International Journal of Industrial Engineering, 32(2), 276-296, 2025

RELIABILITY OPTIMIZATION OF LINEAR AND LINEAR CONSECUTIVE *K-OUT-OF-N* SYSTEMS USING TEACHING-LEARNING-BASED OPTIMIZATION AND GENETIC ALGORITHM

Şefika Büşra Sazak and Özge Elmastaş Gültekin*

Department of Statistics
Ege University
İzmir, Turkey
*Corresponding author's e-mail: ozge.elmastas@ege.edu.tr

Numerous engineering applications involve ensuring the proper functioning of systems, minimizing errors, and optimizing the system and its subcomponents. Achieving desirable outcomes often requires enhancing positive factors through optimization methods while mitigating negative factors. In this context, metaheuristic algorithms are favored to find solutions aligned with the intended objectives. Among such algorithms, Teaching-Learning Based Optimization (TLBO) and Genetic Algorithm (GA) stand out, drawing inspiration from real-life processes. This study focuses on applying the TLBO algorithm to optimize the reliability of linear k-out-of-n: F and G (lin/k/n: F and lin/k/n:G) and linear consecutive k-out-of-n: F and G (lin/con/k/n:F and lin/con/k/n:G) systems. Additionally, the system was analyzed using GA, and the results from both approaches were compared. By employing these powerful metaheuristic algorithms, we aim to attain effective and robust solutions for enhancing system reliability and performance. Also, this study can be a guide in terms of contributing to the reduction of costs by ensuring more efficient use of resources, especially in complex systems. It can also increase productivity by reducing labor by ensuring the efficient operation of machines and processes.

Keywords: System Reliability Optimization, k-out-of-n Systems, Teaching-Learning Based Optimization Algorithm, Genetic Algorithm

(Received on October 1, 2024; Accepted on February 5, 2025)

1. INTRODUCTION

Reliability stands as a paramount factor for any component, system, or product. For a company, a highly reliable product enhances the perception of the company's overall reliability. Similarly, from the customer's perspective, reliable products contribute significantly to heightened satisfaction. While it may appear that reliability adds to the costs, a well-balanced approach to cost and reliability evaluations can lead to maximum performance. Striking the right balance between cost and reliability can result in optimal outcomes and overall efficiency.

Systems consist of interconnected subcomponents, and various factors in these subcomponents influence the overall system's behavior. The simplest types of systems are serial and parallel systems and their combination results in more complex systems. While straightforward systems are easier to solve and comprehend, complexities arise when dealing with intricate systems. An example of complex structured systems includes k-out-of-n systems, where reliability calculation becomes manageable if components have identical properties or failure times. However, if the components in a subsystem differ, estimating the reliability of the system or subsystem becomes more challenging, requiring the consideration of multi-term distributions. This scenario is commonly referred to as component mixing. The reliability assessment of a component-mixed k-out-of-n system involves more intricate mathematical equations (Coit and Liu, 2000). Finding solutions for such systems requires specialized methods and analysis due to their increased complexity.

The use of optimization methods is often preferred in system reliability calculations. The purpose of design in system reliability optimization is to maximize system reliability to meet system constraints such as cost, weight, etc. In applications, designing and improving system reliability to meet system requirements can lead to very complex problems. Therefore, the purpose of a design is usually to select the component type and number of replacement components from the design alternatives to maximize system reliability by considering the constraints associated with the system (Sooktip *et al.*, 2011). The optimization process in system reliability can be classified in different ways according to the system structure and various criteria on which optimization work will be carried out. While analytical optimization methods aim to reach the solution by scanning the entire set of solutions, it takes a very long time to reach the result in large-sized problems. Therefore, it is more

DOI: 10.23055/ijietap.2025.32.2.10511 **ISSN 1072-4761**

advantageous to use intuitive and meta-heuristic optimization methods that approach the solution set intuitively and do not aim to reach a solution close to the best.

Baladeh and Taghipour (2022) present alternative strategies for addressing uncertainty in redundancy allocation problems arising from varying working conditions. They propose a novel dynamic k-out-of-n system wherein potential operating scenarios are represented discretely. Employing a genetic algorithm, the model is applied to optimize a fire detection system. Sharifi *et al.* (2021) presents a new redundancy allocation problem for a system with n-to-k configuration at the subsystems level with active and cold standby redundancy strategies to maximize the reliability of the system and minimize system costs. A sequential Monte Carlo model was also developed for reliability analyses by Peiravi *et al.* (2022). They also developed an effective genetic algorithm to solve the resulting optimization problem by applying the strategies they determined to a serial-parallel system. The reliability analysis of consecutive k-out-of-n systems used in engineering applications by considering the operating risks of the components under stress is calculated in Gökdere and Gürcan (2016). Peng and Xiao (2018) propose a new closed-form expression for a consecutive-k-out-of-n system consisting of three nonidentical types of components. The dynamic survival function and mean time to failure of this system are derived, and its applicability is demonstrated through numerical experiments. A new method for mechanical design optimization problems limited by the TLBO algorithm, which is a meta-heuristic method, was developed by Rao *et al*. (2012). Also, the TLBO algorithms for continuous, nonlinear, large-scale problems inspired by the interaction between teacher and student are used in the study. A study for reliability optimization of bi-state nonrepairable systems is presented in Soltani (2014). In his research, he examined redundancy allocation, reliability allocation, and three types of reliability optimization methods that combine from different perspectives. Linh (2023) presented an optimization method aimed at minimizing power loss by considering faults in distribution networks. The coordinated use of the Genetic Algorithm, Particle Swarm Optimization, and Teaching-Learning-Based Optimization accelerated the solution process and provided reliable results. Taheri *et al.* (2023) demonstrate the use of the TLBO algorithm to optimize the allocation of wind turbines, photovoltaic units, and fuel cells, achieving superior results in reducing voltage losses, lowering costs, and decreasing greenhouse gas emissions. Ullah *et al.* (2023) demonstrate that the TLBO algorithm is an effective tool for cost-based hybrid flowshop scheduling, minimizing labor, energy, maintenance, and delay costs and outperforming other optimization algorithms.

Saleem (2017) tried to optimize system reliability under constraints such as cost and weight by using statistical analysis with the Genetic Algorithm method for reliability optimization of serial and parallel systems. Eryılmaz (2017) stated that in cases where the number of units in a parallel system is randomly distributed, the optimal number of units and replacement time that minimizes the average cost rate is optimized and results in a wider distribution class than the Poisson distribution in the literature have been obtained. A three-step approach to solving multipurpose system reliability optimization problems that take uncertainty into account is presented in Cao *et al*. (2018). To solve the uncertainty problem, an entropy-based approach to the redundancy allocation problem was proposed by determining the deterministic reliability of each component. Wang *et al.* (2018) carried out research on optimizing the reliability of systems by considering the importance-based component improvement costs. The aim was to enhance the overall reliability of the system by identifying the weakest components while considering their importance and maintenance costs as constraints. The parameter optimization of fabric finishing systems in textile industries using the TLBO algorithm is performed by Kumar *et al.* (2018). In Coit and Zio (2019), it's presented a study that addresses the development of system reliability optimization, reliability optimization problems, residual allocation problems, and reliability allocation problems. The Birnbaum significance-based quantum genetics algorithm for reliability optimization of *F* systems with linear consecutive k-out-of-n is proposed by Zhao *et al.* (2019). Nakamura and Yamamoto (2022) developed an algorithm to calculate the system signatures of linear and circularly connected (1,2) or (2,1) n-to-m lattice systems, and the efficiency of the proposed algorithm was examined through numerical experiments, and it was determined that it would contribute to various applications.

The aim of this study is to perform system reliability optimization for complex system structures, specifically lin/k/n and lin/con/k/n systems, using meta-heuristic methods inspired by real-life processes, namely TLBO and GA. While the literature often focuses on simpler serial and parallel configurations, the complexity of the systems addressed in this study allows us to explore the applicability of TLBO and GA to these specific structures. In this context, the novelty of the study lies in filling the knowledge gap regarding how these meta-heuristic methods can be used in complex system structures. The results obtained in this study may also shed light on many different studies in the future. For example, TLBO and Genetic algorithm applications and comparisons can be made on a different system where the components are dependent, and copulas will be used. In addition, similar studies can be done not only on linear and linear consecutive systems but also on complex systems such as circular systems and even phased mission systems.

This study aims to achieve the best system reliability by performing system reliability optimization for complex system structures and to find component reliability that gives the best system reliability value. At the same time, the TLBO and the GA, which are meta-heuristic methods designed by real-life inspiration, was used for system reliability optimization calculations, and the results of both methods were compared for the system in question. In conclusion, the motivation of this study is to try to obtain the best system reliability value using TLBO and GA reliability optimization techniques for linear and linear consecutive k-out-of-n: F and G systems. In this study, TLBO and GA reliability optimization techniques were used to obtain the most appropriate system reliability value for 2 different system structures (linear and linear consecutive). The components in the considered systems are assumed to be independent. Based on the assumption that the components in the system should be independent, it has been accepted that the average knowledge levels of students in TLBO are also independent. It is also to make comments on the most appropriate optimization method from the table created for the different system examples given. It is possible to say that many algorithms inspired by real-life data, such as TLBO and GA, exist and are used for the optimization of various system structures.

This paper is structured as follows: The second section provides an overview of reliability and system reliability, exploring different system types. In the third section, the concept of optimization in system reliability is discussed, followed by an explanation of TLBO and GA as meta-heuristic methods for reliability optimization in systems. The fourth section delves into the application of reliability optimization on lin/k/n:F and lin/con/k/n:F systems, presenting and comparing the findings from both methods. Finally, the fifth section offers recommendations based on the overall study results.

2. RELIABILITY AND SYSTEM RELIABILITY

Reliability can be defined as "a function of time". The reliability represents the probability that components will operate undisturbed at a given time and for a given period (Rausand and Hayland 2004). The reliability function of a unit is defined as follows:

$$R(t) = 1 - F(t) = P(T > t) \quad t > 0 \tag{1}$$

Here $R(t)$ is the reliability function. *R* denotes the reliability, and *t* denotes the time interval. The probability of error $F(t)$ is the probability that the system or a component of the system will fail before the time interval (t) is exceeded.

In system reliability, the system to be evaluated may consist of many elements and subsystems. All these elements and sub-systems have the possibility of performing their functions alone and in error. Therefore, the reliability of the system as a whole includes all the parts that make it up. Since the actual reliability value of the system cannot be known exactly, a value that may be very close to this value can be obtained by probability calculations and statistical methods. Another important element in calculating system reliability is whether system components can be repaired after their error. Because the distribution of failure times in reliability calculations may vary according to this feature. If the system components are repairable after failure, the time between failure is taken into consideration, while if the non-deterioration is nonrepairable, the distribution is found by taking into account the time to failure.

In the reliability literature, systems are generally divided into two groups as repairable and non-repairable systems. Non-repairable systems have no recycling after they are failures. So they fail only once, and the distribution of failure times of such systems can be modeled by a model of life such as Weibull as an example. Repairable systems, on the other hand, are systems that are recovered with any intervention after failure and are handled by modeling the failure times. Non-repairable systems that are not continuously in operation are systems that are conditioned to a specific task. In these systems, the system must continue to operate even in the event of an error of any component that makes up the system. This component may remain defective in the system, or if it can be repaired, it can be repaired and replaced with a new one when necessary. In the case of repairable, continuously operating systems, the error status of the system must be acceptable as long as it is not too frequent and long. In the event of an error, the defective system can be repaired and replaced with a new one, or a similar component can replace the component in this error state until the error is corrected. In short, in such systems, a cycle of operation, error, and repair is mentioned.

3. LINEAR K-OUT-OF-N :*F* AND *G* SYSTEMS *(LIN/K/N:F AND G SYSTEMS*)

Lin/k/n:F and G systems are considered specialized cases of serial and parallel systems comprising more complex structural arrangements. A k-out-of-n:G system consists of n independent binary elements, where at least k components must be operational for the system to function. On the other hand, a k-out-of-n:F system cannot perform its task if k or more components fail. When the systems have components with identical reliability, the reliability of a k-out-of-n:G system is equivalent to that of an (n-k+1)-out-of-n:F system (Kuo and Zuo 2003; Levitin 2005).

Serial systems will correspond to n-out-of-n systems because they can perform their function if they can work with all their elements in k-out-of-n systems, while in parallel systems, they will correspond to 1-out-of-n systems because they have a system structure that can perform their task when at least one component is successful. In parallel systems, the remaining $n - 1$ components can be considered unnecessary since at least one component is needed for the system to function fully. However, these unnecessary components they are included in the system to increase the probability of success of at least one component. Accordingly, the structure functions of series and parallel systems, which provide information about the general structure of the system, are as follows (Levitin 2005):

Serial System:

$$\Phi(x) = \prod_{i=1}^{n} x_i = \min\{x_1, x_2, \dots, x_n\} \tag{2}$$

Parallel System:

$$\Phi(x) = 1 - \prod_{i=1}^{n} (1 - x_i) = \max\{x_1, x_2, \dots, x_n\} \tag{3}$$

Here x_i variable for $1 \leq i \leq n$ is as follows to indicate the status of the component:

$$X_i = \begin{cases} 1, & if\ component\ i\ is\ successful \\ 0, & if\ component\ i\ is\ not\ successful \end{cases}$$

Here $\boldsymbol{x} = (x_1, x_2, \dots, x_n)$ the state vector represents the states of all the components that make up the system. The state of the system, to be shown with Φ ;

$$\Phi = \begin{cases} 1, & the\ system\ functions \\ 0, & the\ system\ fails \end{cases}$$

is expressed as follows. Accordingly, if the status of all components is known, information about the general condition of the system can be obtained. In addition, the system state is written as a deterministic function of the state of the components and $\Phi(\boldsymbol{x})$ is called the structure-function of the system.

$$\Phi(\boldsymbol{x}) = \Phi(x_1, x_2, \dots, x_n)$$

Reliability is used in the process of measuring whether the systems can perform the tasks expected to be performed within the specified time. For the variable x_i, which is the state of the i. component is given above, the expression $p_i = \mathrm{P}(x_i = 1) = E(x_i)$ indicates the reliability of the component i, and $E(.)$ is the expected value of a random variable. $q_i = 1 - p_i$ indicates the unreliability of the component i. Then, the reliability and unreliability of the Φ system for vectors $\boldsymbol{x}$ and $\boldsymbol{p} = (p_1, p_2, \dots, p_n)$ are defined as follows (Kuo and Zuo 2003):

$$R = \mathrm{P}(\Phi(x) = 1) = E\big(\Phi(x)\big) \tag{4}$$

$$Q = \mathrm{P}(\Phi(x) = 0) = 1 - R \tag{5}$$

k-out-of-n:F and G ((6) and (7), respectively) systems structural functions are expressed as follows:

$$\Phi(x) = \begin{cases} 1, \sum_{i=1}^{n} x_i > n - k \\ 0, \sum_{i=1}^{n} x_i \leq n - k \end{cases} \tag{6}$$

$$\Phi(x) = \begin{cases} 1, \sum_{i=1}^{n} x_i \geq k \\ 0, \sum_{i=1}^{n} x_i < k \end{cases} \tag{7}$$

The reliability of a k-out-of-n system is as follows:

$$R_{n:k}(p) = P\left\{\sum_{i=1}^{n} x_i \geq k\right\} = \sum_{j=k}^{n} \binom{n}{j} p^j \, (1-p)^{n-j} \tag{8}$$

3.1 *k-out-of-n* repairable systems

In a k-out-of-n:G system, all n components are initially operational. As time progresses, components fail one by one. The system fails if the number of functioning components drops below k or if the number of failed components reaches n - k + 1. To extend the system's operational life, resources can be allocated to repair failed components, maintaining the number of failed components below n - k + 1 for a longer period. The system is considered to have failed once the number of failed components exceeds n - k. When multiple mechanics or repair facilities are available, they can simultaneously repair several failed components. Increasing the number of repair facilities can significantly prolong the system's operational time before the first failure occurs (Kuo and Zuo, 2003).

3.2 *Lin/con/k/n:F* and *Lin/con/k/n:G* systems

Consecutive systems are regarded as conditional states of k-out-of-n systems. In consecutive systems, calculations are performed solely based on the operation or failure of consecutive components. They comprise n components arranged linearly, where a single linear path deviates. Systems that fail to fulfill their function when at least k or more components fail consecutively out of n components are termed lin/con/k/n:F systems, whereas systems in which at least k or more components can successfully perform their function are termed lin/con/k/n:G systems.

4. SYSTEM RELIABILITY OPTIMIZATION

Optimization is a concept used almost everywhere and aims to maximize the performance of the desired function in any system and minimize its cost. In short, it is the process of selecting the most appropriate alternative solution methods according to the purpose of the problem being interested. In system reliability optimizations, the purpose of optimization may be to try to maximize system reliability. While the factors that have a positive effect on the system can be improved by using optimization methods according to the purpose of the problem, the negative factors are also expected to be minimized or eliminated. While using these methods, the use of meta-heuristic algorithms, one of the sub-branches of artificial intelligence, has recently been preferred in order to find the solutions closest to the targeted purpose. These algorithms inspired by real-life include TLBO, GA, Ant Colony, and Particle Herd optimization algorithms. In this study, TLBO and GA methods are emphasized.

4.1 Meta-Heuristic algorithms

Today, various optimization methods are used to solve most problems. Mathematical and heuristic methods are used in optimization problems to achieve the desired goal under certain constraints. Mathematical methods achieve the goal by scanning the entire solution space in the problem, while heuristics aim to intuitively approach the set of solutions and find a result that is close to the best. As a result of the effective use of heuristics, meta-heuristic algorithms have emerged. In engineering and industry, metaheuristic algorithms play a crucial role in addressing global optimization challenges that classical techniques struggle with, particularly when dealing with irregular or nonlinear surfaces. In response to these challenges, numerous studies have delved into metaheuristic methods to enhance solution quality for intricate problems. In contemporary times, Teaching-Learning Based Optimization (TLBO) has emerged as a prominent metaheuristic algorithm. Drawing inspiration from the teaching-learning process, TLBO stands out as an innovative and robust solution for tackling global optimization problems. Meanwhile, Particle Swarm Optimization (PSO) remains widely employed, offering commendable efficiency in addressing optimization tasks. Additionally, the Genetic Algorithm (GA) remains a pivotal element within the realm of metaheuristics, finding application across diverse research domains (Shukla *et al.* 2019).

4.2 Teaching-Learning Based Optimization (TLBO) algorithm

TLBO algorithms were first designed in 2011 by Rao *et al.* (2011), inspired by the teaching and learning process. This algorithm is an optimization algorithm developed for the improvement of the existing problem by mimicking teacher-student relationships and the interactions of students among themselves.

The solution space of the TLBO algorithm consists of teachers and students in a class. This algorithm aims to achieve the goal by increasing the knowledge levels of average students. In the algorithm, the teacher and students consist of two basic phases known as the teacher phase and the student phase. In the TLBO algorithm, the output is evaluated in terms of students' results or grades, depending on the quality of the teacher. Therefore, a teacher is often considered the most knowledgeable person so that students can achieve better results in terms of their grades. In addition, students also learn from the interaction among themselves, which helps to improve their outcomes, and the best solution for the whole population is considered to be the teacher (Kumar *et al.* 2018).

$$\text{Class}=\begin{bmatrix} X_{11}\ X_{12}\ldots X_{1D} \\ X_{21}\ X_{22}\ldots X_{2D} \\ . \\ . \\ . \\ X_{p_n1}\ X_{p_n2}\ldots X_{p_nD} \end{bmatrix} = \begin{bmatrix} R_1 \\ R_2 \\ . \\ . \\ . \\ R_n \end{bmatrix} \begin{matrix} student\ 1 \\ . \\ . \\ . \\ . \\ student\ n \end{matrix} \tag{9}$$

First of all, there must be an initial population. This population consists of teachers and students. In the matrix in (1), each row is named learner, and $R_i;\ 1 \le i \le n$ represents the values derived from the purpose function created for the solution of the problem. Also, if the problem aims to try to maximize the objective function, the highest value row in the matrix R_i i.e. the student is recorded as the best teacher M_{new}, while if the problem aims to try to minimize the objective function, the lowest value row in the matrix R_i is the best teacher. It is saved as M_{new} and stored for use in later phases. In addition, the mean vector M_j of each column in the matrix, namely the design variables p_i, is used in the algorithm. All these processes are recalculated in each iteration and continue until the final result is reached.

The teacher phase forms the first part, where students learn from their teachers. In the teacher phase, students learn from their teachers by imitating information. Here the goal of the teacher is to give knowledge to the students and try to increase the average level of knowledge in his/her class. In the class, the teacher holds the highest level of experience and knowledge among all individuals, so students have access to as much information as the teacher's level of knowledge (Kumar *et al.* 2018). As described by Rao *et al.* (2011), the teacher tries to increase the average result of the class from any M_1 value to his own level. However, in practice, this is not possible, and a teacher can move the average of the class M_1 to any value of M_2 that is better than M_1 depending on his ability. If, at any iteration *i*, we consider M_j as the mean and T_i as the teacher, T_i will try to improve the current mean M_j towards it, so the new mean T_i will be determined as M_{new}. The difference between the current average and the new average is given in (2). Since r_i is a random number varying between 0-1, the subjectivity in M_{new} and M_i arises only from taking T_F as 1 or 2.

$$Difference\ Mean = r_i\left(M_{new} - T_F M_j\right) \tag{10}$$

$$T_F = round[1 + rand(0,1)\{2-1\}] \tag{11}$$

(2) gives the difference between the current average and the new average. (3) gives a teaching factor that decides the value of the average to be changed in the TF, and the value of TF can be either 1 or 2, which is a heuristic step and is decided randomly with equal probability. According to this difference average, the average knowledge levels of the students in the teacher phase are updated again in each iteration according to the following statement.

$$p_{new,i} = p_{old,i} + Difference\ Mean \tag{12}$$

$$R\left(p_{new,i}\right) > R\left(p_{old,i}\right) \tag{13}$$

If the equation is as in (5), the update is made. Otherwise, it is not.

In the student phase, after the teacher phase, all the best function results are saved for use in the student phase. In this part, students learn information by discussing and interacting with each other. If a student is more knowledgeable than other students, with the help of the knowledgeable student, the average knowledge level of the other students is updated in each iteration to achieve the desired goal.

$$p_{new,i} = p_{old,i} + r_i(p_i - p_{partner}) \quad Eğer\ R(p_i) > R(p_{partner}) \tag{14}$$

$$p_{new,i} = p_{old,i} - r_i(p_i - p_{partner}) \quad Eğer\ R(p_i) < R(p_{partner}) \tag{15}$$

$$R(p_{new,i}) > R(p_{old,i}) \tag{16}$$

If the equation is as in (8), the update is made. Otherwise, it is not.

4.2.1 TLBO algorithm for lin/con/2/4:F system

Before applying the TLBO algorithm, a small example is provided to demonstrate how the algorithm's phases progress. In the case of a lin/con/2/4:F system, the reliability of the system will be calculated based on the parameters and conditions specific to that configuration, following the algorithm's steps. This approach illustrates the effectiveness of TLBO in optimizing such systems.

$$\Phi(x) = (X_1 \cup X_2)(X_2 \cup X_3)(X_3 \cup X_4) \quad ; \quad 1 \le i \le n \tag{17}$$

$$\Phi(x) = (X_1 + X_2 - X_1X_2)(X_2 + X_3 - X_2X_3)(X_3 + X_4 - X_3X_4) \tag{18}$$

$$R(p) = (p_1 + p_2 - p_1p_2)(p_2 + p_3 - p_2p_3)(p_3 + p_4 - p_3p_4) \tag{19}$$

The aim is to find the optimal reliability value of the $R(p)$ function and the component reliability that gives this optimal value.

1st Step: Determine the startup parameters.
N= 4, D(n) = 4
Lower Limit-Upper Limit= rand (0,1)
Aim Function=max $R(p)$

Design variables are expressed by D(n). The variables will represent component reliability that is randomly generated between 0 and 1. In this case, the initial population will consist of a matrix of 4×4. Each row of the matrix refers to the student. Each column of the matrix will consist of p_i, $1 \le i \le 4$ variables.

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} \\ 0{,}742494 & 0{,}370786 & 0{,}207092 & 0{,}951109 \\ 0{,}952521 & 0{,}523495 & 0{,}114473 & 0{,}003138 \\ 0{,}428517 & 0{,}068051 & 0{,}493814 & 0{,}434693 \\ 0{,}483397 & 0{,}707022 & 0{,}331470 & 0{,}426662 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R(p)} \\ 0{,}403623 \\ 0{,}066243 \\ 0{,}176259 \\ 0{,}420857 \end{bmatrix}$$

The average vector is created by averaging each variable. $R(p)$The line with the highest system reliability value, i.e., the student, is recorded as the best teacher. Equation (2) is calculated, and the TF value in equation (3) is chosen randomly.

$$\begin{bmatrix} \boldsymbol{M(p_i)ort} & 0{,}651732 & 0{,}417339 & 0{,}286712 & 0{,}453900 \\ \boldsymbol{P_{teacher}} & 0{,}483397 & 0{,}707022 & 0{,}331470 & 0{,}426662 \\ \boldsymbol{r_i} & 0{,}432786 & 0{,}752905 & 0{,}925760 & 0{,}851210 \\ \boldsymbol{D_i} & -0{,}072853 & 0{,}218104 & 0{,}041435 & -0{,}023186 \\ \boldsymbol{TF} & 1 & & & \end{bmatrix}$$

2nd Step: Procedures at the teacher phase are carried out. Procedures are first initiated through the 1st student. New component values are calculated using Equation (4). Here, limit control must also be carried out when performing the operation.

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} & \boldsymbol{R(p)} \\ 0{,}669641 & 0{,}588890 & 0{,}248527 & 0{,}927923 & \boxed{0{,}564859} \end{bmatrix}$$

Looking at the newly calculated component values, there are no values that violate the upper and lower limits. Otherwise, values greater than the lower and upper limit are randomly reproduced, or values greater than the upper limit are taken equal to the upper limit value, while values smaller than the lower limit are taken equal to the lower limit value, and calculations are performed. Then, the matrix is updated using Equation (5). As a result of the update, the following values were obtained. Since the system reliability value at the teacher phase (0,564859) is higher than the reliability value in the initial solution space (0,403623), the update result was as follows:

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} & \boldsymbol{R(p)} \\ 0{,}669641 & 0{,}588890 & 0{,}248527 & 0{,}927923 & 0{,}564859 \end{bmatrix}$$

3rd Step: Procedures at the student phase are carried out. Equation (6) is applied to the 1st student after the teacher phase. The r_i value in the equation is again randomly generated between 0-1, and the result is as follows:

$$\begin{matrix} \boldsymbol{r_i} & 0{,}146719 & 0{,}678139 & 0{,}748355 & 0{,}647702 \end{matrix}$$

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} & \boldsymbol{R(p)} \\ 0{,}705019 & 0{,}942091 & 0{,}064965 & 1{,}247389 & 1{,}144752 \end{bmatrix}$$

For the 1st student, the 3rd student was randomly selected as the partner, and the results were obtained by comparing them according to the system reliability value. When the limit is checked, the value exceeding the limits (1,247389) is not seen. Therefore, for this value that is greater than the upper limit, it is randomly reproduced as [rand (0,1)], and calculation operations are performed.

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} & \boldsymbol{R(p)} \\ 0{,}705019 & 0{,}942091 & 0{,}064965 & 0{,}125228 & 0{,}169258 \end{bmatrix}$$

After the limit checks are provided, a new update is made using Equation (8). There is no need to update as the system reliability value at the student phase (0,169258) is lower than the reliability value at the teacher phase (0,564859).

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} & \boldsymbol{R(p)} \\ 0{,}669641 & 0{,}588890 & 0{,}248527 & 0{,}927923 & \boxed{0{,}564859} \end{bmatrix}$$

The algorithm for the 1st student has been terminated. Procedures are carried out in the 2nd student from the 2nd step.

2nd Step: Procedures at the teacher phase are carried out. The procedures are initiated by continuing from the 2nd student. New component values are calculated using Equation (4). Here, limit control must also be carried out when performing the operation.

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} & \boldsymbol{R(p)} \\ 0{,}879668 & 0{,}450642 & 0{,}04162 & -0{,}06971 & -0{,}011141 \end{bmatrix}$$

Looking at the calculated new component values, there is a value of (-0,011141) that violates the lower limit. Therefore, the value is randomly reproduced as [rand (0,1)], and the calculation operations are performed.

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} & \boldsymbol{R(p)} \\ 0{,}879668 & 0{,}450642 & 0{,}04162 & 0{,}10055 & \boxed{0{,}061018} \end{bmatrix}$$

Then, the matrix is updated using Equation (5). As a result of the update, the following values were obtained. Since the system reliability value at the teacher phase (0,061018) is lower than the reliability value in the initial solution space (0,066243), the update result was as follows:

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} & \boldsymbol{R}(\boldsymbol{p}) \\ 0{,}952521 & 0{,}523495 & 0{,}114473 & 0{,}003138 & 0{,}066243 \end{bmatrix}$$

3rd Step: Procedures at the student phase are carried out. Equation (6) is applied to the 2nd student after the teacher phase. The r_i value in the equation is again randomly generated between 0-1, and the result is as follows:

$$\boldsymbol{r_i} \quad 0{,}827457 \quad 0{,}003938 \quad 0{,}718825 \quad 0{,}005275$$

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} & \boldsymbol{R}(\boldsymbol{p}) \\ 0{,}564341 & 0{,}524218 & 0{,}270456 & 0{,}005372 & 0{,}142007 \end{bmatrix}$$

For the 2nd student, randomly, the 4th student was selected as a partner, and the results were obtained by comparing them according to the system reliability value. When the limit is checked, the value exceeding the limits is not seen. After the limit checks are also provided, it is decided whether a new update will be made using Equation (8). The update is made because the system reliability value (0,142007) in the student phase is greater than the reliability value in the teacher phase (0,066243).

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} & \boldsymbol{R}(\boldsymbol{p}) \\ 0{,}564341 & 0{,}524218 & 0{,}270456 & 0{,}005372 & 0{,}142007 \end{bmatrix}$$

When all these operations were performed for the 3rd and 4th students, respectively, in the last case (as a result of the 1st iteration), the matrix was found as follows.

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} \\ 0{,}669641 & 0{,}58889 & 0{,}248527 & 0{,}927923 \\ 0{,}564341 & 0{,}524218 & 0{,}270456 & 0{,}005372 \\ 0{,}355664 & 0{,}286155 & 0{,}535249 & 0{,}011634 \\ 0{,}410544 & 0{,}925126 & 0{,}372905 & 0{,}468097 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{p}) \\ 0{,}564859 \\ 0{,}142007 \\ 0{,}195111 \\ \color{red}{0{,}607122} \end{bmatrix}$$

In the initially randomly generated component reliability matrix, the highest system reliability value was 0,420857, while at the end of iteration 1, the system reliability value was found to be 0,607122. This represents the best reliability value at the end of the 1st iteration. In addition, the component reliability values that give this value were found as 0,410544, 0,925126, 0,372905 and 0,468097, respectively. By increasing the number of iterations, the best reliability value can be found at the end of each iteration.

It was calculated for each iteration with the MATLAB R2016Bb program and the best reliability value graph at the end of each iteration was presented. As a result, the best reliability value was found as 0,65363 as a result of the 1st iteration, while when the number of iterations increased, it reached the best solution better and had a system reliability value of 0,9051 at the end of the 20th iteration.

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} \\ 0{,}742494 & 0{,}370786 & 0{,}207092 & 0{,}951109 \\ 0{,}952521 & 0{,}523495 & 0{,}114473 & 0{,}003138 \\ 0{,}428517 & 0{,}068051 & 0{,}493814 & 0{,}434693 \\ 0{,}483397 & 0{,}707022 & 0{,}331470 & 0{,}426662 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}(\boldsymbol{p}) \\ 0{,}403623 \\ 0{,}066243 \\ 0{,}176259 \\ \color{red}{0{,}420857} \end{bmatrix}$$

The initial population is taken the same as the manually calculated sample. The best system reliability values in each iteration are shown in Figure 1. Presented below is the solution space matrix at the conclusion of the 20th iteration, illustrating the maximum system reliability value along with the component reliability that achieves this value.

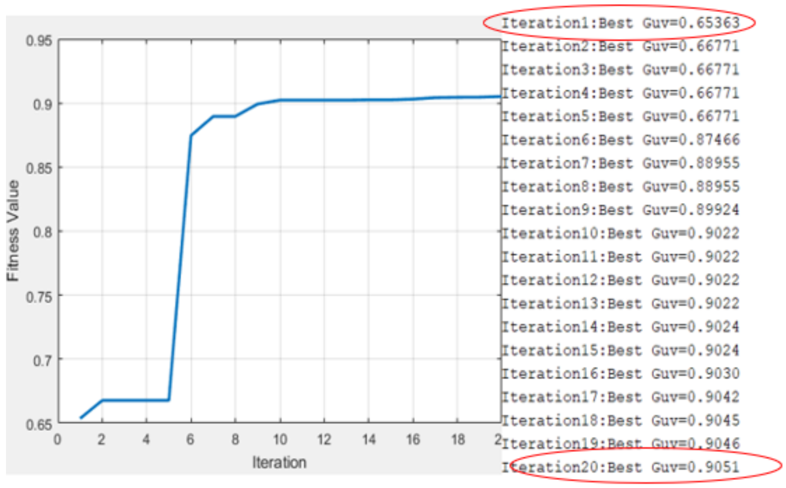

Figure 1. Reliability values at the end of each iteration with Matlab

$$\begin{bmatrix} \boldsymbol{p_1} & \boldsymbol{p_2} & \boldsymbol{p_3} & \boldsymbol{p_4} \\ 0{,}994990 & 0{,}808763 & 0{,}821855 & 0{,}641420 \\ 0{,}992794 & 0{,}809498 & 0{,}824382 & 0{,}641229 \\ 0{,}993171 & 0{,}809434 & 0{,}824236 & 0{,}641185 \\ 0{,}991060 & 0{,}810052 & 0{,}826257 & 0{,}641109 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R(p)} \\ 0{,}903363 \\ 0{,}904402 \\ 0{,}904372 \\ 0{,}905161 \end{bmatrix}$$

4.3 Genetic algorithm (GA)

GA was first proposed by John Holland in 1975. This algorithm, which is inspired by the phenomena observed in nature and evolutionary theory, ensures the survival of individuals by producing the best result in each iteration according to the purpose of the problem in optimization problems. It provides an advantage by being used to offer the best possible real solution in problems where the problem to be solved is not expressed mathematically, and there is no definite solution.

In order to understand GA, it is necessary to first master some basic concepts mentioned in the following figure.

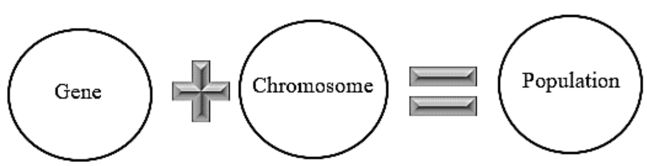

Figure 2. GA basic concepts

Each component in the population created according to the problem to be solved represents genes. Several sequences formed by the combination of genes are called chromosomes (individuals). The whole genes and chromosomes that come together to solve the problem is called a population.

Genetic Algorithms consist of a random initial population of candidate solutions and iteratively rely on biologically inspired operators to arrive at more suitable solutions. (Papazoglou and Biskas 2023).

1. Initial Population: The population size that needs to be taken depends on the complexity of the problems. Keeping the population size too large or too small generally prevents the algorithm's performance from working efficiently.
2. Conformity Value: This value can be derived from a fitness function based on the purpose of the problem or from the purpose function of the problem. Thanks to this function, it is ensured that the best value in the population is transferred to the next generations. These new chromosomes, calculated at each iteration, replace the old chromosomes and provide the best result.

3. Selection Process: In the selection process, new “good” parents (chromosomes) are selected from the population whose conformity to the environment is evaluated. The word good in this statement varies according to the purpose of the problem. There are several methods in the selection process. These methods are as follows.
4. Random Selection: Individuals are randomly selected.
5. Roulette Wheel Selection: This method of selection corresponds to the classical definition of probability. The probability of a chromosome appearing in the next generation is expressed as the ratio of the conformity value of the same chromosome to the total conformity value of the chromosomes in the population.

As a result of all these operations, the cycle continues until the termination criterion is met. When the desired goal is achieved, the termination criterion presents the best goal function value created for problem-solving as the closest and best solution output to the actual optimal answer in the solution space.

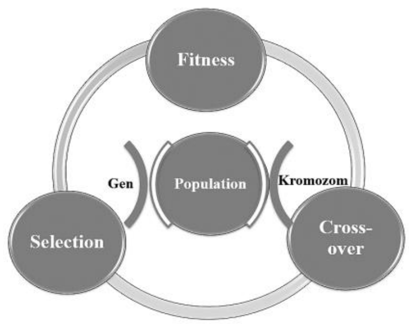

Figure 3. GA general cycle

5. RELIABILITY OPTIMIZATION IN K-OUT-OF-N SYSTEMS

This study is designed as quantitative research to optimize the reliability of lin/k/n and lin/con/k/n systems using TLBO and GA meta-heuristic algorithms. The research aims to evaluate the applicability of these algorithms to specific system structures by using parameters such as population size, number of iterations, crossover rate, and mutation rate. These parameters were selected based on previous studies in the literature, tailored to the complexity of the systems. In the TLBO algorithm, the assumption that the knowledge levels of students are independent is aligned with the independence of the components in the analyzed systems. In the GA algorithm, the reliability of each component was optimized through crossover and mutation processes. A fair comparison was conducted by applying the same system structures and parameters for both algorithms, and the results obtained for different population sizes and iteration numbers were analyzed. The novelty of this study lies in demonstrating the performance of TLBO and GA in complex system structures, thereby filling knowledge gaps in the existing literature.

The data used in this study was generated by a combination of mathematical modeling and simulation. The reliability optimization of the lin/k/n:F, lin/k/n:G, lin/con/k/n:F and lin/con/k/n:G systems was performed by applying TLBO and GA algorithms. System models were created based on predefined reliability structures, and simulations were performed to evaluate the performance of the algorithms under different parameter settings (e.g., population size, number of iterations, crossover rate and mutation rate). The parameters were selected according to the references in the literature and experimental arrangements and adapted to ensure the consistency and validity of the results. The outputs obtained from each simulation (e.g., reliability values for each system configuration) were recorded systematically. These outputs constituted the data set used to compare the performance of the TLBO and GA algorithms in terms of efficiency, accuracy and optimization ability.

In the previous section, it was stated how the TLBO algorithm was applied to a small system. In this section, the TLBO Algorithm and GA methods are used and compared with the meta-heuristic methods for reliability optimization of lin/k/n:F system and lin/con/k/n:F system for larger systems. The reliability functions obtained from the structure-function of these systems are determined as the purpose function required for system reliability optimization. The goal is to optimize system reliability and find component reliabilities that give the highest system reliability value. For this, the results calculated using the MATLAB R2016b program were compared with both algorithm methods. For the study, the Genetic Algorithm was also designed as a Real Coded Genetic Algorithm.

Table 1: Definitions required for TLBO and GA methods

TLBO	GA
Purpose Function: $\boldsymbol{max(R(p))}$	Purpose Function: $max(R(p))$
Population $\boldsymbol{P_n}$: 10, 50, 100	Population P_n: 10, 50, 100
Number of Variables $\boldsymbol{D_n}$: 10	Number of Variables D_n: 10
Number of Iterations $\boldsymbol{İ_n}$: 20, 50, 100	Number of Iterations $İ_n$: 20, 50, 100
Variable Range: $\boldsymbol{0 \leq p_1, p_2, \ldots, p_{10} \leq 1}$	Variable Range: $0 \leq p_1, p_2, \ldots, p_{10} \leq 1$
	Cross Ratio: 0,7; 0,8; 0,9
	Mutation Rate: 0,1; 0,2
	Cross Type: Arithmetic Cross
	Mutation Type: Normal Dispersed Mutation
	Selection Method: Random

Table 1 provides some definitions for use in the TLBO and GA methods. Comparisons of the lin/2/10:F and lin/con/2/10:F systems will be presented. The purpose functions of both types of systems are to maximize system reliability. The number of populations required for the initial population was first taken as 10, then 50, and then 100 and created from the same data set to be compared. The variable numbers are constant for both methods and are taken as 10. The number of iterations was the same for both algorithms, and trials were performed as 10, 20, 50 and 100, respectively. Because of component reliability, the upper and lower limit values are between 0 and 1. In the GA method, in addition to TLBO, the cross rate was taken as 0,7, 0,8 and 0,9, and the mutation rate was formed from smaller values as opposed to the cross rate. The cross-type, arithmetic cross, and the type of mutation are presented as normal dispersive mutations. In the selection method, the selection procedures were carried out randomly.

5.1 Optimization and comparison of the lin/2/10:F system with TLBO and GA

When the TLBO and GA methods were used to calculate the reliability of a lin/2/10:F system, the randomly generated initial component reliability for the system under consideration was taken as the same for both methods. The purpose function of the system is given as R_p system reliability.

$$\Phi(x) = (X_1 \cup X_2)(X_1 \cup X_3)(X_1 \cup X_4) \ldots (X_9 \cup X_{10}) \ ; \ \ 1 \leq i \leq n \tag{20}$$

$$\Phi(x) = (X_1 + X_2 - X_1X_2)(X_1 + X_3 - X_1X_3) \ldots \ (X_9 + X_{10} - X_9X_{10}) \tag{21}$$

$$R_p = (p_1 + p_2 - p_1p_2)(p_1 + p_3 - p_1p_3) \ldots \ (p_9 + p_{10} - p_9p_{10}) \tag{22}$$

<u>Initial component reliability</u>

$\boldsymbol{p_1}$	$\boldsymbol{p_2}$	$\boldsymbol{p_3}$	$\boldsymbol{p_4}$	$\boldsymbol{p_5}$	$\boldsymbol{p_6}$	$\boldsymbol{p_7}$	$\boldsymbol{p_8}$	$\boldsymbol{p_9}$	$\boldsymbol{p_{10}}$		$\boldsymbol{R(p)}$
0,742494	0,370786	0,207092	0,951109	0,663568	0,951759	0,507471	0,149387	0,204428	0,991631		0,071139
0,952521	0,523495	0,114473	0,003138	0,750896	0,323302	0,493768	0,506119	0,233175	0,722217		0,009981
0,428517	0,068051	0,493814	0,434693	0,963063	0,467606	0,934832	0,750566	0,172260	0,162045		0,039062
0,483397	0,707022	0,331470	0,426662	0,591464	0,549272	0,187015	0,047393	0,186121	0,200941		0,002952
0,970599	0,229230	0,209536	0,229554	0,846114	0,844191	0,089506	0,125205	0,629511	0,173823	=	0,010521
0,047987	0,803769	0,301624	0,266876	0,823950	0,679397	0,873601	0,770902	0,598692	0,985864		0,236740
0,700809	0,444155	0,978896	0,809006	0,904185	0,267142	0,797172	0,646386	0,155901	0,747468		0,326738
0,661753	0,558728	0,697705	0,304829	0,556989	0,505504	0,151534	0,649512	0,931877	0,381886		0,120006
0,405882	0,244154	0,885121	0,517032	0,731639	0,738188	0,864595	0,411887	0,738064	0,482645		0,249649
0,787814	0,529531	0,459164	0,718206	0,134419	0,921456	0,466229	0,714487	0,691321	0,905117		0,288137

For the TLBO algorithm, the mean vector M_p of each variable is calculated to be used in the intermediate operations. Then the line (student) that gives the highest system reliability value is again recorded as the best teacher to be used in intermediate operations.

Average vector and best teacher

M_p	0,618177	0,447892	0,467890	0,466110	0,696629	0,624782	0,536572	0,477184	0,454135	0,575364
$P_{teacher}$	0,700809	0,444155	0,978896	0,809006	0,904185	0,267142	0,797172	0,646386	0,155901	0,747468

Then, after the teacher and the student phases have been implemented, the 1st iteration ends. These operations are made and presented with the MATLAB R2016b program. The system and component reliability values at the end of 1st iteration are as follows**.**

The component and system reliability values at the end of the 1st and 20th iterations for the TLBO method are as follows:

Component and system reliability at the end of TLBO 1st iteration (lin/2/10: F)

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}		$R(p)$
0,783254	0,513559	0,433803	0,741638	0,187657	0,924505	0,470378	0,657632	0,642334	0,913821		0,000991
0,788550	0,5132880	0,434593	0,676185	0,284391	0,813994	0,496694	0,648047	0,558293	0,904352		0,000565
0,296965	0,876808	0,471117	0,387446	0,833316	0,175373	0,830885	0,616660	0,883658	0,077162		0,000008
0,564017	0,609547	0,571547	0,568442	0,707427	0,444653	0,413272	0,269510	0,174915	0,403603		0,000000
0,646124	0,823396	0,897216	0,897216	0,492966	0,823396	0,920809	0,823396	0,823396	0,823396	=	0,159222
0,897216	0,823396	0,897216	0,080121	0,892865	0,635507	0,880323	0,823396	0,494643	0,823396		0,031731
0,873459	0,576589	0,823396	0,823396	0,823396	0,003636	0,776683	0,796947	0,897216	0,914507		0,040532
0,909988	0,342278	0,851896	0,248608	0,368951	0,730154	0,897216	0,671867	0,964689	0,589247		0,003021
0,423064	0,144356	0,950429	0,537474	0,715600	0,826969	0,869913	0,379585	0,715097	0,546609		0,000462
0,850726	0,433108	0,661396	0,476394	0,255188	0,822948	0,688159	0,692541	0,832088	0,742465		0,002220

Component and system reliability at the end of TLBO 20th iteration (lin/2/10: F)

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}		$R(p)$
0,863754	0,899082	0,998959	0,746828	0,380390	0,991213	0,870852	0,893722	0,641606	0,969547		0,281128
0,935477	0,886858	0,996837	0,889712	0,333075	0,961940	0,857699	0,906798	0,626322	0,930404		0,321051
0,936207	0,877815	0,999624	0,907604	0,248593	0,999892	0,855407	0,912156	0,618738	0,970484		0,336277
0,977144	0,858505	0,999071	0,985599	0,257876	0,988317	0,848413	0,922237	0,613351	0,946785		0,375622
0,978295	0,858143	0,996973	0,974066	0,345021	0,958695	0,849864	0,917659	0,620446	0,913768	=	0,368294
0,977536	0,858459	0,996987	0,972835	0,344677	0,958781	0,849989	0,917637	0,620481	0,914035		0,367466
0,940413	0,882251	0,996943	0,901976	0,337317	0,960034	0,856597	0,908967	0,625758	0,926041		0,326358
0,905106	0,865244	0,997842	0,845635	0,344219	0,983309	0,861438	0,901709	0,620552	0,970105		0,304577
0,940064	0,876616	0,998581	0,910836	0,289092	0,982653	0,855397	0,911860	0,621364	0,950251		0,334644
0,938495	0,842998	0,998194	0,909422	0,322705	0,993609	0,855282	0,908741	0,612170	0,975350		0,340886

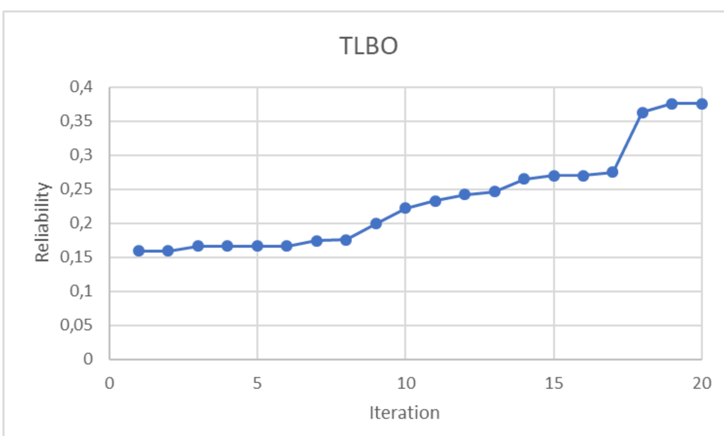

Figure 4. Best system reliability graph for each TLBO iteration (lin/2/10: F)

In the TLBO algorithm, the best system reliability value at the end of the 1st iteration was 0,159222, while the system reliability value increased to 0,375622 at the end of the 20th iteration. In addition, the reliability values that each component must have to give the highest reliability of the system are also seen. In Figure 4, the best system reliability values for each iteration result are shown in the graph.

The initial component reliability for GA is the same, and the component and system reliability at the end of the 1st and 20th iterations is found as follows:

Component and system reliability at the end of GA 1st iteration (lin/2/10: F)

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	$R(p)$
0,732247	0,662904	0,695904	0,901689	0,895688	0,923036	0,666904	0,856904	0,896904	0,934729	0,223593
0,736055	0,666904	0,890002	0,857425	0,670064	0,666904	0,694605	0,666904	0,666904	0,804369	0,035358
0,776352	0,666904	0,817536	0,726887	0,666904	0,835103	0,673679	0,677449	0,666904	0,778281	0,032199
0,666904	0,666904	0,827629	0,669040	0,666904	0,867180	0,694256	0,666904	0,716630	0,736235	0,024079
0,712270	0,666904	0,666904	0,800324	0,666904	0,666904	0,666904	0,683423	0,666904	0,874303	0,017781
0,702293	0,666904	0,666904	0,718176	0,666904	0,813504	0,666904	0,688441	0,727931	0,745550	0,017325
0,747273	0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,675557	0,895806	0,666904	0,013846
0,666904	0,666904	0,666904	0,668228	0,717533	0,792463	0,666904	0,670970	0,712754	0,666904	0,010497
0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,666904	0,005031
0,700809	0,444155	0,978896	0,809900	0,904185	0,267142	0,797172	0,646386	0,155901	0,747468	0,001804

Component and system reliability at the end of GA 20th iteration (lin/2/10: F)

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	$R(p)$
0,784425	0,784425	0,799167	0,897878	0,804811	0,897878	0,784425	0,784425	0,806279	0,897878	0,245500
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,792950	0,784425	0,806279	0,897878	0,243040
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,792950	0,784425	0,806279	0,897878	0,243040
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,784425	0,784425	0,806279	0,897878	0,239703
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,784425	0,784425	0,806279	0,897878	0,239703
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,784425	0,784425	0,806279	0,897878	0,234885
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,784425	0,784425	0,784425	0,897878	0,231245
0,784425	0,784425	0,784425	0,897878	0,804811	0,897878	0,784425	0,784425	0,784425	0,897878	0,229068
0,784425	0,784425	0,784425	0,897878	0,784425	0,897878	0,784425	0,784425	0,784425	0,897878	0,223521
0,784425	0,799210	0,784425	0,784425	0,784425	0,784425	0,784425	0,795720	0,798301	0,784425	0,127351

According to the mutation rate of 0,1 in GA and the cross rate of 0,7, the best system reliability value at the end of the 1st iteration was 0,22359289, while the system reliability value increased to 0,245500 at the end of the 20th iteration. Figure 5 gives the best system reliability values for GA at the end of each iteration. In addition, the reliability values that each component must have to give the highest reliability are also seen.

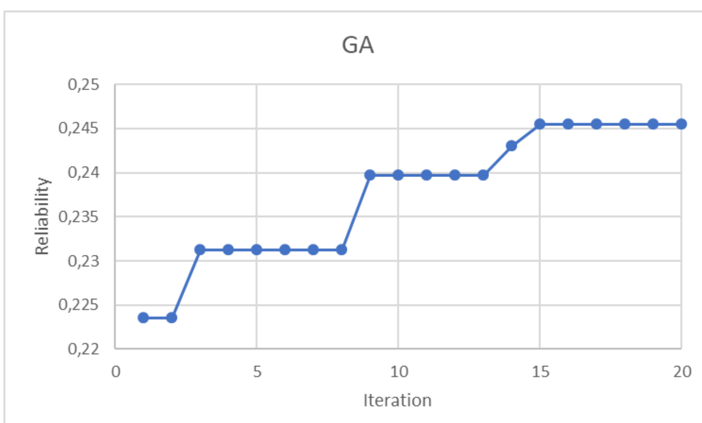

Figure 5. Best system reliability graph for each GA iteration (lin/2/10: F)

The graph with the results in comparison for the TLBO and GA methods applied for the reliability of a lin/2/10:F system is shown in Figure 6.

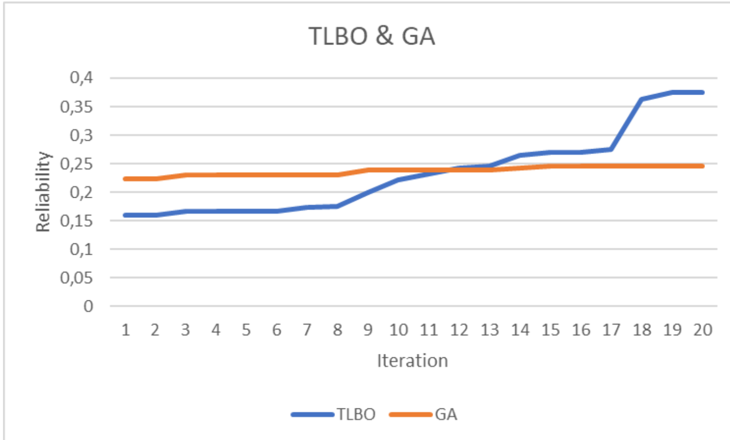

Figure 6. TLBO and GA comparison chart for 20 iterations (lin/2/10: F)

In Figure 6, the best system reliability values in the TLBO and GA methods are shown on a single graph. Accordingly, it is seen that the TLBO method had a higher system reliability value at the end of the 20th iteration than the GA method.

5.2 Optimization and comparison of the lin/con/2/10:F system with TLBO and GA

The reliability of a lin/con/2/10:F system was calculated by TLBO and GA methods. The randomly generated initial component reliability for the system under consideration is the same for both methods. The purpose function of the system is given as R_p system reliability.

$$\Phi(x) = (X_1 \cup X_2) \cap (X_2 \cup X_3) \cap (X_3 \cup X_4) \cap \dots \cap (X_9 \cup X_{10}) \;;\; 1 \le i \le n \tag{23}$$

$$R_p = (p_1 + p_2 - p_1 p_2)(p_2 + p_3 - p_2 p_3) \dots (p_9 + p_{10} - p_9 p_{10}) \tag{24}$$

The component and system reliability values at the end of the 1st and 20th iterations for the TLBO method are as follows:

Component and system reliability at the end of TLBO 1st iteration (lin/con/2/10: F)

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}		$R(p)$
0,395989	0,587513	0,261933	0,613574	0,746917	0,810139	0,694612	0,463048	0,397440	0,991273		0,169907
0,728335	0,550660	0,564153	0,235745	0,601391	0,463782	0,229901	0,616677	0,771883	0,459818		0,085219
0,493622	0,082789	0,784079	0,622043	0,847222	0,335506	0,868804	0,758009	0,460717	0,311755		0,161374
0,610658	0,582219	0,299620	0,375176	0,657981	0,626307	0,161545	0,067718	0,301939	0,193858		0,005230
0,838644	0,446022	0,389747	0,582319	0,332330	0,899970	0,361468	0,550617	0,674133	0,701756	=	0,155471
0,406973	0,837330	0,266815	0,271003	0,853460	0,707986	0,958950	0,791351	0,575728	0,869620		0,267127
0,737702	0,387656	0,869620	0,986543	0,869620	0,120162	0,869620	0,883195	0,096138	0,974023		0,518832
0,655755	0,559455	0,709736	0,311052	0,552989	0,509262	0,144474	0,652470	0,946290	0,374866		0,123629
0,417712	0,235786	0,913949	0,558687	0,752933	0,693851	0,892252	0,438164	0,695875	0,513895		0,263696
0,802034	0,428544	0,766904	0,929573	0,583253	0,360975	0,766396	0,889660	0,243773	0,923154		0,385272

The system and component reliability at the end of the 20th iteration for the TLBO algorithm is shown in the figure below:

Component and system reliability at the end of TLBO 20th iteration (lin/con/2/10: F)

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	$R(p)$
0,804084	0,746183	0,554577	0,949365	0,330643	0,998546	0,310321	0,998808	0,644367	0,786635	0,733148
0,800786	0,746042	0,557883	0,949433	0,333503	0,993708	0,312876	0,998194	0,641494	0,788178	0,728104
0,803842	0,747055	0,554938	0,949838	0,329547	0,999467	0,309311	0,999630	0,644752	0,787076	0,735740
0,803283	0,745465	0,555756	0,949562	0,330512	0,998618	0,311029	0,998347	0,645231	0,786580	0,732922
0,803073	0,746660	0,555844	0,949403	0,331297	0,997002	0,310817	0,998916	0,643285	0,787399	0,732145
0,789860	0,748565	0,573241	0,953559	0,328499	0,991273	0,313809	0,999714	0,643071	0,795946	0,735435
0,804068	0,747138	0,554615	0,949883	0,329274	0,999931	0,309001	0,999773	0,644972	0,786963	0,736356
0,802096	0,745875	0,556965	0,949854	0,330906	0,997404	0,311332	0,998561	0,644346	0,787411	0,732502
0,803174	0,745898	0,555742	0,949720	0,330197	0,998864	0,310579	0,998734	0,645173	0,786782	0,733831
0,804812	0,747617	0,552812	0,948948	0,331453	0,997846	0,309317	0,999759	0,642442	0,786708	0,733003

In the TLBO algorithm, the best system reliability value at the end of the 1st iteration was 0,518832, while the system reliability value increased to 0,736356 at the end of the 20th iteration.

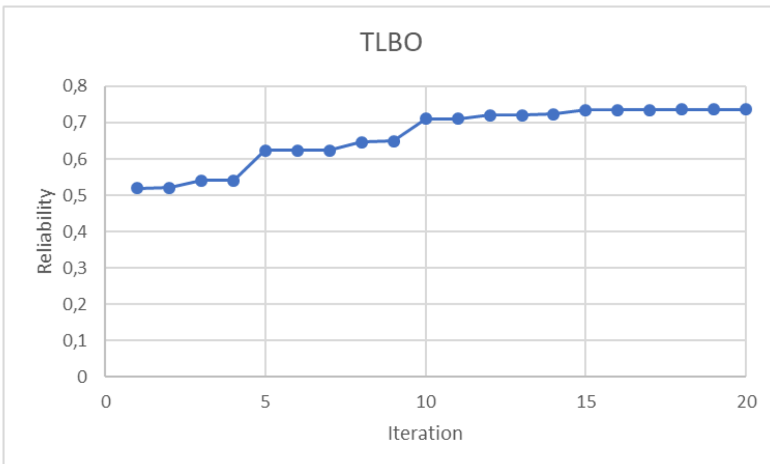

Figure 7. Best system reliability graph for each TLBO iteration (lin/con/2/10: F)

Figure 7 gives the best system reliability values in TLBO at the end of each iteration. As can be seen, as a result of each iteration, the system has reached its best result by increasing the reliability value.

According to the initial component reliability considered for GA, the highest value of system reliability is recorded as the best chromosome in GA. Accordingly, the component and system reliability at the end of the 1st and 20th iterations are as follows:

Component and system reliability at the end of GA 1st iteration (lin/con/2/10: F)

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	$R(p)$
0,700809	0,444155	0,978896	0,809006	0,904185	0,267142	0,797172	0,646386	0,155901	0,747468	0,326738
0,787814	0,529531	0,459164	0,718206	0,134419	0,921456	0,466229	0,714487	0,691321	0,905117	0,288138
0,405882	0,244154	0,885121	0,517032	0,731639	0,738188	0,864595	0,411887	0,738064	0,482645	0,249648
0,047987	0,803769	0,301624	0,266876	0,823950	0,679397	0,873601	0,770902	0,598669	0,985864	0,236740
0,699652	0,444155	0,699652	0,699652	0,699652	0,267142	0,699652	0,646386	0,155901	0,699652	0,163525
0,661753	0,558728	0,697705	0,304829	0,556989	0,505504	0,151534	0,649512	0,931877	0,381886	0,120006
0,742494	0,370786	0,207092	0,951109	0,663568	0,951759	0,507471	0,149387	0,204428	0,991631	0,071139
0,428517	0,068051	0,493814	0,434693	0,963063	0,467606	0,934832	0,750566	0,172260	0,162045	0,039061
0,970599	0,229230	0,209536	0,229554	0,846114	0,844191	0,089506	0,125205	0,629511	0,173823	0,010521
0,952521	0,523495	0,114473	0,003138	0,750896	0,323302	0,493768	0,506119	0,233175	0,722217	0,009931

Component and system reliability at the end of GA 20th iteration (lin/con/2/10: F)

p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}	$R(p)$
0,762958	0,531473	0,800691	0,800691	0,800691	0,385267	0,797172	0,646386	0,385267	0,747468	0,350311
0,700809	0,444155	0,800691	0,800691	0,800691	0,385267	0,797172	0,712574	0,385267	0,747468	0,343889
0,700809	0,531473	0,800691	0,800691	0,800691	0,385267	0,797172	0,646386	0,385267	0,747468	0,338836
0,745881	0,531473	0,800691	0,800691	0,800691	0,385267	0,797172	0,646386	0,385267	0,697355	0,334498
0,700809	0,531473	0,800691	0,800691	0,800691	0,385267	0,797172	0,663973	0,385267	0,697355	0,332262
0,700809	0,468345	0,800691	0,800691	0,800691	0,385267	0,800691	0,646386	0,385267	0,747468	0,328040
0,700809	0,531473	0,800691	0,800691	0,800691	0,385267	0,800691	0,646386	0,385267	0,697355	0,327725
0,700809	0,531473	0,800691	0,800691	0,800691	0,385267	0,800691	0,646386	0,385267	0,697355	0,327725
0,700809	0,444155	0,978896	0,800691	0,904185	0,267142	0,797172	0,646386	0,155901	0,747468	0,326738
0,700809	0,531473	0,800691	0,800691	0,800691	0,385267	0,797172	0,646386	0,385267	0,697355	0,326479

According to the mutation rate of 0,1 in GA and the cross rate of 0,7, the best system reliability value at the end of the 1st iteration was 0,326738, while the system reliability value increased to 0,350311 at the end of the 20th iteration. When the system reliability value is low, component reliability values are also observed to be lower. In Figure 8, the best system reliability values at the end of each iteration in GA are given in the graph. In Figure 9, the best system reliability values in TLBO and GA methods are presented in a single graph. Looking at the graph, it is seen that the TLBO method has a higher system reliability value at the end of the 20th iteration compared to the GA method.

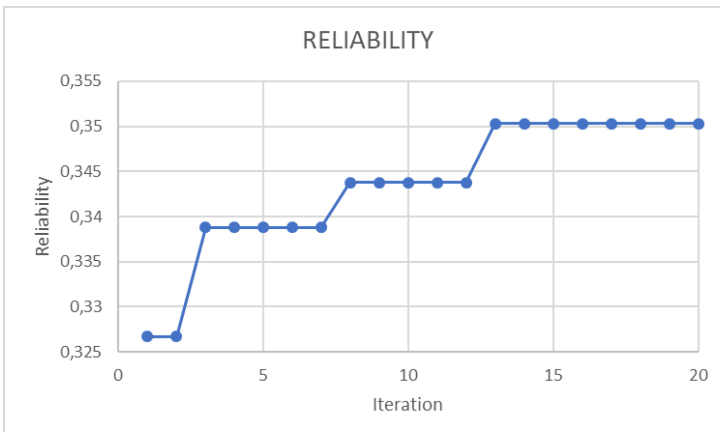

Figure 8. Best system reliability graph for each GA iteration (lin/con/2/10: F)

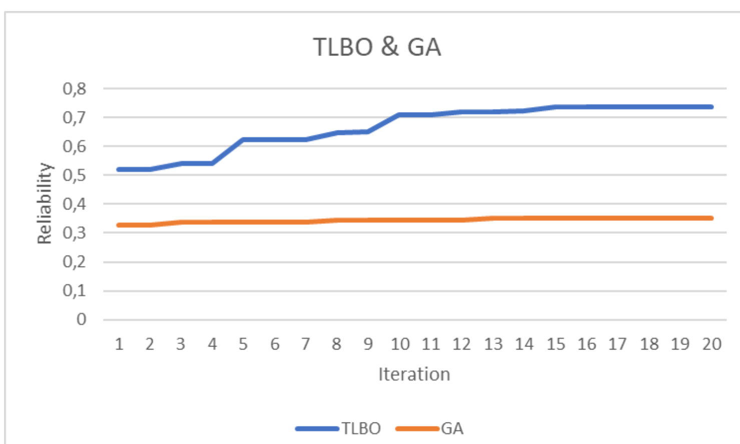

Figure 9. TLBO and GA comparison chart for 20 iterations (lin/con/2/10: F)

When the population size and the number of iterations are increased to 50, the comparison of the lin/2/10:F system and lin/con/2/10:F system with the TLBO and GA method is as in Figures 10 and 11, respectively. Here, the cross and mutation rates considered in the GA method for both systems are 0,8 and 0,2, respectively.

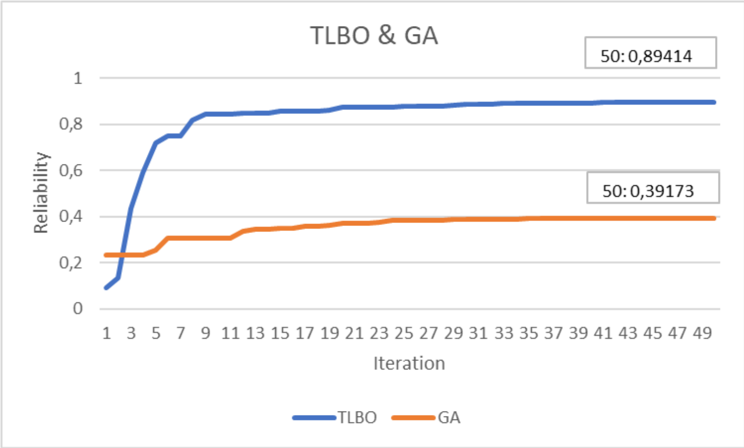

Figure 10. TLBO and GA comparison chart for 50 iterations and population (lin/2/10: F)

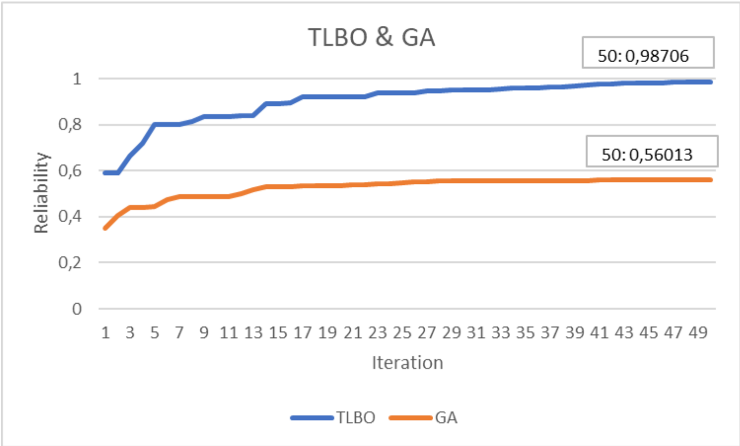

Figure 11. TLBO and GA comparison chart for 50 iterations and population (lin/con/2/10: F)

Looking at Figures 10 and 11, it is seen that when the population size and the number of iterations are increased, the system reliability values increase in both the TLBO method and the GA method according to the situation where the population size is 10, and the number of iterations is 20. As a result, it can be said that at the end of the 50th iteration, the TLBO method gives a higher system reliability value for both systems than the GA method.

Table 2: System comparisons according to TLBO and GA methods

Systems	Population	Iteration	Cross Ratio (for GA)	Mutation Rate (for GA)	TLBO	GA
lin/2/10:F	10	20	0,7	0,1	0,375622	0,24550
	50	50	0,8	0,2	0,89414	0,39173
	100	100	0,9	0,2	0,96049	0,55797
lin/2/10:G	10	20	0,7	0,1	1,00000	1,00000
	50	50	0,8	0,2	1,00000	1,00000
	100	100	0,9	0,2	1,00000	1,00000
lin/con/2/10:F	10	20	0,7	0,1	0,736356	0,350311
	50	50	0,8	0,2	0,98706	0,56013

Systems	Population	Iteration	Cross Ratio (for GA)	Mutation Rate (for GA)	TLBO	GA
	100	100	0,9	0,2	0,99993	0,66388
lin/con/2/10:G	10	20	0,7	0,1	0,99995	0,99500
	50	50	0,8	0,2	1,00000	0,99911
	100	100	0,9	0,2	1,00000	0,99948
lin/con/3/10:F	10	20	0,7	0,1	0,91405	0,79548
	50	50	0,8	0,2	0,99993	0,86531
	100	100	0,9	0,2	0,99999	0,96291
lin/con/3/10:G	10	20	0,7	0,1	0,99985	0,97136
	50	50	0,8	0,2	0,99999	0,99227
	100	100	0,9	0,2	1,00000	0,99987

Looking at Table 2, the reliability of lin/k/n:F and G and lin/con/k/n:F and G systems was maximized under the given definitions and the component reliability that gives the highest value of system reliability was calculated and compared with TLBO and GA methods. System and component reliability calculated with different population sizes and different iteration values were found to give higher system reliability values in TLBO than in the GA method. At the same time, the cross and mutation rates in the literature for the GA method were also taken into consideration. When the results of lin/2/10:F systems and lin/con/2/10:F systems in both methods were examined, it was seen that the lin/con/2/10:F system gave higher system reliability values. This can be interpreted as the probability of failing two consecutive components out of 10 than the probability of failing two of the 10 components randomly. Comparisons of the lin/2/10:F system and the lin/con/3/10:F system were also made. As a result of the comparison, it was observed that the reliability values of the lin/con/3/10:F system are higher in both methods than the lin/con/2/10:F system. This can be interpreted as the fact that three components are less likely to fail in a row than if two components fail consecutively. At the same time, lin/con/2/10:G and lin/con/3/10:G system and a lin/2/10:G system are calculated according to both methods and presented in Table 2. When the system reliability of *G* systems is examined, it is seen that very close to 1 and 1 results are obtained. This is mainly related to the k value. For *G* systems, if at least k or more components work successfully, the system will also succeed can be interpreted as a very high probability that 2-3 components out of 10 components will succeed in the system.

It is shown how the reliability optimization is calculated for the example system considered in the study step by step. Considering a real-life application of a similar situation; an example of electricity distribution systems can be considered. Today, electricity distribution systems are complex and critically important systems. Understanding and analyzing the relationships between components in these systems is crucial for their stability and reliability. Therefore, the electric transformers used in these systems are electromechanical devices used to transmit electricity from one circuit to another. They are generally used to convert electricity from high-voltage transmission lines to lower voltage levels. The arrangement of electric transformers is usually done in parallel. This ensures more reliable and effective transmission of energy. The parallel connection enables load sharing between transformers and provides a more balanced energy flow. Moreover, in case of a transformer failure, the capacity of other transformers to maintain the system increases. This article discusses the concept of linear k-out-of-n and linear consecutive k-out-of-n F and G systems and relates them to real-life applications such as electric transformers. For example, analyzing how the system would be affected and potentially rendered inoperative if multiple electric transformers in a region were to fail consecutively could provide valuable insights into the reliability and resilience of electricity distribution systems.

6. CONCLUSION

In this study, comparisons of the results of the different population sizes of the lin/k/n:F and G systems and lin/con/k/n:F and G systems and the reliability values of TLBO and GA optimization methods for different iteration numbers, cross rate and mutation rates are given. As can be seen from the results, the highest reliability values for all the systems considered were revealed by the TLBO method. In addition, the reliability values that each component must have to give the highest reliability were also obtained. When we look at the methods in general, it is possible to say that the TLBO method reaches optimum values in a shorter time than GA and has a more positive effect on the overall system reliability. Although both methods are meta-heuristics inspired by nature and give the best answer for the desired purpose rather than scanning the entire set of solutions to solve the problem, it can be said that the TLBO method is more advantageous than the GA method. The reason is that the TLBO method has a more understandable and easier process simplicity than GA, as well as GA has many more constraints and includes many criteria to consider.

In conclusion, it can be said that in the reliability optimization problems of lin/k/n systems and lin/con/k/n systems, which are considered special cases of serial and parallel systems, the use of meta-heuristic algorithms provides more benefit in terms of time and cost in finding the closest and possibly the best solution to the real answer in the solution space due to the complex structure of the systems.

The results show that TLBO and GA are valuable tools for achieving optimal reliability values in complex system structures. TLBO's ability to reach optimum values more quickly and with a simpler process makes it particularly significant for applications requiring fast and cost-effective solutions. In this context, the novelty of the study is in demonstrating the applicability of TLBO to optimize complex structures and how the comparison with GA contributes to the existing knowledge in the literature.

This study examines the performance of TLBO and GA algorithms in the reliability optimization of lin/k/n and lin/con/k/n systems. However, there are some challenges with the developed techniques and limitations in the study. First, the inherent randomness of meta-heuristic algorithms can introduce variability in the results. This can affect the consistency of the outcomes and lead to differences in results when different parameters are tested. Additionally, the parameter settings and initial conditions of the algorithms have a significant impact on the accuracy of the results, and this limitation means the algorithms do not guarantee an optimal solution in all cases. Future studies could examine the performance of the algorithms across a broader range of systems, providing more comprehensive results. Furthermore, comparing other meta-heuristic algorithms could offer more insight into the advantages and challenges of TLBO and GA. A more detailed analysis of parameter sensitivity and the performance of the algorithms could help develop more robust strategies for reliability optimization.

REFERENCES

Baladeh, A.E. and Taghipour S. 2022. "Reliability Optimization of Dinamic k-out-of-n Systems with Competing Failure Modes". *Reliability Engineering & System Safety* Volume 227.

Cao, R., Hou, W. and Gao, Y. 2018. "An Entropy-Based Three-Stage Approach for Multi-Objective Systems Reliability Optimization Considering Uncertainty". *Engineering Optimization* 50(9): 1453-1469.

Coit, D.W. and Liu, J. 2000. "System Reliability Optimization with k-out-of-n Subsystems". *International Journal of Reliability, Quality and Safety Engineering* 7(2): 129-142.

Coit, D.W., and Zio, E. 2019. "The evolution of system reliability optimization". *Reliability Engineering and System Safety* 192: 1-11.

Eryılmaz, S. 2017. "A Note on Optimization Problems of a Parallel Systems with a Random Number of Units". *International Journal of Reliability, Quality and Safety Engineering* 24(5).

Gökdere, G. and Gürcan, Y. 2016. "Mühendislik Uygulamalarında Kullanılan Ardışık *n*'den *k* Çıkışlı Sistemlerin Güvenilirlik Analizi". *Afyon Kocatepe Üniversitesi Fen ve Mühendislik Bilimi Dergisi* 461-467.

Kumar. R., Tewari. P.C. and Khanduja. D. 2018. "Parameters Optimization of Fabric Finishing System of a Textile Industry using Teaching-Learning-Based Optimization Algorithm". *International Journal of Industrial Engineering Computations*: 221-234.

Kuo, W., & Zuo, M. J. 2003. *Optimal Reliability Modeling: Principles and Applications*, John Wiley & Sons: Haboken.

Levitin, G. 2005. The Universal Generating Function in Reliability Analysis and Optimization, The State University of New Jersey 96 Frelinghuysen Road Piscataway.

Linh, N.T. 2023. "A Novel Combination of Genetic Algorithm, Particle Swarm Optimization, and Teaching-Learning Based Optimization for Distribution Network Reconfiguration in Case of Faults". *Engineering, Technology & Apply Science Research 14(1):* 12959-12965.

Nakamura, T. and Yamamoto, H. 2022. "Algorithm for Computing the System Signature of Linear and Circular Connected –(*1*,*2*)-or-(*2*,*1*)-out-of(*m*,*n*):F Lattice Systems". *International Journal of Reliability, Quality and Safety Engineering* 29(04).

Papazoglou, G. and Biskas, P. 2023. Review and Comparison of Genetic Algorithm and Particle Swarm Optimization in the Optimal Power Flow Problem, *Energies* 16,1152.

Peiravi, A., Nourelfath, M. and Zanjani, M.K. 2022. “Redundancy Strategies Assessment and Optimization of k-out-of-n Systems based on Markov Chains and Genetic Algorithm”. *Reliability Engineering & System Safety* Volume 221.

Peng, R. and Xiao, H. 2018. “Reliability of Linear Consecutive-k-out-of-n Systems With Two Change Points”. *IEEE Transactions on Reliability 67(3):*1019-1029.

Rao, R.V., Savsani, V.J. and Vakharia, D.P. 2011. “Teaching-Learning-Based Optimization: A Novel Method for Constrained Mechanical Design Optimization Problems”. *Computer -Aided Design* 43: 303-315.

Rao, R.V., Savsani, V.J. and Vakharia, D.P. 2012. “Teaching–Learning-Based Optimization: An Optimization Method for Continuous Non-Linear Large-Scale Problems”. *Information Sciences* 183: 1-15.

Rausand, M. and Hayland, A. 2004. *System Reliability Theory: Models, Statistical Methods and Applications*, John Wiley&Sons: Haboken.

Saleem, E.A. 2017. “Series-Parallel Systems Reliability Optimization Using Genetic Algorithm and Statistical Analysis”. *International Journal of Mechanical and Mechatronics Engineering* 11(5): 960-966.

Sharifi, M., Cheragh, G., Maljaii, K.D., Zaretalab, A. and Shahriari, M. 2021. “Reliability and Cost Optimization of a System with k-out-of-n Configuration and Choice of Decreasing the Component Failure Rates”. *Scientia Iranica* 28(6):3602-3616.

Shukla, A.K., Sanjeev, K.P. and Chauhan, S.S. 2019. An Empirical Evaluation of Teaching-Learning-Based Optimization, Genetic Algorithm and Particle Swarm Optimization, *International Journal of Computers and Applications*: Volume 45, 36-50.

Soltani, R. 2014. “Reliability Optimization of Binary State Non-Repairable Systems: A State-of-The-Art Survey”. *International Journal of Industrial Engineering Computations:* 339-364.

Sooktip, T., Wattanapongsakorn, N. and Coit, D.W. 2011. “System reliability optimization with *k*-out-of-*n* subsystems and changing k”. *The Proceedings of 2011 9th International Conference on Reliability, Maintainability and Safety* 1382-1387.

Taheri, S. I., Davoodi, M., & Ali, M. H. 2023. A Simulated-Annealing-Quasi-Oppositional-Teaching-Learning-Based Optimization Algorithm for Distributed Generation Allocation. *Computation*, *11*(11), 214.

Ullah, V., Mu’tasim, M.A.N. and Rashid, M.F.F. 2023. “Optimization of Cost-Based Hybrid Flowshop Scheduling Using Teaching-Learning-Based Optimization Algorithm”. *International Journal of Automotive and Mechanical Engineering 21(3):* 11616-11628.

Wang, N., Zhao, J., Jiang, Z. and Zhang, S. 2018. “Reliability Optimization of Systems with Component Improvement Cost based on Importance Measure”. *Mechanical Engineering* 10(11): 1-15.

Zhao, J.B., Zhang, S., Zhu, W.J. and Du, Li. 2019. “Reliability Optimization of Linear Consecutive *k*-out-of-*n*: *F* systems with Birnbaum Importance-based Quantum Genetic Algorithm”. *Mechanical Engineering* 11(4): 1-16.