Industrial Engineering Journal
ISSN: 0970-2555
Volume : 55, Issue 03 : 2026

RELIABILITY ANALYSIS OF A THREE-UNIT DEGRADABLE COMPUTING SYSTEM WITH COVERAGE FACTOR USING PARTICLE SWARM OPTIMIZATION

Dr. Kalika Patrai, Associate Professor, Department of Management, Jagan Institute of Management Studies, New Delhi, India

ABSTRACT
This paper provides a reliability analysis for a three-unit degenerative computer system with a coverage factor that works in a non-repairable environment. This system is made up of three processing units that operate in active redundancy. The system degrades gradually through operational states before complete failure. A Continuous-Time Markov Process will be used to formulate the probability model of the system. The Kolmogorov differential equations will then be solved to get the transient state probabilities. The analytical solutions for the reliability of the system and Mean Time to System Failure (MTTF) will be obtained in closed form. Particle Swarm Optimization (PSO) is utilized to optimize system reliability by minimizing failure rate and maximizing the coverage factor.

Keywords: Reliability, Degradable Computing System, Coverage Factor, Markov Process, Particle Swarm Optimization, MTTF.

I. Introduction and Literature Review
Modern computing systems such as distributed computing, cloud computing, industrial automation, communication systems, aerospace systems, and telecommunication networks require high reliability and uninterrupted operation because failures may cause severe operational losses, economic damage, and degradation of system performance. Redundancy techniques are therefore widely employed to improve reliability and fault tolerance in critical computing environments [1]. Reliability engineering has become one of the most significant research areas in computer systems, communication networks, and industrial systems because system failures may lead to catastrophic consequences in real-time applications.
In many real-world systems, failure of an individual component does not immediately lead to total system collapse. Instead, the system continues operating in degraded mode with reduced efficiency and performance. Such systems are known as degradable systems or gracefully degradable systems. Degradable systems are extensively used in cloud computing, multiprocessor systems, distributed architectures, and aviation control systems where uninterrupted service is essential [2]. Shama [3] has study the reliability measures of a degradable system in which switching failures and reboot delay are considered. Failure times and repair times of failed units are assumed to be exponentially distributed. Pandey and Tyagi [4] estimated profust reliability of a powerloom plant, which is modelled as a two unit gracefully degradable system. The Markov process has been widely adopted for modelling degradable systems because it efficiently handles state transitions, degradation behaviour, fault coverage, and stochastic failures. Uprety and Patrai [5] evaluated system characteristics (MTTF, reliability) of a two unit degradable system using Markov modelling approach, in which times to failure and times to repair of the operating units are, assumed to follow fuzzified exponential distribution.
Coverage factor is another important reliability parameter in fault-tolerant computing systems. The coverage factor represents the probability that the system successfully detects, isolates, and recovers from failed components. Higher coverage significantly improves system reliability and survivability. Imperfect coverage may cause catastrophic failure even when redundant units are available. Therefore, incorporation of coverage factor makes the reliability model more practical and realistic [6]. Uprety and Zaheeruddin [7] developed a fuzzy Markov model of gracefully degradable

redundant computing system to capture the effect of coverage factor and repair on its reliability. Patrai and Gupa [8] estimated the reliability of a degradable system with imperfect coverage and the concept of Weibull intuitionistic fuzzy set was used to deal with the uncertain information about system components.
Several researchers have contributed extensively in the field of reliability optimization and presented many efficient methods. Wei et al. [9] developed a mathematical model for condition-based maintenance optimization of multi-state systems. A continuous-time discrete-state Markov chain model is used to characterize the deterioration stochastic process. It considers a multi-state system with each discrete state characterized by a degradation level. Mellal and Zio [10] proposed adaptive Particle Swarm Optimization methods for multi-objective system reliability optimization and demonstrated the effectiveness of PSO in redundancy allocation problems. Particle Swarm Optimization (PSO), originally introduced by Kennedy and Eberhart [11], has become one of the most powerful metaheuristic optimization algorithms because of its simplicity, fast convergence, computational efficiency, and ability to solve nonlinear optimization problems effectively. PSO is inspired by the collective social behavior of bird flocking and fish schooling. Zavala et al. [12] proposed a particle swarm optimization (PSO) approach named PESDRO to solve a bi-objective redundant reliability problem; and the reliability redundant problems of series system; parallel system and K-out-of-N system are resolved. Arya and Choube [13] described a methodology for reliability allocation to various components of a system in which particle swarm optimization (PSO) has been used to get optimum solution. Malhotra and Negi [14] used PSO algorithm to estimate the parameters of Software Reliability Growth Models. Garg and Sharma [15] used PSO to solve multi-objective reliability redundancy allocation problem of a series system. Pnt et al. [16] gives the details of PSO development and its applications to reliability optimization. Abdullah and Hassan [17] examined the reliability of a complex system by optimizing the reliability of every component of the system using Particle Swarm algorithm. He et al. [18] proposed a reliability allocation methodology for power system based on the Particle Swarm Optimization (PSO) algorithm.
Motivated by the above research contributions, this paper presents a mathematical reliability model of a three-unit degradable computing system with coverage factor using Continuous-Time Markov Processes. Exact analytical expressions for transient state probabilities, system reliability, and Mean Time to System Failure (MTTF) are derived. Further, Particle Swarm Optimization is employed to maximize system reliability by optimizing system parameters such as failure rate and coverage factor.

II. System Description

We consider a gracefully degradable computing system with three identical and Independent modules. Each module has only two states: failed or functioning. The time to breakdown for each module follows an exponential distribution with parameter, λ. When failure comes to same module, the system immediately takes reconfiguration operation with negligible time, to remove logically the faulty module whereas other fault free modules continue to do their work if the reconfiguration operation is performed successfully.
The system states are defined in Table I and the system transitions from fully operational state toward degraded states and finally to failed state. The Markov model of the state transitions is given in Figure 1.

Table I. Description of states of the system

State	Description
0	All three units operational
1	Two units operational
2	One unit operational
3	Complete system failure

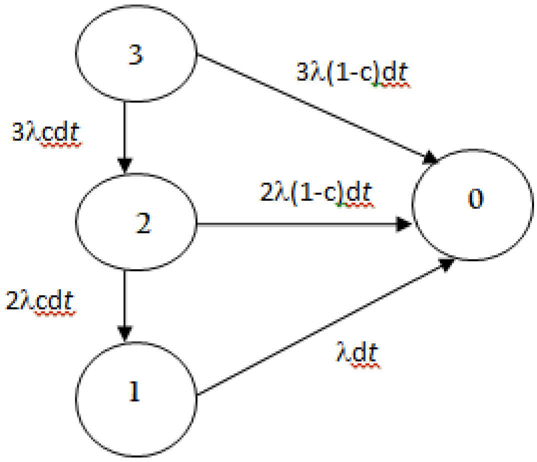

Figure 1: Markov model for a 3-unit gracefully degradable system.

The differential equations describing the system are:

$$\frac{dP_3}{dt} = -3\lambda P_3(t)$$

$$\frac{dP_2}{dt} = -2\lambda P_2(t) + 3c\lambda P_3(t)$$

$$\frac{dP_1}{dt} = -\lambda P_1(t) + 2\lambda c P_2(t)$$

$$\frac{dP_0}{dt} = \lambda P_1(t) + 2\lambda(1-c)P_2(t) + 3\lambda(1-c)P_3(t)$$

The system is initially in state S3, thus $P_3(0)=1$ *and* $P_2(0)=0, P_1(0)=0, P_F(0)=0$. Refereeing to the state transition diagram as shown in Figure 1, the system differential equations using Laplace transform are obtained and given by:

$$s\widetilde{P}_3(s) = -3\lambda\widetilde{P}_3(s)$$

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 55, Issue 03 : 2026

$$s\widetilde{P}_2(s) = -2\lambda\widetilde{P}_2(s) + 3c\lambda\widetilde{P}_3(s)$$

$$s\widetilde{P}_1(s) = -\lambda\widetilde{P}_1(s) + 2\lambda c\widetilde{P}_2(s)$$

$$s\widetilde{P}_0(s) = \lambda\widetilde{P}_1(s) + 2\lambda(1-c)\widetilde{P}_2(s) + 3\lambda(1-c)\widetilde{P}_3(s)$$

On solving these equations the reliability of the system is computed as

$$R(t) = \sum_{i=1}^{3} P_i(t) = e^{-3\lambda t} + 3ce^{-2\lambda t}\left(1 - e^{-\lambda t}\right) + 3c^2 e^{-\lambda t}\left(1 - e^{-\lambda t}\right)^2 \quad \ldots. (1)$$

Also, the mean-time-to-failure (MTTF) is given as

$$M = \int_0^{\infty} R(t)dt = \frac{2 + 3c + 6c^2}{6\lambda} \quad \ldots.. (2)$$

III. Reliability Optimization Using Particle Swarm Optimization

The Particle Swarm Optimization algorithm is an evolutionary approach to optimization derived from the natural behaviour exhibited by swarms of bird species and fish. This algorithm uses a swarm of particles to explore the solution space cooperatively, thus determining the optimal solution for an optimization problem. Each particle is a representative of a solution, and the particle movements are controlled based on the position of the particle and its velocity. During iteration k, the position and velocity of the ith particle are denoted by $x_i(k)$ and $v_i(k)$**.**

As each individual particle explores the search space, it records the best position attained during the search, termed personal best ($pbest_i$**).** Additionally, each particle is influenced by the best position achieved by all particles in the search space, termed global best ($gbest_i$**).** As mentioned earlier, the movement of the particles is controlled through the velocity update formula, and hence the velocity controls the movement of the particle.

$$\mathrm{v_i(k+1)} = w * v_i(k) + c_1 * r_1 * \left(pbest_i(k) - x_i(k)\right) + c_2 * r_2 * \left(gbest_i(k) - x_i(k)\right)$$

$$x_i(k+1) = x_i(k) + v_i(k+1)$$

- $\mathrm{v_i(k+1)}$ = updated velocity of the ith particle at iteration (k+1)
- $v_i(k)$ = current velocity of the particle
- w = inertia weight controlling exploration capability
- c_1= cognitive acceleration coefficient
- c_2 = social acceleration coefficient
- r_1, r_2 = uniformly distributed random numbers in the interval (0,1)
- $pbest_i$= best previous position achieved by the particle
- $gbest_i$ = best position achieved by the entire swarm
- $x_i(k)$ = current position of the particle

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 55, Issue 03 : 2026

The velocity equation consists of three important components:

- **Inertia Component**
  $w * v_i(k)$
  This component ensures that the particle continues its current movement and contributes to the ability to explore the solution space. High inertia weight leads to a global search, while low inertia weight enhances local search around the optimum.

- **Cognitive Component**
  $c_1 * r_1 * \left(pbest_i(k) - x_i(k)\right)$
  This component is associated with self-learning. It guides the particle towards its best personal position found until now.
- **Social Component**
  $c_2 * r_2 * \left(gbest_i(k) - x_i(k)\right)$
  This component is associated with social learning. It steers the particle towards the best global position achieved by the swarm.

Together, all three components ensure a proper balance between exploitation, exploration, and convergence towards the reliability parameters that optimize the objective function value. For stable convergence, the commonly used parameter values are given in Table II.

Table II. Commonly used values of PSO parameters

Parameter	Typical Value
Inertia weight www	0.4 – 0.9
Cognitive coefficient c_1	2
Social coefficient c_2	2

The updated velocity is then used to compute the new particle position, enabling the swarm to iteratively search for optimal reliability solutions.
The reliability function used as the fitness function in PSO is:

$$R(t) = e^{-3\lambda t} + 3ce^{-2\lambda t}\left(1 - e^{-\lambda t}\right) + 3c^2 e^{-\lambda t}\left(1 - e^{-\lambda t}\right)^2$$

Hence, the optimization problem is:

$$\text{Maximize } F(\lambda, c) = R(t)$$

$$\text{Subject to: } 0 < c < 1 \text{ and } \lambda > 0$$

where:

- λ = component failure rate
- c = coverage factor
- $F(\lambda,c)$= objective (fitness) function
- t = mission time

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 55, Issue 03 : 2026

The optimization objective seeks minimum possible failure rate, maximum possible coverage factor and maximum system reliability. For every particle generated in PSO, the fitness value is evaluated using the reliability expression. If the i[th] particle is represented as:

$X_i = (\lambda_i, c_i)$ and its corresponding fitness function becomes $F_i = R(t, \lambda_i, c_i)$

The particle having the maximum fitness value is selected as the optimal solution during the iterative optimization process. The flow chart of particle swarm optimization algorithm is presented in Figure2.

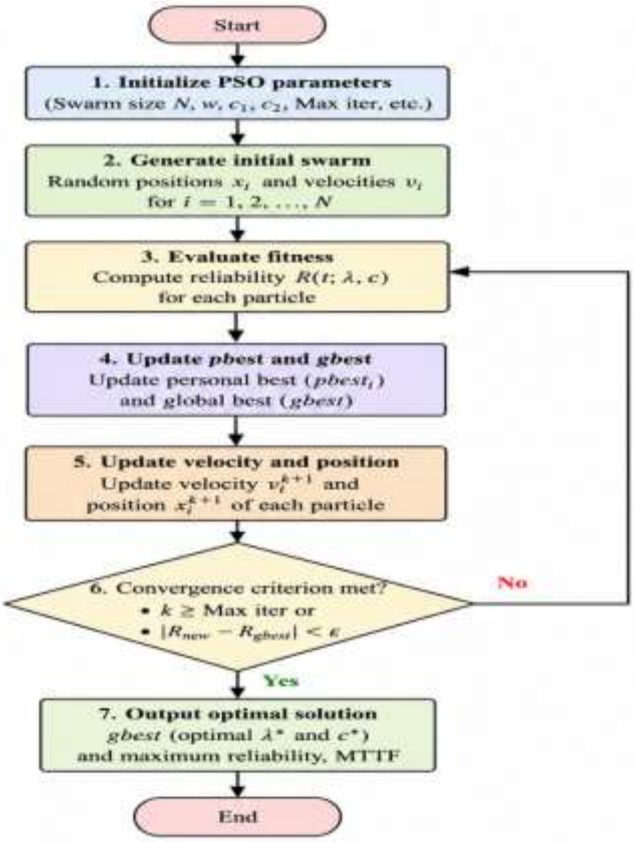

Figure 2: Flowchart of PSO algorithm for reliability optimization of the degradable computing system

IV. Numerical Illustration

In order to verify the efficiency of the reliability model proposed as well as the PSO algorithm, numerical analysis is performed based on a three-unit degradable computer system under imperfect coverage in a non-repairable environment. The parameter values considered are given in Table III.

4.1 Reliability Computation

In the derived reliability expression given in equation (1) for the aforesaid system, substituting the parameter values: λ = 0.002, c = 0.95, the reliability function becomes

$$R(t) = e^{-0.006t} + 2.85e^{-0.004t}(1 - e^{-0.002t}) + 2.7075e^{-0.002t}(1 - e^{-0.002t})^2$$

The reliability values computed for different mission times are tabulated in Table IV.

Table III. Reliability of System without Optimization

Parameter	Description	Value
(λ)	Failure rate of each unit	0.002 failures/hour
(c)	Coverage factor	0.95
(t)	Mission time	$(0 \leq t \leq 500)$ hours

4.2 Mean Time to System Failure (MTTF)

The value of MTTF, obtained by substituting λ = 0.002, c = 0.95 in the expression given in equation (2) for the Mean Time to System Failure is as follows:

$$MTTF = \frac{1}{0.006} + \frac{2.85}{0.012} + \frac{2.7075}{0.018} = 554.59\ hours$$

Table IV. Reliability of System without Optimization

Mission Time (t) (hours)	Reliability (R(t))
0	1.0000
50	0.9853
100	0.9676
150	0.9410
200	0.9096
250	0.8746
300	0.8368
350	0.7971
400	0.7560
450	0.7141
500	0.6717

From the above result, it is clear that the reliability of the system declines steadily with an increase in mission time owing to the failures of the components in the system. The system initially retains a high level of reliability, near unity, since all units are functioning properly.

However the presence of redundancy and coverage ensures that the system continues to work even in the case of a failure in one or two of the units. As a result, it is found that redundancy increases the reliability of the system, graceful degradation and coverage factor improves survivability.

4.3 Reliability Behaviour Analysis

Figure 3 shows the reliability of the degradable computing system with varying failure rates depending on the coverage factor values at t = 100 hours. It is evident from Figure 3 that the reliability decreases gradually when the failure rate is increased. This is anticipated since when the failure rate of components is high; there is an increased probability of breakdown, which leads to degradation and eventual system failure.

The impact of the coverage factor on the performance of the system is another notable feature that can be observed from Figure 3. As the value of the coverage factor varies from 0 to 1, the reliability of the system improves for all values of the failure rate. In the case where the coverage factor is low, the system becomes ineffective in identifying and handling faults; thus, its reliability is low.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 55, Issue 03 : 2026

However, with high coverage factors, the system is able to successfully detect faults and perform system reconfigurations and recoveries.
The reliability of the system at perfect coverage c = 1.0 is found to be relatively higher than those at other coverage factors; hence, the curve remains the highest at all the failure rates. Thus the inclusion of effective coverage mechanism improves system survivability and the overall reliability performance.

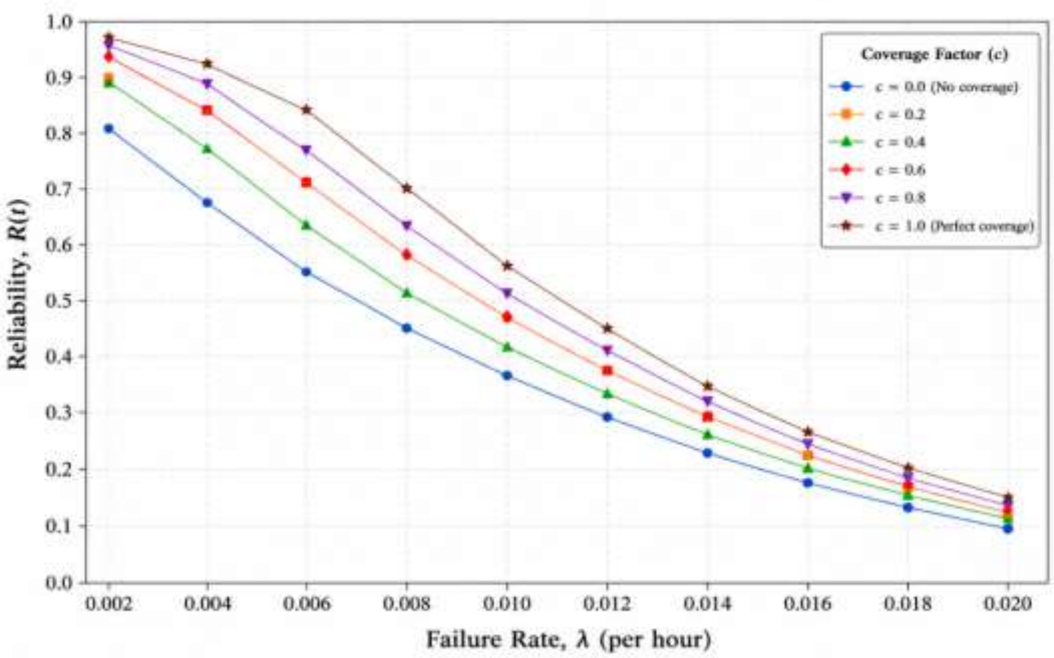

Figure 3: Variation of system reliability with failure rate for different values of coverage factor at t = 100 hours.

4.4 Reliability Optimization Using PSO

To optimize the reliability using particle swarm optimization the PSO parameters are selected are given in Table V. The convergence behaviour of PSO is shown in Table VI through iteration-wise improvement of reliability. The PSO method increases its ability to find reliable solutions by optimizing the failure rate and coverage factor. The results of the optimization procedure indicate quick convergence of the procedure to the optimum point.

Table V. Values of parameters selected for PSO

Parameter	Value
Swarm size	40
Iterations	100
Inertia weight	0.8
Cognitive factor	2
Social factor	2

The iterative optimization demonstrates the convergence property of the PSO algorithm. In the beginning, system reliability is equal to 0.9676 with the following parameters λ = 0.0020 and c =

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 55, Issue 03 : 2026

0.950. During the further optimization, the PSO method gradually reduces the failure rate and increases the coverage factor.

Table VI. Convergence behaviour of PSO

Iteration Number	Failure Rate (λ)	Coverage Factor (c)	Reliability (R(t))
1	0.0020	0.950	0.9676
20	0.0018	0.963	0.9751
40	0.0016	0.971	0.9828
60	0.0014	0.981	0.9899
80	0.0012	0.988	0.9941
100	0.0011	0.992	0.9964

At the 100th iteration, the optimized values of failure rate, coverage factor and hence the improved reliability are obtained as:

$$\lambda^* = 0.0011, \qquad c = 0.992 \; and \; R^*(t) = 0.9964$$

It is evident from the obtained results that PSO algorithm improves the reliability of the system by efficiently exploring the search space and converging towards an optimal solution. The increasing trend of reliability with increasing iterations signifies the effectiveness and stability of optimization in degradable systems.

Further convergence characteristics suggest that:

- The PSO algorithm efficiently explores and exploits the solution,
- Particles converge towards the global optimal solution,
- Optimization process is stable without any oscillations.

Thus, the PSO algorithm is extremely effective in enhancing system reliability.

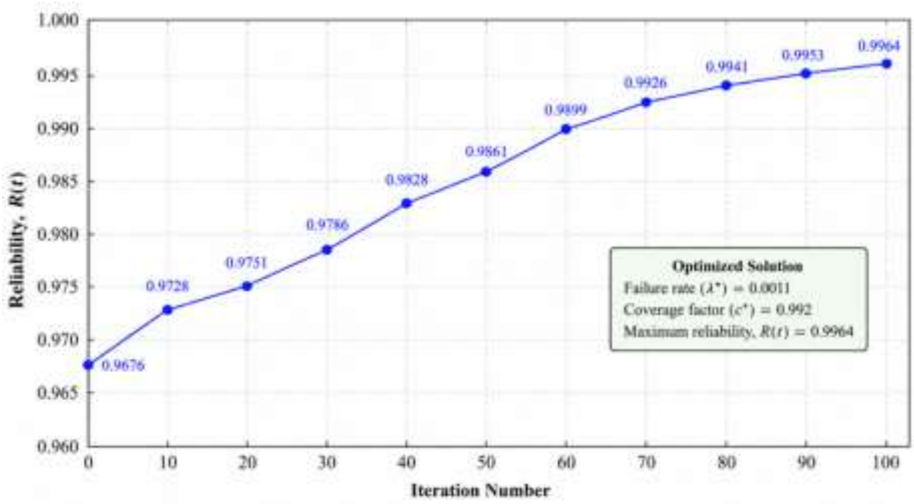

Figure 4: Convergence behaviour of Particle Swarm Optimization for maximizing system reliability.

Figure 4 depicts the convergence behaviour of the PSO algorithm shows the improvement in the reliability of the system after each iteration. Initially, the reliability will increase rapidly as different areas are explored by the particles. After that, convergence starts to happen more smoothly until it reaches the optimal point.

Table VII shows the reliability of the system without and with PSO for various mission times. From this comparison, it is clearly evident that PSO substantially improves system reliability and operational lifetime. The reliability deteriorates rather rapidly without optimization due to increase in the failure rate. But once the system is optimized, its reliability stays very high for a long time period.

Table VII. Reliability Comparison Before and After Optimization

Mission Time (t)	R(t) without PSO	R(t) with PSO
0	1.0000	1.0000
50	0.9853	0.9990
100	0.9676	0.9964
150	0.9410	0.9923
200	0.9096	0.9869
250	0.8746	0.9802
300	0.8368	0.9724
350	0.7971	0.9635
400	0.7560	0.9536
450	0.7141	0.9429
500	0.6717	0.9313

V. Conclusion

This paper provided a full reliability study of a 3-unit degradable computing system having coverage factor without repair ability. Continuous-Time Markov Chains were used to develop the probabilistic model. The Kolmogorov differential equation was solved in an analytical form to get accurate state probabilities. Explicit formulae for reliability and MTTF were established satisfactorily. Particle Swarm Optimization technique was used to maximize reliability through optimization of system parameters.

Numerical analysis showed that reliability reduces as the mission time increases, increases as the coverage factor rises. The PSO technique significantly improves the system reliability. The optimization process exhibited rapid and consistent convergence towards the global optimum based on the iterative refinement of the failure rate and coverage factor values. The results of convergence provided a proof of efficiency, stability, and fast search capacity of the PSO algorithm in the reliability optimization problem-solving process. Comparison of optimized and non-optimized systems demonstrated that application of PSO significantly increases the reliability and operational lifetime of the system. Therefore, the method proposed can be used effectively for assessing and optimizing the fault-tolerant degradable systems used in cloud computing, distributed systems, industrial automation, communications, etc.

References

[1] Trivedi, K.S. Probability and Statistics with Reliability, Queueing, and Computer Science Applications, 2nd ed.; Wiley, 2016.
[2] Dhillon, B.S. Reliability Engineering in Systems Design and Operation; CRC Press, Boca Raton, FL, USA, 2007.
[3] Shama, M. Reliability measures of a degradable system with standby switching failures and reboot delay. Int. J. Comput. Appl. 2013, 1(1), 1–16.

[4] Pandey, D.; Tyagi, S.K. Profust reliability of a gracefully degradable system. Fuzzy Sets Syst. 2007, 158(7), 794–803.
[5] Uprety, I.; Patrai, K. Estimating reliability of a repairable system with imperfect coverage and fuzzy parameters using parametric nonlinear programming approach. OPSEARCH 2016, 53(1), 1–15.
[6] Shooman, M.L. Reliability of Computer Systems and Networks: Fault Tolerance, Analysis, and Design; Wiley, 2002.
[7] Uprety, I.; Zaheeruddin, Z. Fuzzy reliability of gracefully degradable computing systems. In Proceedings of the International Conference on Methods and Models in Computer Science (ICM2CS'09); IEEE, 2009; pp. 1–4.
[8] Patrai, K.; Gupta, S. Reliability estimation of a degradable system using intuitionistic fuzzy Weibull lifetime distribution. Evergreen 2023, 10(4), 2331–2342.
[9] Wei, S.; Nourelfath, M.; Nahas, N. Condition-based maintenance optimization of degradable systems. Int. J. Math. Eng. Manag. Sci. 2022, 7(1), 1–15.
[10] Mellal, M.A.; Zio, E. An adaptive particle swarm optimization method for multi-objective system reliability optimization. Proc. Inst. Mech. Eng. Part O J. Risk Reliab. 2019, 233(6), 990–1001.
[11] Kennedy, J.; Eberhart, R. Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks; IEEE, 1995; Vol. 4, pp. 1942–1948.
[12] Zavala, A.E.M.; Diharce, E.R.V.; Aguirre, A.H. Particle evolutionary swarm for design reliability optimization. In Lecture Notes in Computer Science; Springer, 2005; Vol. 3410, pp. 856–869.
[13] Arya, R.; Choube, S.C.; Arya, L.D. System reliability enhancement using particle swarm optimization (PSO). J. Inst. Eng. India Electr. Eng. Div. 2008, 89(12), 44–47.
[14] Malhotra, R.; Negi, A. Reliability modeling using particle swarm optimization. Int. J. Syst. Assur. Eng. Manag. 2013, 4(3), 275–283.
[15] Garg, H.; Sharma, S.P. Multi-objective reliability-redundancy allocation problem using particle swarm optimization. Comput. Ind. Eng. 2013, 64(1), 247–255.
[16] Pnt, S.; Kumar, A.; Ram, M. Reliability optimization: A particle swarm approach. In Nature Inspired Optimization Techniques for Image Processing Applications; Springer, 2017; pp. 117–134.
[17] Abdullah, G.; Hassan, Z. Using particle swarm optimization (PSO) to address reliability allocation of complex networks. J. Phys. Conf. Ser. 2020, 1664(1), 012125.
[18] He, T.; Guo, J.; Xu, A.; Liu, J.; Zhang, Y.; Yang, B. A PSO-based sensitivity analysis method for reliability allocation in power systems. In 2025 15th International Conference on Power and Energy Systems (ICPES); IEEE, 2025; pp. 383–389.