Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen
 Published by Research Publishing, Singapore.
doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P8405-cd

Optimization of Maintenance Planning in Nuclear Power Plant Security Systems Using Cuckoo Optimization Algorithm

João Victor Alegrio

Department of Nuclear Engineering, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil.
E-mail: joaoalegrio@gmail.com

Andressa dos Santos Nicolau

Program of Nuclear Engineering, COPPE, Federal University of Rio de Janeiro, Rio de Janeiro, Brazil.
E-mail: andressa@lmp.ufrj.br

Optimizing performance is one of the main goals in science and technology. However, it is a significant challenge, as it is usually constrained by safety requirements and budget limitations. As a result, optimization algorithms utilizing artificial intelligence to enhance the performance of industrial systems have been developed over the last few decades. Due to advances in computing and the proven effectiveness of artificial intelligence methods, there has been a growing application of these methods in the nuclear sector. This has encouraged the publication of studies that cover everything from the design phase to maintenance policies and life extension strategies. A nuclear power plant is typically designed for a 40-year service life, during which the reliability of system components, including main and auxiliary safety systems, is ensured by high safety standards and management policies such as maintenance, inspection, and testing. As the end of this period approaches, plants can choose between decommissioning or extending their service life, often opting for life extension to operate for 40 to 60 years, based on feasibility studies that assess component performance and the need for replacements. Maintenance is crucial but can increase system downtime. Therefore, planning for redundant trains, which allow maintenance while the system continues operating, is fundamental. Backup systems must also maintain high levels of reliability to avoid failures when needed. This work developed an optimization tool using Probabilistic Safety Assessment (PSA) and the Cuckoo Optimization Algorithm (COA). Applied to a simplified model of the High-Pressure Injection System of a Pressurized Water Reactor (PWR), the tool aims to increase component reliability and reduce costs and downtime associated with maintenance. COA was chosen for its optimization efficiency, and the fitness function focused on minimizing unavailability and maintenance costs. The results of this work demonstrate an improvement of up to 60% compared to literature.

Keywords: Reliability, cuckoo, Artificial Intelligence.

1. Introduction

Optimizing performance is often one of the main goals in science and technology. However, this presents a significant challenge, as it is typically constrained by safety requirements and budget limitations. As a result, optimization algorithms utilizing artificial intelligence have been developed to enhance the performance of industrial systems over the past decades Castiglioni et al. (2021); Vaishya et al. (2020); Kathen et al. (2021). With advances in computational power and the proven effectiveness of artificial intelligence methods, there has been a growing application of these techniques in the nuclear sector. This trend has encouraged the publication of studies addressing topics ranging from the design phase to maintenance policies and lifetime extension Nicolau et al. (2023). A nuclear power plant is typically designed for a 40-year lifespan. During this period, the reliability of the system components, both primary and auxiliary, including safety systems, is ensured by high safety standards and the management policies (maintenance, inspection and testing) imposed on nuclear facilities CNEN (2002, 2005, 2011). When the operational period approaches its end, plants can choose between decommissioning CNEN (2012) or lifetime extension. Many opt for the latter, which can extend the operation from 40 to 60 years. Feasibility studies are essential to ensure that components can continue to operate and to identify which ones need replacement. Maintenance is crucial but can increase system downtime. Therefore, planning for redundant trains, which allow maintenance while keeping a system operational, is fundamental. Backup systems must also maintain high reliability levels to prevent failures when necessary. This paper developed an optimization tool using Probabilistic Safety Assessment (PSA) and the Cuckoo Optimization Algorithm (COA). Applied to a simplified model of the High-Pressure Injection System of a Pressurized Water Reactor (PWR), the tool aims to enhance component reliability and reduce costs and downtime associated with maintenance. The COA was chosen for its optimization efficiency, and the fitness function focused on minimizing unavailability and maintenance costs.

2. Metodology

Using a Python interface, it was possible to develop the cuckoo algorithm optimization, utilizing a cost versus reliability equation as the evaluation function.

2.1. *Cuckoo Optimization Algorithm*

The Cuckoo Optimization Algorithm (COA) is a swarm algorithm based on the behavior of the cuckoo bird, proposed by Yang and Deb (2009). This bird exhibits parasitic brood behavior. Unlike other birds, cuckoos do not build their nests; instead, they lay their eggs in the nests of other species, leaving the host bird to incubate their eggs.

The optimization algorithm based on cuckoo behavior (COA) begins with an initial population composed of mature cuckoos and cuckoo eggs. Mature cuckoos lay their eggs in the nests of other birds and if these eggs are not detected, they develop and become mature cuckoos. The survival of these eggs forms the basis of the algorithm, where unsuccessful eggs or cuckoos perish during competition, while successful ones migrate to better environments and continue reproducing. The process involves three main rules:

(i) Each host nest can contain only one cuckoo egg at a time, chosen randomly.
(ii) The nests with the best fitness results remain in the next generations.
(iii) A fraction of nests are discarded with a certain probability to explore new nests.

Each cuckoo egg represents a solution to the problem, and the goal is to generate better solutions by replacing low-performing ones. The algorithm aims to optimize the evaluation function, approaching the global minimum.

2.1.1. *Generation of the Search Space*

In optimization problems solved by genetic algorithms, solutions are represented by chromosomes, which are vectors of binary values. In the cuckoo algorithm, these vectors are called nests or habitats. Each nest represents a potential solution to the problem and has the same dimension as the search space. In an N-dimensional problem, a nest is a vector that represents a position in the N-dimensional space, as shown in Equation 1, i.e., a possible solution. Each component of this vector is a real number and this position defines the nest within the search space.

$$Nest = [x_1, x_2, x_3, x_4, ..., x_n] \quad (1)$$

The performance of a given nest is measured by evaluating the objective function f_o, which can be defined by Equation 2.

$$f_o(Nest) = fo(x_1, x_2, x_3, x_4, ..., x_n) \quad (2)$$

At the start of the algorithm, a number of N-dimensional nests are created to search for the solution to the problem. The evolution of the COA is represented by Equation 3, where Lévy denotes the Lévy Flight and is a scaling parameter. In this work, α = 1 was used.

$$Nest_i^{t+1} = Nest_i^t + \alpha \cdot \text{Lévy}(\lambda) \quad (3)$$

The COA code was developed using Python (Version 3.11) with the help of the NumPy library (Version 1.24) to perform some mathematical operations and manipulate matrices.

2.1.2. *Lévy Flight*

Lévy flight is a type of random movement where step distances follow the Lévy distribution. This pattern can be observed in the foraging behavior of albatrosses and other animals. In the COA, the Lévy distribution is used, as proposed by Yang and Deb (2009), to simulate the nest site selection process during optimization.

The Lévy distribution can be expressed, in a simplified form, by the equation $L(s) \sim u^{-\lambda}$, with $1 < \lambda < L \leq 3$ being maintained, or by Equations 4 and 5.

$$L(s, \gamma, \mu) = \sqrt{\frac{\gamma}{2\pi}} \exp\left(-\frac{\gamma}{2(s-\mu)}\right) \frac{1}{(s-\mu)^{\frac{3}{2}}} \tag{4}$$

$$L(s, \gamma, \mu) = 0 \tag{5}$$

With γ being a scaling factor for the given problem, and $\mu > 0$ representing the minimum step size, the step length can be obtained using Equation 6:

$$S = \frac{u}{|v|^{\frac{1}{\beta}}} \tag{6}$$

The Lévy flight involves choosing a random direction, generating a random number, and taking a step for the algorithm.In addition to the Lévy flight, the COA discards a percentage of nests based on the abandonment probability p_a and replaces these nests with new ones generated randomly. This discard process requires a higher processing capacity; however, it aids in the search for the global fitness, as it constantly creates new random nests, thus reducing the probability of the algorithm becoming trapped in a local maximum or minimum Kennedy and Eberhart (1995).

2.2. *Reliability Model*

This paper employs the methodology used by Lapa et al. (1999, 2000) to estimate the reliability of a component under a prevention and maintenance policy, building on a model by Lewis (1996). The approach extends the calculation of reliability across varying time intervals, not limited to periodic maintenance. The reliability of a component, $R(t)$, under a corrective maintenance policy is defined in terms of its maintenance schedule. The component's reliability at time t, given the last maintenance at $T_m(last)$, is expressed using an exponential function derived from the Weibull distribution Da Luz (2009), with parameters such as the component's characteristic life(θ), aging factor(m, and the probability of unsatisfactory maintenance(p), Equation7 is obtained, with problem parameters Nicolau et al. (2023):

$$\begin{aligned} R_m[t, T_m(i), T_m(last)] = \\ exp[-((t - T(last)/\theta_j)_j^m + (-[p(last)] + \\ \sum_{i=1}^{last}[-((T_m(i) - T_m(i-1))/\theta_j)_j^m]] \end{aligned} \tag{7}$$

While Equation 7 describes the reliability of a single component, the focus of this work is to estimate the availability of a system with multiple components. This requires considering the combination of all components in the system, as shown in Equation 8, where the mixed configuration of components directly influences system performance and reliability. Where U_s represents the unavailability of the system as a whole, g is the number of parallel paths, and p is the number of components in series in each path.

$$U_s = \prod_{g=1}^{n} \prod_{p=1}^{n} U_{rp} \tag{8}$$

2.3. *Cost Model*

The cost involved with testing and repairs plays a key role in the choice of maintenance for the components of a system, and is considered in the calculations presented in this work. The equation representing the cost model in this paper was developed by Da Luz (2009) and is expressed in Equation9. In this equation, Q denotes the component index, and j denotes the maintenance index. C_m and C_r refer to the costs associated with preventive maintenance and repairs, respectively while T_{mis} represents the total mission time.

$$
\begin{aligned}
C_{T_{sis}}^{0\to T_{mis}} = \sum_{Q=1}^{X} \Bigg\{ \sum_{j=1}^{last} C_{m_Q}^{(j-1)\to j} \frac{R_Q\big(T_{m_Q}(j)\big)}{R_Q\big(T_{m_Q}(j-1)\big)} \\
+ C_{r_Q}^{(j-1)\to j} \left(1 - \frac{R_Q\big(T_{m_Q}(j)\big)}{R_Q\big(T_{m_Q}(j-1)\big)}\right) \\
+ C_{r_Q}^{(last)\to T_{mis}} \frac{R_Q\big(T_{mis}\big)}{R_Q\big(T_{m_Q}(last)\big)} \Bigg\}.
\end{aligned} \tag{9}
$$

This equation results in the sum of the total cost (C_T) of the planned maintenance for the X components of the system under analysis.

2.4. *Fitness Function*

The fitness function, which evaluates a nest in the COA (as proposed by Da Luz (2009)), is a weighted sum that considers the average unavailability of the system throughout the component's mission, as well as the total costs related to maintenance activities. Equation 11 shows the impact of a certain maintenance policy on the reliability of the system.

$$
Fun = T_{mis}^{-1} \int_0^{T_{mis}} R_{sis}\, dt \tag{10}
$$

The fitness function then becomes a combination of Equation11 and the total costs related to the maintenance policy (presented in Equation 9).

$$
Fit = W_d(1 - Fun) + W_c\, C_{T_{sis}}^{0\to T_{mis}} \tag{11}
$$

In this expression, W_d and W_c correspond to the weights assigned to unavailability and the total maintenance costs of the system components, respectively. The value of W_d ranges from 0 to 1, while W_c ranges from 0 to 1, divided by the product of the number of components and the maximum number of interventions (maintenances).

2.5. *Problem Modeling*

In this study, a simplified model of the High Pressure Safety Injection System (HPIS) was chosen. It is a standby system, consisting of three pumps and four valves, forming an interconnected set of components. Figure 1 shows the diagram of the HPIS adopted in this work.

The operational characteristics used, such as test time and repair time, are fictitious values; however, they are based on typical values found in the literature for similar components Harunuzzaman and Aldemir (1996).

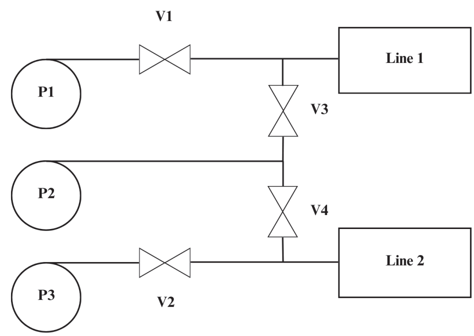

Fig. 1. Simplied diagram of the HPIS system

The problem was modeled based on the study by Luz, A. F., 2009 [14]. Initially, a limit of 23 interventions, or maintenances, was defined for each component of the system. Thus, considering the 7 components mentioned, up to 161 maintenances (7x23) could be scheduled, which represents the maximum number of elements that the candidate solution vector – or the nest in the COA – could contain.

Thus, the search space of the COA will have 161 dimensions. The position vector (nest) of the COA is composed of $X = x_0, x_1, x_2, ..., x_160$, totaling 161 elements, with each component having 23 elements. A 540-day operation period was defined for the system, with a time interval of 1 day. Therefore, the candidate solution to the problem considers the following conditions:

- The elements of the solution vector are real numbers ranging from -540 to 540 (mission time);
- Each real value is rounded to the nearest integer;
- Values between 1 and 540 represent valid days for a stop;
- Values below 1 are not considered as stops, resulting in fewer than 23 interventions per component;
- Identical stop values are not accepted for the

same component.

3. Tests and Results

3.1. *Tests*

For this study, typical values from the literature were adopted for the maintenance and repair costs of the HPIS components Harunuzzaman and Aldemir (1996). The following parameters were considered:

- Mission time of 540 days, with the average interval between reloads.
- Time required for performing a maintenance (or outage time):
  - Valve: $T_M V = 8h$
  - Pump: $T_M P = 24h$
- Weibull parameters:
  - $m_v(valves) = 1,35$
  - $m_p(pumps) = 1,8$ (Aging factor)
  - $\theta_v(valves) = 600days$
  - $\theta_p(pumps) = 600days$ (characteristic life)

It is assumed that, since the time required to repair a pump is greater than the time allocated for valve maintenance, when a valve is stopped for any type of maintenance (preventive or corrective), only tests (preventive maintenance) will be performed on the pump. However, when the pump is stopped, both types of maintenance can be performed on both components.

Five tests were conducted for each population (20, 50, and 100 particles), across 500, 1000, and 2000 generations. The best results from each generation were compared with those reported in the literature, which addressed the same problem using the Particle Swarm Optimization algorithm. These results are presented in Table 1.

Table 1. Comparison of results between COA and PSO algorithms.

Population	Generations	COA	PSO
20	500	0.002733	0.004917
20	1000	0.002666	0.002967
20	2000	0.002421	0.004938
50	500	0.002420	0.003210
50	1000	0.002441	0.002705
50	2000	0.002176	0.001730
100	500	0.002424	0.002423
100	1000	0.002196	0.002399
100	2000	0.002205	0.006679

Source: Own elaboration.

3.2. *Results*

Analyzing the result of the best fitness (50 nests with 2000 generations).

Table 2. Maintenance schedule of components.

Components	Stops	Days of stops
Valve 1	2	9-15
Valve 2	6	72-158-251-302-333-404
Valve 3	8	19-56-161-193-252-373-400-485
Valve 4	10	8-13-18-36-40-58-210-255-270-457
Pump 1	3	194-199-499
Pump 2	4	129-203-265-387
Pump 3	5	50-68-242-256-426

Source: Own elaboration.

It can be observed that the algorithm did not schedule maintenance on the same day, thereby avoiding a low system reliability during maintenance periods.

4. Conclusion

This paper presented the use of the COA optimization algorithm for maintenance planning of the HPSI system components in a nuclear power plant. The main objective was to evaluate the reliability of these components and the costs associated with performing maintenance when required. The parameters used are consistent with those reported in the literature Da Luz (2009); Harunuzzaman and Aldemir (1996). When compared to the results reported in the literature Longen (2022), the COA showed an average improvement of 21.05%. Furthermore, it achieved performance gains of up to 66.98% compared to the PSO algorithm, highlighting its effectiveness in solving the proposed problem.

In terms of computational efficiency, for the most computationally intensive case(2000 iterations and a population size of 100), the COA algorithm completed execution in 77.9 seconds, whereas the PSO algorithm took 98.8 seconds. This represents a 21.15% reduction in execution time, reinforcing the COA's advantage in optimizing both performance and computational cost.

References

Castiglioni, I., L. Rundo, M. Codari, et al. (2021). Ai applications to medical images: From machine learning to deep learning. *Physica Medica 83*, 92–94.

CNEN (2002). *Licenciamento de Instalações Nucleares*. Norma CNEN-NE-1.04.

CNEN (2005). *Diretrizes Básicas de Proteção Radiológica*. CNEN-NE-3.01.

CNEN (2011). *Gerenciamento de Rejeitos Radioativos em Usinas Nucleoelétricas*. Posição Regulatória 1.26 / 001.

CNEN (2012). *Descomissionamento de Usinas Nucleoelétricas*. Norma CNEN-NN-9.01.

Da Luz, A. F. (2009). Uma metodologia baseada no algoritmo de otimização por enxame de partículas para manutenção preventiva focada em confiabilidade e custo. Master's thesis, CNEN/IEN.

Harunuzzaman, M. and T. Aldemir (1996). Optimization of standby safety system maintenance scheduling in nuclear power plants. *Nuclear Technology 13*, 199–205.

Kathen, M. J. T., I. J. Flores, and D. G. Reina (2021). An informative path planner for a swarm of asvs based on an enhanced pso with gaussian surrogate model components intended for water monitoring applications. *Electronics 10*(13).

Kennedy, J. and R. Eberhart (1995). Particle swarm optimization. In *Proceedings of ICNN'95 - International Conference on Neural Networks*, Volume 4, Perth, WA, Australia, pp. 1942–1948.

Lapa, C. M. F., P. C. M. N. A. Pereira, and A. C. A. Mol (1999). Aplicação de algoritmos genéticos na otimização da política de manutenções preventivas de um sistema nuclear centrada em confiabilidade. In *Proceedings of the 15th Brazilian Congress of Mechanical*, Volume 62.

Lapa, C. M. F., P. C. M. N. A. Pereira, and A. C. A. Mol (2000). Maximization of a nuclear system availability through maintenance scheduling optimization using genetic algorithm. *Nuclear Engineering & Design 196*, 95–107.

Lewis, E. E. (1996). *Introduction to Reliability Engineering*. John Wiley & Sons.

Longen, M. (2022). Planejamento de manutenções e testes otimizado pelo algoritmo pso para componentes de um modelo simplificado do sistema de injeção de alta pressão de um reator pwr.

Nicolau, A. S. et al. (2023). Deep neural networks for estimation of temperature values for thermal ageing evaluation of nuclear power plant equipment. *Progress in Nuclear Energy*.

Vaishya, R., M. Javaid, I. H. Khan, et al. (2020). Artificial intelligence (ai) applications for covid-19 pandemic. *Diabetes & Metabolic Syndrome: Clinical Research & Reviews 14*(4), 337–339.

Yang, X. and S. Deb (2009). Cuckoo search via lévy flights. In *Proceedings of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009)*, pp. 210–214.