population increases. The population is directly related to the number of decision variables. In this paper, the Decision Variables (DV) are calculated through the formula $N(N-1)/2$, where N is the number of nodes. The number of the population is always considered to be twice the value of the decision variable. Also, the other parameters of the algorithms have been obtained manually and based on trial and error so that the algorithms are in the best solution conditions according to the problem. It should be mentioned that the crossover type of the DE algorithm is binomial crossover. Also, the GA uses binary single point, binary double point, and binary uniform crossover randomly in each iteration [69].

The stopping condition of the algorithms is the number of function evaluations (NFE). This makes the algorithms to be compared under fair conditions. In this way, even the difference in the complexity of the algorithms is ineffective in the results. To record the results of the experiments, the mean and standard deviation of ten independent runs of each algorithm are considered. The settings related to the various parameters of the selected algorithms are listed in Table 1. In the table, NDV indicates the Number of Decision Variables.

It is necessary to explain that in metaheuristic algorithms, the choice of parameters and their impact on the optimization process are crucial. The rationale behind the choice of parameters in the GPC algorithm can be explained using concepts including population size, worker movement, substitution, and number of iterations [57,58].

The population size represents the number of candidate solutions in the population. As mentioned earlier, as the network model becomes larger, the population increases. A larger population size allows the algorithm to explore more solutions, increasing the chances of finding a better solution. However, a larger population size also increases the computational cost. The stone block displacement or worker movement rate is the probability that a selected worker or stone block will be moved. This rate is very dependent on settings such as the angle of the ramp, initial velocity, minimum friction, and maximum friction. A higher rate allows the algorithm to explore new solutions more frequently, potentially leading to better solutions. However, a high movement rate may also lead to premature convergence. The substitution rate is the possibility of substitution a worker with another worker. A higher substitution rate allows the algorithm to change the worker information, potentially leading to better solutions. The substitution probability in this paper is 80 %. This rate can be a different number based on trial and error and according to the type of problem. However, a high substitution rate in some problems may also lead to premature convergence. The number of iterations represents the number of times the algorithm will iterate through the population, applying the worker displacement and substitution. A higher number of iterations allows the algorithm to explore more solutions, potentially leading to better solutions. However, a higher number of iterations also increases the computational cost. Although as stated, the stopping condition of the algorithms is the NFE because of fairness for iterations.

Overall, the choice of parameters in the GPC algorithm is crucial for controlling the exploration-exploitation trade-off, the balance between exploring new solutions and exploiting the best solutions found so far. The rationale behind the choice of parameters is to strike a balance between exploration and exploitation that maximizes the chances of finding a good solution while minimizing the computational cost.

Table 1
The values used to adjust the parameters of the algorithms.

Algorithm	Parameters	Values
ABC	Colony size	NDV * 2
	Number of onlooker bees	NDV * 2
	Abandonment limit parameter	(NDV * ColonySize) / 2
	Acceleration coefficient	1
ACO	Colony size	NDV * 2
	Sample size	50
	Intensification factor	0.5
	Deviation distance ratio	1
BBO	Population size	NDV * 2
	Keep rate	0.2
	Number of kept habitats	Keep rate * PopSize
	Number of new habitats	PopSize - Kept habitats
	Immigration rates	Rand()
DE	Population size	NDV * 2
	Lower bound of scaling factor	0.2
	Upper bound of scaling factor	0.8
	Crossover probability	0.2
FA	Swarm size	NDV * 2
	Light absorption coefficient	1
	Attraction coefficient base value	2
	Mutation coefficient	0.2
	Mutation coefficient damp ratio	0.98
GA	Population size	NDV * 2
	Crossover percentage	0.7
	Mutation percentage	0.4
GWO	Search agents	NDV * 2
	Initial convergence factor	2
IWO	Population size	NDV * 2
	Maximum population size	NDV * 2
	Minimum number of seeds	0
	Maximum number of seeds	5
	Variance reduction exponent	2
	Initial standard deviation	0.5
	Final standard deviation	0.001
SA	Population size (Inner iterations)	NDV * 2
	Initial temperature	100
	Temperature damping rate	0.5
TLBO	Population size	NDV * 2
GPC	Population size	NDV * 2
	Angle of ramp	14
	Initial velocity	Rand()
	Minimum friction	1
	Maximum friction	10
	Substitution probability	0.8

5.3. Instances creation method

To evaluate the performance of the algorithms, we created our network models. It has been tried as much as possible that the created instances have variety so that it challenges the algorithms in different conditions. The number of created instances is 12. The space intended for placing nodes is a square-shaped space. For each model, nodes with x and y coordinates were placed completely randomly in different locations in the considered area. The distance of each node from other nodes is calculated through the formula $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$. The probability of correct operation is considered to be 0.9 for all models. The minimum value of reliability for models, which we denote by $R_{\min}$, is equal to 0.85. This means that we want the reliability to be no lower than 85 %. The minimum cost of the models is also 10 % of the total cost of the model if the model is considered a fully connected model. The minimum cost is denoted by $C_{\min}$. This means that the reliable network cost should not exceed $C_{\min}$. In Table 2, there are details about the created models. All evaluation experiments have been run on an Intel® CoreT^M i7–6700HQ CPU @3.40 GHz with 16 GB of DDR4 RAM. Implementations have been run on MATLAB R2023a for coding.

5.4. Final results

In this section, the results of the experiments are described. Experiments have been performed based on two different objective functions. The first objective function, namely Eq. (8), minimizes the network creation cost and takes the maximization of reliability as a constraint. In the second objective function, that is Eq. (9), the maximization of the reliability is the main objective and the minimization of network creation cost is the constraint of the function. Table 3 shows the results of the first objective function. Table 4 is also the result of the second objective function. For both objective functions, the mean and standard deviation of 10 independent runs of the algorithms are reported as the cost of the objective function. Also, the mean and standard deviation of the constraint of each objective function are reported in each table. To better understand the method of implementation and solution of the proposed approach, we executed the created model number #5 once based on the number of iterations and displayed the network graph at the start of the algorithm, iteration number 10, iteration number 20, and the last iteration in Fig. 10. This figure shows well how to solve the model by observing the connectivity and two-connectivity conditions, by the approach proposed in this paper.

As stated, Table 3 shows the results of the running of the algorithms based on the first objective function. The GPC algorithm has performed best among other competitors in 9 out of 12 models. In the two models, the GPC algorithm along with the DE and ABC algorithm jointly performed best. DE performed better than GPC in only one model. Among the 10 competing algorithms, SA and TLBO performed the worst. Due to the nature of the SA algorithm, which is a trajectory-based algorithm, it was expected that it might not provide appropriate solutions. Regarding the TLBO algorithm, it can also be said that it seems that this algorithm is not suitable for solving the reliability problem. This algorithm lacks a specific parameter so it cannot be adjusted and fit for this problem. In the meantime, the GA algorithm had an acceptable performance along with the FA. Although the ABC algorithm was better than these two algorithms. The interesting thing about the DE, ACO, and IWO algorithms is that as the model becomes more difficult, the solutions of these three algorithms get exponentially worse. However, these algorithms work relatively well in simple and small models. The GWO algorithm also recorded a moderate performance.

Table 4 shows the results of running the algorithms based on the second objective function. The performance of the algorithms is almost similar to the performance based on the first objective function. The best solutions are provided by the GPC algorithm and after that, the ABC algorithm provides better solutions. SA and TLBO algorithms also had the worst performance. The conditions of other algorithms are similar to the first objective function. GA, FA, and BBO algorithms had better performance than other algorithms after the ABC algorithm.

In Table 3, it is possible to compare the cost and the level of reliability obtained through each algorithm based on each model. Based on our models and experiments, the desired level of reliability is 85 %. Therefore, it is possible to compare the cost and the obtained reliability and get a deep understanding of the performance of the algorithms. For example, in model number #8, about 85.04 % reliability has been achieved through the GPC algorithm. The cost that GPC has spent to achieve this reliability is about 1126.1. However, in this model, the FA algorithm meets 85.05 % reliability. Although the percentage of reliability in this model for these two algorithms is almost close to each other, the FA algorithm meets this percentage of reliability at a much higher cost than GPC. This cost is for FA 1227.1, which is a high cost compared to GPC. Table 4 also compares the maximum reliability achieved with the minimum possible cost of setting up the network. For example, in model number #2, the four algorithms GPC, ABC, DE, and GA pay the lowest cost to minimize the network setup cost.

In a general view, the results and tables show that the proposed algorithm and approach can be used well to solve the reliability problem. Also, the results show that the proposed approach based on the GPC algorithm can solve the problem better than other algorithms and has better performance.

Normally, the performance of the algorithms can be influenced by the specific problem at hand. In a few models, the GPC algorithm may not be the best choice, while in others, it is excellent. The performance of the SA or ACO algorithms can be attributed to their nature of being trajectory-based algorithms. These algorithms may not be suitable for solving complex reliability problems due to their inherent limitations. The performance of the TLBO algorithm can be attributed to its lack of adjustable parameters. This algorithm may not be suitable for solving the reliability problem as they cannot be adjusted and fit for this problem. The performance of the GA, FA, GWO, and BBO algorithms can be attributed to their ability to adapt and optimize their solutions based on the problem at hand. These algorithms are suitable for solving complex reliability problems compared to the SA and TLBO algorithms. The performance of the GPC algorithm can be attributed to its unique characteristics, such as its ability to find the best solution in a short amount of time. This algorithm is suitable for solving complex reliability problems compared to the ABC, GA and other compared algorithm. As a sample, the convergence diagram of all algorithms in solving model number #6 is shown in Fig. 11.

In summary, the GPC algorithm performs well in solving the reliability problem, outperforming other competing algorithms in most cases. The ABC algorithm also provides good solutions, while the SA and TLBO algorithms perform poorly. The performance of the DE, ACO, and IWO algorithms can vary depending on the complexity of the problem and model. The GA, FA, and BBO algorithms have acceptable performance, while the GWO algorithm recorded a moderate performance. Overall, the proposed algorithm and approach can effectively solve the reliability problem, with further potential for improvement through parameter tuning or optimization techniques

Table 2
Details of created instances.

Model number	Number of nodes	Area (m^2)	$R_{\min}$	$C_{\min}$
#1	5	100	0.85	19
#2	7	225	0.85	47
#3	10	400	0.85	153
#4	12	625	0.85	252
#5	15	900	0.85	575
#6	18	1225	0.85	969
#7	20	1600	0.85	1329
#8	25	2500	0.85	2665
#9	30	3600	0.85	4363
#10	35	4900	0.85	8071
#11	40	6400	0.85	11304
#12	45	8100	0.85	16943

6. Statistical analysis

By observing the table of results (Table 3 and Table 4), it can be seen

Table 3
Objective function cost and obtained R based on first objective function (Eq. (8)) for all models.

Model	Criteria	Algorithm										
		ABC	ACO	BBO	DE	FA	GA	GWO	IWO	SA	TLBO	**GPC**
#1	Cost	141.03	148.38	154.20	**138.74**	149.95	149.09	168.40	143.65	234.99	372.27	139.46
		±8.2481	±6.8930	±14.267	**±5.8881**	±12.421	±9.8910	±15.697	±8.2652	±175.31	±0.0000	±6.3521
	Obt R	0.9569	0.9569	0.9585	0.9569	0.9577	0.9569	0.9600	0.9569	0.9675	0.9569	0.9569
		±0.0000	±0.0000	±0.0050	±0.0000	±0.0024	±0.0000	±0.0065	±0.0000	±0.0121	±0.0000	±0.0000
#2	Cost	**178.91**	203.51	214.94	**178.91**	189.63	189.42	222.44	188.89	320.01	506.54	**178.91**
		±0.5598	±14.357	±26.550	**±0.5598**	±10.749	±12.935	±30.749	±9.2156	±98.537	±0.0000	**±0.5598**
	Obt R	0.9413	0.9484	0.9449	0.9413	0.9427	0.9428	0.9434	0.9420	0.9753	0.9413	0.9413
		±0.0000	±0.0065	±0.0061	±0.0000	±0.0044	±0.0032	±0.0048	±0.0024	±0.0151	±0.0000	±0.0000
#3	Cost	307.89	360.40	350.69	319.34	313.35	325.38	405.56	342.06	701.38	998.70	**307.87**
		±1.9550	±27.473	±24.798	±7.3970	±7.5702	±12.142	±45.436	±22.788	±207.10	±2.3e−13	**±1.2321**
	Obt R	0.9335	0.9316	0.9294	0.9280	0.9304	0.9213	0.9257	0.9270	0.9777	0.9228	0.9335
		±0.0000	±0.0082	±0.0050	±0.0069	±0.0063	±0.0061	±0.0068	±0.0064	±0.0087	±0.0071	±0.0000
#4	Cost	**309.71**	379.15	350.79	326.31	312.40	321.87	448.07	387.64	1203.8	1202.4	**309.71**
		±5.9e−14	±34.507	±22.645	±8.0629	±3.9256	±9.6187	±63.969	±25.521	±257.92	±2.3e−13	**±5.9e−14**
	Obt R	0.9173	0.9188	0.9163	0.9185	0.9172	0.9188	0.9132	0.9175	0.9989	0.9192	0.9173
		±0.0000	±0.0080	±0.0045	±0.0022	±0.0001	±0.0031	±0.0070	±0.0010	±0.0014	±0.0051	±0.0000
#5	Cost	559.33	731.33	628.28	677.92	565.65	601.06	860.97	761.67	2747.7	2651.7	**559.12**
		±0.5147	±45.969	±54.482	±37.097	±10.797	±30.309	±62.040	±81.611	±551.96	±4.7e−13	**±0.3651**
	Obt R	0.8905	0.9087	0.9014	0.9165	0.8935	0.8976	0.8953	0.9016	0.9982	0.9200	0.8916
		±0.0060	±0.0056	±0.0084	±0.0103	±0.0042	±0.0079	±0.0069	±0.0096	±0.0030	±0.0103	±0.0074
#6	Cost	1693.0	972.05	834.84	1064.8	1688.9	1720.1	1118.5	1245.7	4225.2	3463.8	**687.54**
		±10.575	±116.61	±88.975	±62.802	±7.9432	±33.277	±127.85	±96.575	±731.94	±0.0000	**±7.4652**
	Obt R	0.8755	0.8867	0.8870	0.9137	0.8738	0.8787	0.8780	0.8852	0.9991	0.9211	0.8789
		±0.0042	±0.0120	±0.0035	±0.0162	±4.8e−05	±0.0072	±0.0065	±0.0070	±0.0011	±0.0151	±0.0078
#7	Cost	921.62	1265.5	1072.8	1494.2	882.73	967.77	1389.7	1636.0	6206.9	6574.9	**879.14**
		±29.659	±125.44	±69.548	±106.20	±6.2616	±45.677	±183.28	±68.781	±607.91	±9.5e−13	**±3.1541**
	Obt R	0.8644	0.8741	0.8715	0.9124	0.8602	0.8647	0.8635	0.8893	0.9923	0.9118	0.8631
		±0.0102	±0.0099	±0.0104	±0.0145	±0.0004	±0.0067	±0.0069	±0.0103	±0.0061	±0.0135	±0.0055
#8	Cost	1863.5	2279.5	1347.2	2727.0	1227.1	1221.2	1979.8	4738.2	12030.8	13261.0	**1126.1**
		±332.864	±238.25	±107.53	±78.008	±9.7433	±76.931	±298.61	±180.15	±1488.5	±0.0001	**±4.2362**
	Obt R	0.8616	0.8916	0.8523	0.9227	0.8505	0.8511	0.8510	0.9527	0.9999	0.9065	0.8504
		±0.0079	±0.0163	±0.0030	±0.0122	±0.0002	±0.0009	±0.0012	±0.0076	±4.1e−05	±0.01808	±0.0012
#9	Cost	1817.5	4485.6	1680.1	4963.4	1582.7	1537.0	2897.4	10719.9	19378.5	19286.2	**1529.0**
		±52.726	±612.78	±78.877	±201.46	±14.697	±45.981	±201.47	±451.92	±1290.2	±3.8e−12	**±27.089**
	Obt R	0.8664	0.9387	0.8520	0.9434	0.8696	0.8503	0.8516	0.9879	1.0000	0.9101	0.8517
		±0.0137	±0.0171	±0.0019	±0.0131	±0.0001	±0.0011	±0.0026	±0.0067	±0.0000	±0.0190	±0.0024
#10	Cost	2511.7	9248.9	2126.3	11435.8	1918.0	2018.9	4016.4	24672.1	36073.9	38404.8	**1914.8**
		±117.51	±538.94	±66.772	±323.48	±11.608	±45.120	±387.11	±740.24	±1971.1	±7.6e−12	**±29.541**
	Obt R	0.8583	0.9602	0.8521	0.9718	0.8526	0.8517	0.9498	0.9970	1.0000	0.9108	0.8519
		±0.0077	±0.0126	±0.0017	±0.0102	±0.0007	±0.0017	±0.0009	±0.0037	±0.0000	±0.0174	±0.0010
#11	Cost	3669.4	18285.8	2599.6	22086.4	2439.2	2760.6	5048.4	39854.7	49129.2	44226.3	**2392.5**
		±64.820	±722.47	±105.36	±482.84	±6.4242	±148.11	±554.52	±644.13	±2517.5	±0.0000	**±23.540**
	Obt R	0.8726	0.9888	0.8508	0.9949	0.8504	0.8532	0.8507	0.9998	1.0000	0.91733	0.8503
		±0.0174	±0.0071	±0.0017	±0.0047	±0.0004	±0.0031	±0.0018	±0.0004	±0.0000	±0.0144	±0.0006
#12	Cost	5174.8	48781.7	3491.4	44439.3	3564.7	4111.3	7406.2	66904.7	75967.0	71605.9	**3116.1**
		±171.13	±2257.2	±167.96	±913.81	±12.204	±221.35	±944.26	±1002.2	±3959.9	±1.5e−11	**±24.081**
	Obt R	0.872	0.9988	0.8509	0.9998	0.8504	0.8645	0.8512	0.9999	1.0000	0.9185	0.8508
		±0.01345	±0.0031	±0.0006	±0.0003	±0.0003	±0.0114	±0.0016	±3.0e−05	±0.0000	±0.0089	±0.0008

that the GPC algorithm is somewhat superior in solving the network reliability problem. But merely observation is not enough. Statistical analysis is needed to prove superiority. The purpose of statistical analysis is to discover the meaning of the data and find significant differences in the results so that identified trends can be found. In this paper, Friedman and Iman-Davenport tests were used to discover the meaning of the data. Friedman test works by ranking the data across all samples and then calculating the sum of ranks for each observation. The Iman-Davenport test is a modification of Friedman's test that is used to compare multiple related samples. It is a non-parametric test that uses the ranks of the data within each group to calculate the test statistic. It is a more powerful test than the original Friedman's test and is used to test the overall hypothesis that there is no difference among the groups. Table 5 shows Friedman's ranking based on the results obtained from two Tables 3 and 4. It should be noted that the ranking is based on the final cost of the objective function reported in the tables.

The table shows that the best ranking is related to the GPC algorithm, followed by ABC and GA respectively. The results of Friedman and Iman-Davenport tests are shown in Table 6. According to the table and according to the obtained results, the hypothesis is rejected. In this way, it can be concluded that there is a significant difference in the performance of the algorithms.

Because a significant difference has been observed in the results of the ranking test, we have used Holm's method as a post-hoc test for

Table 4
Objective function cost and obtained C based on second objective function (Eq. (9)) for all models.

Model	Criteria	Algorithm ABC	ACO	BBO	DE	FA	GA	GWO	IWO	SA	TLBO	**GPC**
#1	Cost	9.6991	9.7326	9.8254	**9.6657**	9.7326	**9.6657**	9.6991	**9.6657**	74.680	57.279	**9.6657**
		±0.1057	±0.1409	±0.3037	**±0.0000**	±0.1409	**±0.0000**	±0.1057	**±0.0000**	±21.963	±0.0000	**±0.0000**
	Obt C	18.554	16.492	16.666	20.615	16.492	20.615	18.554	20.615	158.21	18.554	20.615
		±6.5192	±8.6923	±8.8011	±0.0000	±8.6923	±0.0000	±6.5192	±0.0000	±42.643	±6.5192	±0.0000
#2	Cost	**7.5545**	7.9011	7.8540	**7.5545**	7.5919	**7.5545**	8.6311	7.5919	54.165	77.145	**7.5545**
		±0.0000	±0.5597	±0.5475	**±0.0000**	±0.1183	**±0.0000**	±1.0016	±0.1183	±18.434	±0.0000	**±0.0000**
	Obt C	44.241	44.810	44.294	44.241	44.669	44.241	43.638	44.669	302.35	45.765	44.241
		±0.0000	±2.6512	±2.2331	±0.0000	±1.3541	±0.0000	±8.2244	±1.3541	±87.630	±2.4708	±0.0000
#3	Cost	4.8732	4.5096	4.2431	3.8309	4.0474	3.9500	5.1982	4.1880	43.387	31.647	**3.8302**
		±0.0795	±0.3713	±0.2551	±0.0759	±0.3034	±0.0992	±0.4901	±0.3356	±16.505	±3.7e−15	**±0.0332**
	Obt C	161.72	148.72	152.82	162.48	158.20	157.33	152.21	156.19	813.95	158.90	151.96
		±5.7075	±5.6910	±6.6155	±4.7553	±6.3168	±5.4419	±13.775	±7.9629	±260.20	±9.9708	±3.4544
#4	Cost	0.9239	2.2211	1.2954	0.9461	0.9516	1.0471	2.8616	1.8964	36.579	37.696	**0.9227**
		±0.0304	±0.6685	±0.2152	±0.0575	±0.0502	±0.1542	±0.7567	±0.5718	±7.2213	±7.4e−15	**±0.0238**
	Obt C	248.01	266.31	255.12	249.56	250.15	252.49	251.96	252.69	1174.2	253.86	250.46
		±2.2879	±20.927	±10.038	±3.7440	±3.7177	±5.1305	±14.794	±7.2391	±182.07	±3.6102	±4.2614
#5	Cost	0.1022	1.0042	0.4364	0.2765	0.1300	0.2756	1.4662	1.2686	37.010	33.187	**0.1009**
		±0.0006	±0.4129	±0.1768	±0.1301	±0.0853	±0.1489	±0.3381	±0.3388	±6.6182	±0.0000	**±0.0006**
	Obt C	568.33	567.32	570.65	574.13	567.69	571.06	563.79	580.82	2702.6	568.11	571.16
		±4.6423	±9.3765	±12.884	±12.435	±8.4339	±7.4288	±12.169	±10.056	±380.51	±6.0572	±5.6888
#6	Cost	0.4432	0.2098	0.0501	0.0571	0.0443	0.0491	0.5375	0.7623	40.272	37.171	**0.0416**
		±0.4332	±0.1652	±0.0047	±0.0056	±0.0021	±0.0043	±0.3494	±0.3730	±5.2024	±7.4e−15	**±0.0003**
	Obt C	964.12	952.04	966.15	951.87	961.17	963.36	956.69	973.08	4872.4	963.03	964.48
		±0.8971	±20.140	±3.4366	±12.187	±5.3477	±6.5185	±8.7744	±13.338	±504.22	±2.7033	±8.7248
#7	Cost	**0.0296**	0.2592	0.0372	0.0586	0.0306	0.0362	0.0353	1.6948	37.009	36.325	0.0300
		±0.0008	±0.1564	±0.0048	±0.0039	±9.1e−05	±0.0045	±0.1624	±0.7750	±4.7381	±7.4e−15	±0.0036
	Obt C	1325.4	1320.2	1322.2	1310.7	1321.6	1316.6	1319.2	1403.8	6246.6	1312.0	1319.9
		±2.9715	±15.019	±9.2962	±16.918	±7.3071	±5.9180	±13.521	±83.141	±630.18	±16.064	±8.1499
#8	Cost	0.0056	0.0851	0.0061	0.0325	0.0041	0.0051	0.0496	7.0506	37.223	36.800	**0.0040**
		±0.0004	±0.0735	±0.0008	±0.0038	±0.0004	±0.0009	±0.0118	±1.4736	±3.1156	±7.4e−15	**±0.0003**
	Obt C	2654.5	2609.4	2658.7	2612.6	2659.8	2652.5	2630.0	4425.6	12585.5	2607.7	2660.6
		±4.6151	±63.069	±9.8276	±57.312	±4.3362	±10.343	±27.736	±428.48	±830.33	±53.559	±3.6507
#9	Cost	0.0015	0.0622	0.0012	0.0290	0.0007	0.0008	0.0251	14.522	34.836	41.178	**0.0006**
		±0.0002	±0.0149	±0.0003	±0.0055	±8.0e−05	±0.0001	±0.0107	±0.8884	±4.0380	±0.0000	**±5.0e−05**
	Obt C	4292.8	4298.1	4346.1	4272.3	4357.6	4351.4	4309.8	10675.3	19559.7	4290.3	4353.9
		±48.682	±50.133	±17.102	±73.894	±8.1485	±10.936	±40.748	±418.28	±1761.5	±79.430	±6.8027
#10	Cost	0.0004	0.0879	0.0001	2.4586	5.4e−05	0.0001	0.0096	20.111	33.546	32.362	**4.8e−05**
		±6.8e−05	±0.0602	±4.6e−05	±0.3837	±6.2e−06	±2.3e−05	±0.0031	±0.6410	±2.8377	±0.0000	**±3.5e−06**
	Obt C	7990.9	7865.2	8047.0	9912.2	8066.81	7992.1	8011.6	24301.8	35145.7	7864.1	8062.1
		±75.715	±230.78	±15.460	±288.18	±3.8364	±84.375	±50.430	±517.45	±2290.2	±196.44	±7.1995
#11	Cost	0.0001	5.6954	2.0e−05	9.3195	3.7e−06	4.5e−05	0.0053	25.247	32.831	34.927	**3.1e−06**
		±3.7e−05	±1.0647	±5.1e−06	±0.7497	±5.4e−07	±1.4e−05	±0.0027	±0.5336	±1.9422	±7.4e−15	**±3.9e−07**
	Obt C	11107.6	17685.3	11283.9	21818.5	11298.4	11226.3	11183.0	39841.0	48414.5	11143.3	11299.0
		±199.90	±1223.6	±14.897	±838.78	±7.5535	±121.15	±116.69	±601.68	±2195.3	±115.24	±3.2458
#12	Cost	0.0001	19.288	9.3e−06	15.980	7.3e−06	4.1e−05	0.0041	29.085	34.056	36.829	**1.0e−06**
		±2.2e−05	±0.6853	±4.2e−06	±0.6019	±4.0e−07	±1.5e−05	±0.0016	±0.4376	±2.3787	±7.4e−15	**±2.2e−07**
	Obt C	16614.5	49621.9	16933.0	44016.4	16929.3	16869.0	16485.7	66222.9	74645.5	16474.1	16930.0
		±260.44	±1161.9	±9.3747	±1021.0	±10.606	±50.419	±402.69	±741.47	±4030.3	±397.16	±10.182

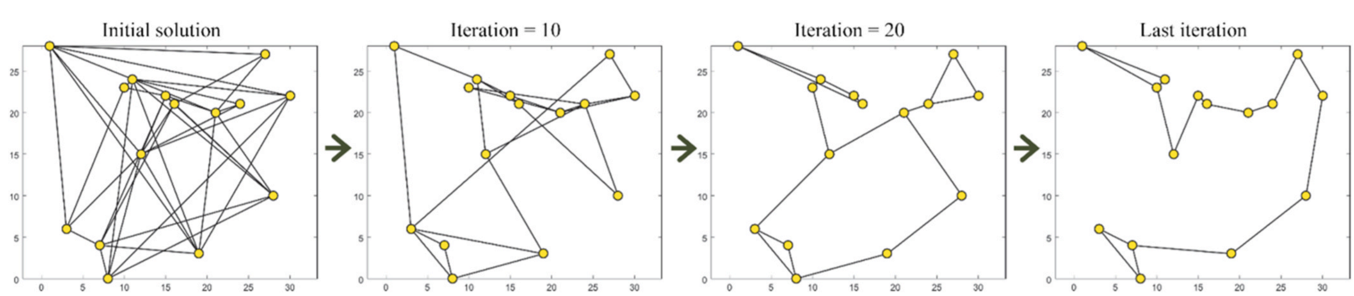


Fig. 10. An example of solving the network reliability model by the approach based on the GPC algorithm.

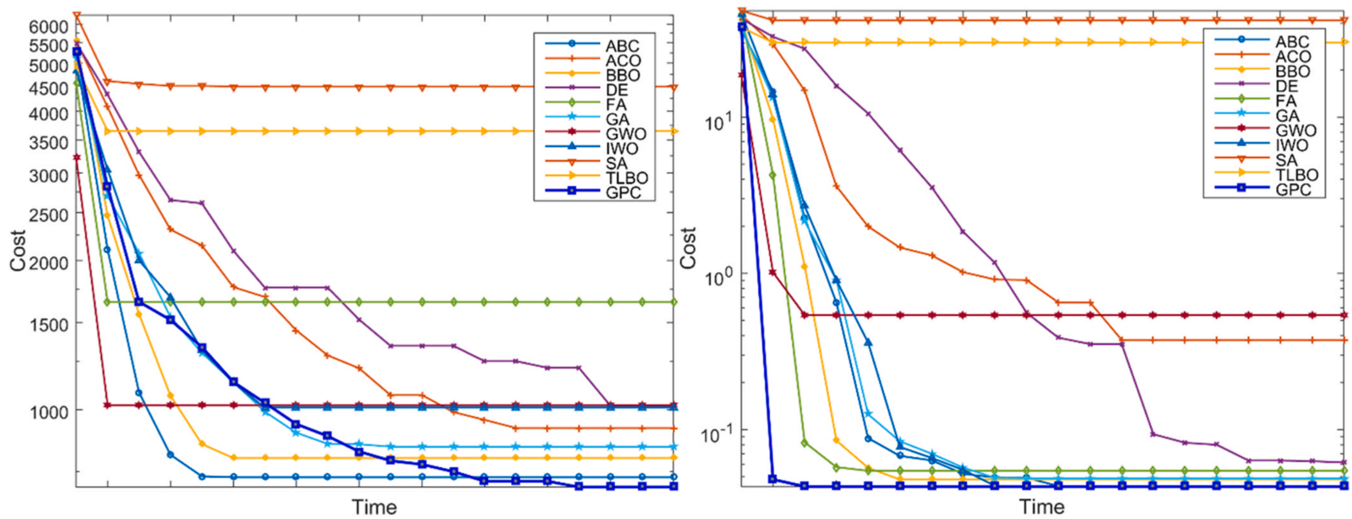


Fig. 11. Convergence diagram of all algorithms in solving model number #6 (as a sample). Left is based on first objective function (Eq. (8)), and Right is based on second objective function (Eq. (9)).

Table 5
Ranking of the algorithms (sorted out).

Algorithm	Mean rank
GPC	**1.73**
ABC	3.69
GA	3.94
FA	3.96
BBO	5.06
DE	5.42
GWO	6.73
ACO	6.94
IWO	7.54
TLBO	10.46
SA	10.54

Table 6
Results of Friedman's and Iman–Davenport's tests.

Test method	Chi-Square	Degrees of freedom (DF)	*p*-Value	Hypothesis
Friedman	170.0685	10	2.68e−31	Rejected
Iman–Davenport	54.8940	10	$<$ 2.20e−16	Rejected

Table 7
Result of Holm's method (GPC is the control algorithm).

Algorithm	*j*	*α/j*	*z*-Score	*p*-Value	Hypothesis
ABC	1	0.0500	2.047153234	0.020324	Rejected
GA	2	0.0250	2.308269718	0.010494	Rejected
FA	3	0.0166	2.329159037	0.009927	Rejected
BBO	4	0.0125	3.478071566	0.000253	Rejected
DE	5	0.0100	3.854079303	0.000058	Rejected
GWO	6	0.0083	5.222329679	$<$ 0.00001	Rejected
ACO	7	0.0071	5.441667525	$<$ 0.00001	Rejected
IWO	8	0.0062	6.068347087	$<$ 0.00001	Rejected
TLBO	9	0.0055	9.118187619	$<$ 0.00001	Rejected
SA	10	0.0050	9.201744894	$<$ 0.00001	Rejected

better analysis. The purpose of Holm's method as a post-hoc test is to determine which specific groups or levels are different from one another. This method used to adjust the significance level of multiple comparison tests to prevent false positives. It involves ordering the p-values from smallest to largest and then comparing each p-value with the adjusted significance level. In this test, the best rank obtained from Friedman's ranking is compared one by one with the results of other algorithms. As Table 5 shows, the best rank is related to the GPC algorithm, so this algorithm is considered the control algorithm. It should also be noted that the confidence interval is 95 % ($\alpha = 0.05$). The results obtained from Holm's method are specified in Table 7. The results show that the GPC algorithm is significantly different from other algorithms. Therefore, it can be concluded that the approach to solving the network reliability problem used in this paper based on the GPC algorithm is a more effective approach than other competing algorithms.

7. Solving a case study

Karaj Azad University is located in the north of Karaj city in Alborz province and at a distance of about 40 km from Tehran, the capital of Iran. The size of this university is estimated at 43 ha only in the urban area. This university includes eight independent buildings that make up the faculties. There are also other buildings with administrative, security, directorate, laboratory and mosque uses. In this paper, as a case study, we generate the university network and calculate its reliability so that the condition of connectivity and two-connectivity is met. For this purpose, two nodes on the sides of the building are considered for each independent building. Also, a node has been considered for each office building, laboratory, etc. A total of 50 active nodes are considered in specific locations. Fig. 12 shows the satellite map of Karaj Azad University.

To provide the university reliability model, we ran the GPC algorithm with the two mentioned objective functions (Eq. (8) and Eq. (9)) independently. All algorithm settings and problem demands such as failure probability, and $R_{\min}$ remained unchanged. Also, for this case study, the value of $C_{\min}$ is considered as one percentage of the total cost, which according to the number of nodes and their distance, this cost is 32071. To solve the model, the number of iterations is considered as a stopping condition. For this purpose, the number of iterations is 500. This is roughly equivalent to 2.5 times the NFE in the experiments section. The purpose of this adjustment is to give the algorithm the necessary time to get the best solution. Fig. 13 shows the reliability network graph of the university obtained from two objective functions. Also, Fig. 14 shows the network plot on the satellite image. Table 8 also shows the results obtained from the objective functions. The presented table, figure, and graphs show that there are no violations in the considered restrictions. Therefore, the GPC algorithm has been able to solve the model well.

Fig. 12. Satellite map of Karaj Azad University (Powered by Google).

8. Limitations

The network reliability problem is a complex task that involves determining the probability of a network remaining operational under certain conditions. While the GPC algorithm has been used to address this problem in this paper, there are scenarios where it may not perform optimally. In this section, we will discuss some of the limitations of the network reliability problem and the GPC algorithm, as well as suggest directions for future research.

The network reliability problem is inherently complex due to the large number of variables and constraints involved. Calculating the network reliability can be computationally expensive, especially for large networks. This is because the number of possible network configurations grows exponentially with the number of network elements. The network reliability problem often involves making assumptions and approximations about the network and its components. For example, the failure probabilities of network elements may be estimated based on historical data, which can introduce uncertainty and error into the calculations.

Despite these limitations, the proposed GPC algorithm and other soft computing methods remain important tools for network analysis and optimization. There are several directions for future research that could help to address the limitations discussed. The reliability of the network can be calculated with less computational cost. The solution could involve using parallel computing techniques or the development of the GPC algorithm. This development could be developing new variants of the GPC algorithm that are less susceptible to slow convergence, or developing methods for automatically tuning the GPC parameters. Finally, there is potential for integrating multiple optimization techniques, such as GPC and evolutionary algorithms, to address the network reliability problem. This could involve developing hybrid algorithms that combine the strengths of multiple optimization techniques to find the global optimum more efficiently.

In this paper specifically, hardware limitations made us unable to run larger models. Running larger models requires a lot of processing time, which was not possible with the hardware used for this study. This limitation existed not only for the running of the GPC algorithm but also for all algorithms. As a future work, one solution can be the use of cloud computing space. Also, as another future work, it is possible to develop a multi-objective version of the proposed algorithm and compare it with multi-objective algorithms in solving the reliability problem while applying it to different network models other real-world cases.

9. Discussion and conclusions

In general, the advantages of using metaheuristics over traditional methods are flexibility, efficiency, robustness, scalability, and ability to handle NP-hard problems. Metaheuristics offers several advantages when applied to the network reliability problem, which is a complex optimization task. Some of these advantages are mentioned below:

- Flexibility and Versatility: Metaheuristics are not problem-specific and can be adapted to a wide range of network reliability problems, regardless of their complexity, size, or specific constraints. This flexibility allows them to handle different types of variables (continuous, discrete, mixed) and various objective functions.
- Efficiency in Large-scale Problems: Traditional optimization methods, such as linear programming or dynamic programming, may struggle with the combinatorial nature of large-scale network reliability problems due to the exponential growth of possible solutions. Metaheuristics, on the other hand, can efficiently navigate the

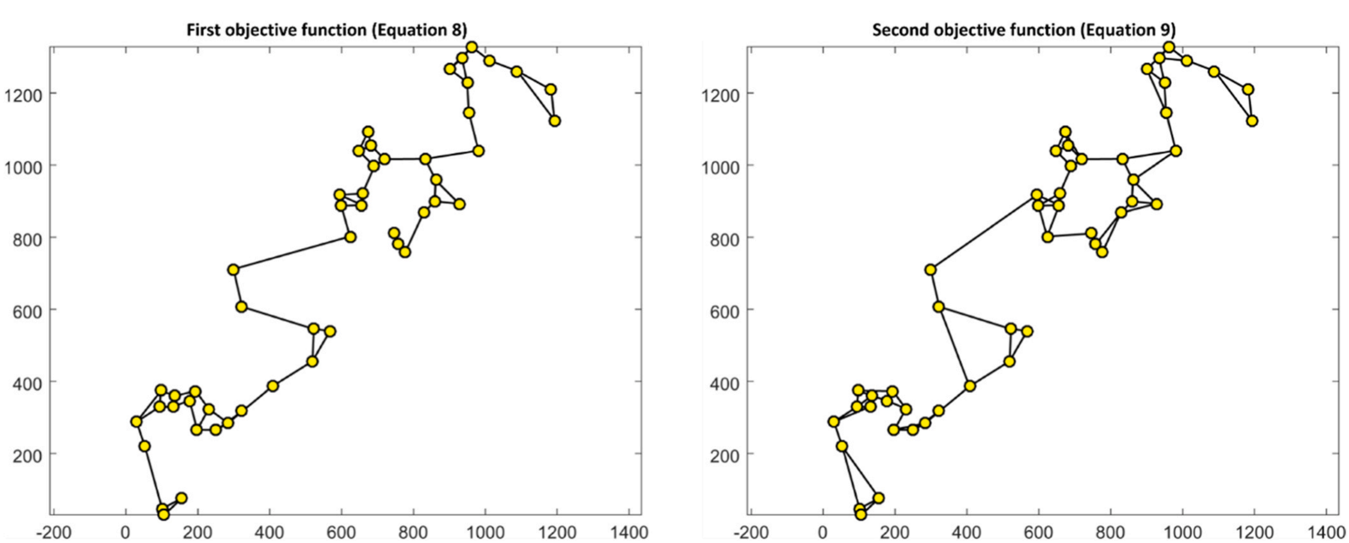


Fig. 13. Reliability network graph of the university obtained from two objective functions.

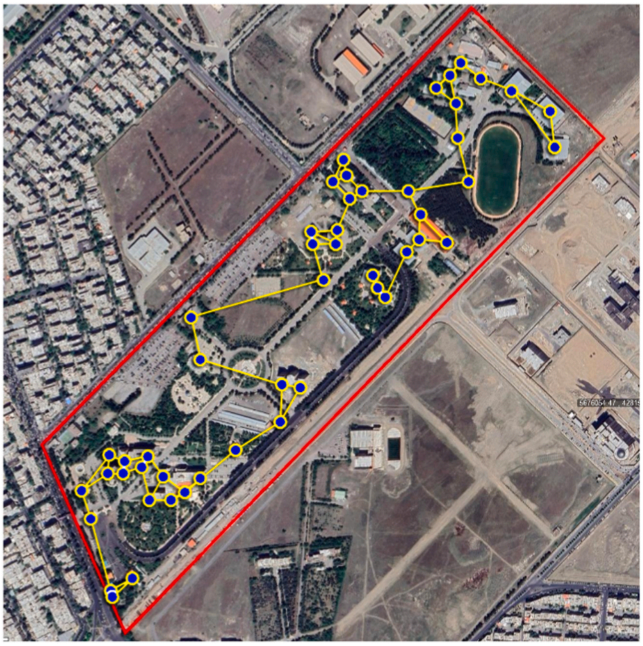


Fig. 14. Network plot based on first objective function (Eq. 8) on the satellite image.

Table 8
Case study results obtained from two objective functions.

First objective function			Second objective function		
Cost	$R_{\min}$	Obtained R	Cost	$C_{\min}$	Obtained C
25274.1255	0.85	0.8524	0.06868	32,071	31980.1737

solution space to find near-optimal solutions in a reasonable amount of time.

- Ability to escape local optima: Metaheuristics incorporate mechanisms that allow them to avoid getting trapped in local optima, which is a common issue with traditional local search methods.
- Robustness to stochastic elements: Network reliability often involves stochastic elements. Metaheuristics are well-suited to deal with such stochasticity and can incorporate probabilistic models to enhance the robustness of the solutions.
- Balance between exploration and exploitation: Metaheuristics strikes a balance between exploring the solution space to find new promising areas (exploration) and exploiting the best solutions found so far to refine them (exploitation). This balance is crucial for the effectiveness of the optimization process in complex network reliability problems.
- Lower computational costs: While traditional methods may require significant computational resources to solve network reliability problems exactly, metaheuristics can often provide high-quality solutions with much less computational effort, making them more cost-effective.
- Applicability to real-world constraints: Real-world network reliability problems often come with a set of practical constraints that may not be easily incorporated into traditional optimization models. Metaheuristics can more naturally integrate these real-world constraints into the optimization process.
- Continuous improvement: Metaheuristics are iterative and can be run for as long as computational resources allow, continuously improving the quality of the solution until a stopping criterion is met.

As a result, metaheuristics provide a powerful toolkit for addressing the challenges posed by network reliability problems, offering a good trade-off between solution quality and computational effort, and proving to be particularly effective for large and complex networks.

The GPC algorithm has several advantages over other optimization algorithms. It is easy to implement, computationally efficient and can handle complex optimization problems with a large number of variables. Moreover, GPC has that shown to perform well in solving optimization problems with non-convex and non-differentiable objective functions. One of the main advantages of GPC is its ability to handle large-scale optimization problems. This is well proven even in the main paper of GPC. This algorithm is scalable and can handle optimization problems with thousands of variables. Although the GPC algorithm is a new algorithm that has been published recently but has been applied to various real-world problems. Compared to other optimization algorithms, GPC has been shown to perform well in terms of computational complexity and scalability. GPC has lower computational complexity than other population-based algorithms, such as GA, FA, ACO, and so on. Moreover, GPC has shown to be more scalable than other optimization algorithms, making it suitable for solving large-scale optimization problems. However, PSO also has some limitations, such as sensitivity to parameter choice.

In this paper, an effective approach based on the GPC algorithm was proposed to solve the network reliability problem. The goal was to achieve the desired goals, such as increasing reliability and simultaneously reducing the cost of network design. For this purpose, two independent objective functions with constraints were defined. Then 12 different complex models of random networks were generated and the proposed algorithm was compared and evaluated with 10 other metaheuristic algorithms. The proposed approach was also used to solve the real model of university network reliability. The results of observation and statistical analysis showed that the proposed approach to solving the reliability problem works in a completely effective way.

In general, the key findings in this paper can include the following:

- The GPC has been shown to be effective in solving the network reliability problem. It can find near-optimal solutions in a reasonable amount of time, making it a practical approach for solving complex network reliability problems.
- The GPC has been shown to be effective in finding the optimal network topology that maximizes network reliability. This is important because the network topology has a significant impact on network reliability.
- The GPC has been shown to be effective in compliance conditions such as connectivity and two-connectivity.

In summary, the GPC algorithm demonstrated superior performance in solving the reliability problem compared to other algorithms. The ABC algorithm also performed well, while SA and TLBO algorithms showed poor performance. The DE, ACO, and IWO algorithms performed moderately well in simple and small models but their solutions worsened exponentially as the model complexity increased. The GA, FA, and BBO algorithms had acceptable performance after the ABC algorithm. Also, the statistical analysis proved that the GPC algorithm was significantly different from the other compared algorithms, indicating that the network reliability problem approach using the GPC algorithm is more effective than the other competing algorithms.

CRediT authorship contribution statement

Sasan Harifi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data Curation, Writing - Original Draft, Supervision, Visualization, Project administration, Writing - Review & Editing. **Amirmasoud Razavi:** Resources, Writing - Review & Editing, Visualization, Project administration. **Melika Heydari Rad:** Resources, Writing - Review & Editing, Visualization, Project administration. **Alireza Moradi:** Resources, Writing - Review & Editing, Visualization, Project administration.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data Availability

No data was used for the research described in the article.

References

[1] E. Zio, Reliability engineering: Old problems and new challenges, Reliab. Eng. Syst. Saf. 94 (2) (2009) 125–141.
[2] H. Pérez-Rosés, Sixty years of network reliability, Math. Comput. Sci. 12 (2018) 275–293.
[3] C.J. Colbourn, Concepts of network reliability, Wiley Encyclopedia of Operations Research and Management Science, 2010.
[4] L. Schäfer, S. Garcia, V. Srithammavanh, Simplification of inclusion–exclusion on intersections of unions with application to network systems reliability, Reliab. Eng. Syst. Saf. 173 (2018) 23–33.
[5] M. Ram, J.P. Davim (Eds.), Advances in Reliability and System Engineering, Springer, 2017.
[6] Y. Lyu, P. Yin, Internet of Things transmission and network reliability in complex environment, Comput. Commun. 150 (2020) 757–763.
[8] D.W. Coit, E. Zio, The evolution of system reliability optimization, Reliab. Eng. Syst. Saf. 192 (2019) 106259.
[9] T. Kundu, H. Garg, INNA: An improved neural network algorithm for solving reliability optimization problems, Neural Comput. Appl. 34 (23) (2022) 20865–20898.
[10] A. Davila-Frias, N. Yodo, T. Le, O.P. Yadav, A deep neural network and Bayesian method based framework for all-terminal network reliability estimation considering degradation, Reliab. Eng. Syst. Saf. 229 (2023) 108881.
[11] A. Heidari, Z.Y. Dong, D. Zhang, P. Siano, J. Aghaei, Mixed-integer nonlinear programming formulation for distribution networks reliability optimization, IEEE Trans. Ind. Inform. 14 (5) (2017) 1952–1961.
[12] H.Z. Iqbal, A. Ashraf, A. Ahmad, Power economic dispatch using particle swarm optimization. In 2015 Power Generation System and Renewable Energy Technologies (PGSRET), IEEE, 2015, pp. 1–7.
[13] A.J. Mezger, K.C. de Almeida, Short term hydrothermal scheduling with bilateral transactions via bundle method, Int. J. Electr. Power Energy Syst. 29 (5) (2007) 387–396.
[14] S. Yang, Analysis for the reliability of computer network by using intelligent cloud computing method, Int. J. Comput. Appl. 41 (4) (2019) 306–311.
[15] S. Moazzeni, M.R. Khayyambashi, N. Movahhedinia, F. Callegati, On reliability improvement of software-defined networks, Comput. Netw. 133 (2018) 195–211.
[16] L. Schäfer, S. Garcia, V. Srithammavanh, Simplification of inclusion–exclusion on intersections of unions with application to network systems reliability, Reliab. Eng. Syst. Saf. 173 (2018) 23–33.
[17] G. Bai, Z. Tian, M.J. Zuo, Reliability evaluation of multistate networks: an improved algorithm using state-space decomposition and experimental comparison, Iise Trans. 50 (5) (2018) 407–418.
[18] D.P. Kroese, K.P. Hui, S. Nariai, Network reliability optimization via the cross-entropy method, IEEE Trans. Reliab. 56 (2) (2007) 275–287.
[19] A.K. Solanki, H. Kumar, K.K. Singh, Calculation and evaluation of network reliability using ANN approach, Procedia Comput. Sci. 167 (2020) 2153–2163.
[20] A. Davila-Frias, N. Yodo, T. Le, O.P. Yadav, A deep neural network and Bayesian method based framework for all-terminal network reliability estimation considering degradation, Reliab. Eng. Syst. Saf. 229 (2023) 108881.
[21] M.R. Hassan, S. Alkhalaf, A.M. Hemeida, M. Ahmed, E. Mahmoud, Artificial intelligent applications for estimating flow network reliability, Ain Shams Eng. J. 14 (8) (2023) 102055.
[22] T. Kundu, H. Garg, INNA: An improved neural network algorithm for solving reliability optimization problems, Neural Comput. Appl. 34 (23) (2022) 20865–20898.
[23] C. Srivaree-Ratana, A. Konak, A.E. Smith, Estimation of all-terminal network reliability using an artificial neural network, Comput. Oper. Res. 29 (7) (2002) 849–868.
[24] Y. Lyu, P. Yin, Internet of Things transmission and network reliability in complex environment, Comput. Commun. 150 (2020) 757–763.
[25] H. Garg, An efficient biogeography based optimization algorithm for solving reliability optimization problems, Swarm Evolut. Comput. 24 (2015) 1–10.
[26] C.T. Yeh, L. Fiondella, Optimal redundancy allocation to maximize multi-state computer network reliability subject to correlated failures, Reliab. Eng. Syst. Saf. 166 (2017) 138–150.
[27] P. Kumari, S.K. Sahana, Swarm based hybrid ACO-PSO meta-heuristic (HAPM) for QoS multicast routing optimization in MANETs, Wirel. Pers. Commun. (2022) 1–23.
[28] K. Guleria, A.K. Verma, Meta-heuristic ant colony optimization based unequal clustering for wireless sensor network, Wirel. Pers. Commun. 105 (2019) 891–911.
[29] M. Chahal, S. Harit, Optimal path for data dissemination in vehicular ad hoc networks using meta-heuristic, Comput. Electr. Eng. 76 (2019) 40–55.
[30] L. Guo, Research on anomaly detection in massive multimedia data transmission network based on improved PSO algorithm, IEEE Access 8 (2020) 95368–95377.
[31] G. Abdullah, Z.A.H. Hassan, Using of particle swarm optimization (PSO) to addressed reliability allocation of complex network. Conference Series, IOP Publishing, J. Phys. Vol. 1664 (2020) 012125.
[32] S. Li, X. Chi, B. Yu, An improved particle swarm optimization algorithm for the reliability–redundancy allocation problem with global reliability, Reliab. Eng. Syst. Saf. 225 (2022) 108604.
[33] W.C. Yeh, Y.C. Lin, Y.Y. Chung, M. Chih, A particle swarm optimization approach based on Monte Carlo simulation for solving the complex network reliability problem, IEEE Trans. Reliab. 59 (1) (2010) 212–221.
[34] K. Watcharasitthiwat, P. Wardkein, Reliability optimization of topology communication network design using an improved ant colony optimization, Comput. Electr. Eng. 35 (5) (2009) 730–747.
[35] A.Y. Hamed, M.H. Alkinani, M.R. Hassan, A genetic algorithm to solve capacity assignment problem in a flow network. Computers, Mater. Contin. 64 (3) (2020) 1579–1586.
[36] H. Ai, Y. Fan, J. Zhang, K.Z. Ghafoor, Topology optimization of computer communication network based on improved genetic algorithm, J. Intell. Syst. 31 (1) (2022) 651–659.
[37] L.J. Liu, Optimization design of computer network reliability based on genetic algorithms, Chem. Eng. Trans. 51 (2016) 775–780.
[38] Y.K. Lin, C.T. Yeh, System reliability maximization for a computer network by finding the optimal two-class allocation subject to budget, Appl. Soft Comput. 36 (2015) 578–588.
[39] O. Ozkan, M. Ermis, I. Bekmezci, Reliable communication network design: The hybridisation of metaheuristics with the branch and bound method, J. Oper. Res. Soc. 71 (5) (2020) 784–799.
[40] Du, Y., & Zhang, H. (2020). Research on computer network reliability optimization method based on genetic algorithm. In 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE) (pp. 1606-1610). IEEE.
[41] K. Huang, Computer network reliability improvement method based on genetic algorithm, IOP Publishing, J. Phys. 2066012039 (2021).
[42] Y. Xiong, Optimization of Computer Network Reliability based on Genetic Algorithm. In 2023. 4th International Conference for Emerging Technology (INCET), IEEE, 2023, pp. 1–6.
[43] J.M. Won, A. Malisia, F. Karray, Reliable network design using hybrid genetic algorithm based on multi-ring encoding, Comput. Intell. Expens. Optim. Probl. (2010) 609–635.
[44] J.E. Ramirez-Marquez, C.M. Rocco, All-terminal network reliability optimization via probabilistic solution discovery, Reliab. Eng. Syst. Saf. 93 (11) (2008) 1689–1697.
[45] B. Liu, K. Iwamura, Topological optimization models for communication network with multiple reliability goals, Comput. Math. Appl. 39 (7-8) (2000) 59–69.
[46] N. Dehghani, R. Dashti, Optimization of recloser placement to improve reliability by genetic algorithm, Energy Power Eng. 3 (4) (2011) 508–512.
[47] Y.K. Lin, C.T. Yeh, Computer network reliability optimization under double-resource assignments subject to a transmission budget, Inf. Sci. 181 (3) (2011) 582–599.
[48] K.E. Adetunji, I.W. Hofsajer, A.M. Abu-Mahfouz, L. Cheng, A review of metaheuristic techniques for optimal integration of electrical units in distribution networks, IEEE Access 9 (2020) 5046–5068.
[49] M. Amohadi, M.F. Firuzabad, Optimal placement of switching and protection devices in radial distribution networks to enhance system reliability using the AHP-PSO method, Turk. J. Electr. Eng. Comput. Sci. 27 (1) (2019) 181–196.
[50] O. Kahouli, H. Alsaif, Y. Bouteraa, N. Ben Ali, M. Chaabene, Power system reconfiguration in distribution network for improving reliability using genetic algorithm and particle swarm optimization, Appl. Sci. 11 (7) (2021) 3092.
[51] M. Memari, A. Karimi, H. Hashemi-Dezaki, Reliability evaluation of smart grid using various classic and metaheuristic clustering algorithms considering system uncertainties, Int. Trans. Electr. Energy Syst. 31 (6) (2021) e12902.
[52] H. Shan, Y. Sun, W. Zhang, A. Kudreyko, L. Ren, Reliability analysis of power distribution network based on PSO-DBN, IEEE Access 8 (2020) 224884–224894.
[53] B.A. Tolson, H.R. Maier, A.R. Simpson, B.J. Lence, Genetic algorithms for reliability-based optimization of water distribution systems, J. Water Resour. Plan. Manag. 130 (1) (2004) 63–72.
[54] J. Rong-Hong, Design of reliable networks, Comput. Oper. Res. 20 (1) (1993) 25–34.
[55] S. Harifi, J. Mohammadzadeh, M. Khalilian, S. Ebrahimnejad, Giza Pyramids Construction: an ancient-inspired metaheuristic algorithm for optimization, Evol. Intelligence 14 (2021) 1743–1761.
[56] S. Harifi, M. Khalilian, J. Mohammadzadeh, S. Ebrahimnejad, New generation of metaheuristics by inspiration from ancient. In 2020 10th international conference on computer and knowledge engineering (ICCKE), IEEE, 2020, pp. 256–261.
[57] S. Ebrahimnejad, S. Harifi, An optimized evacuation model with compatibility constraints in the context of disability: an ancient-inspired Giza Pyramids Construction metaheuristic approach, Appl. Intell. 52 (13) (2022) 15040–15073.
[58] S. Harifi, A binary ancient-inspired Giza Pyramids Construction metaheuristic algorithm for solving 0-1 knapsack problem, Soft Comput. 26 (22) (2022) 12761–12778.
[59] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. 39 (2007) 459–471.
[60] M. Dorigo, M. Birattari, T. Stutzle, Ant colony optimization, IEEE Comput. Intell. Mag. 1 (4) (2006) 28–39.

[61] D. Simon, Biogeography-based optimization, IEEE Trans. Evolut. Comput. 12 (6) (2008) 702–713.
[62] R. Storn, K. Price, Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (1997) 341–359.
[63] X.S. Yang, Firefly algorithms for multimodal optimization. In International Symposium on Stochastic Algorithms, Springer Berlin Heidelberg, Berlin, Heidelberg, 2009, pp. 169–178.
[64] J.H. Holland, Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, MIT press, 1992.
[65] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey wolf optimizer, Adv. Eng. Softw. 69 (2014) 46–61.
[66] A.R. Mehrabian, C. Lucas, A novel numerical optimization algorithm inspired from weed colonization, Ecol. Inform. 1 (4) (2006) 355–366.
[67] S. Kirkpatrick, C.D. Gelatt Jr, M.P. Vecchi, Optimization by simulated annealing, science 220 (4598) (1983) 671–680.
[68] R.V. Rao, V.J. Savsani, D.P. Vakharia, Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems, Comput. -Aided Des. 43 (3) (2011) 303–315.
[69] S. Harifi, R. Mohamaddoust, Zigzag mutation: a new mutation operator to improve the genetic algorithm, Multimed. Tools Appl. 82 (29) (2023) 45411–45432.
[70] R. Ranjbarzadeh, P. Zarbakhsh, A. Caputo, E.B. Tirkolaee, M. Bendechache, Brain tumor segmentation based on optimized convolutional neural network and improved chimp optimization algorithm, Comput. Biol. Med. 168 (2024) 107723.
[73] W.L. Liu, J. Zhong, P. Liang, J. Guo, H. Zhao, J. Zhang, Towards explainable traffic signal control for urban networks through genetic programming, Swarm Evolut. Comput. 88 (2024) 101588.
[74] X. Xu, W. Liu, L. Yu, Trajectory prediction for heterogeneous traffic-agents using knowledge correction data-driven model, Inf. Sci. 608 (2022) 375–391.
[75] H.R. Ibraheem, N.D. Zaki, M.I. Al-mashhadani, Anomaly detection in encrypted HTTPS traffic using machine learning: a comparative analysis of feature selection techniques, Mesop. J. Comput. Sci. 2022 (2022) 17–28.