

OPTIMIZATION OF PREVENTIVE MAINTENANCE BY A COMPARATIVE APPROACH BASED ON EXACT RESOLUTION METHODS AND GENETIC ALGORITHMS: APPLICATION TO A PRODUCTION UNIT

Ngnassi Djami A.B ^{1*}; Samon J.B²; Nzié W^{1,2}

¹Department of Fundamental Sciences and Techniques of Engineer, Chemical Engineering and Mineral Industries School, University of Ngaoundere, Ngaoundere, Cameroon

²Department of Mechanical Engineering, National School of Agro-Industrial Sciences, University of Ngaoundere, Ngaoundere, Cameroon

ngnassbris@yahoo.fr^{1*}, boscosamon@yahoo.fr², wnzie@yahoo.fr fr^{1,2}

Correspondence email : ngnassbris@yahoo.fr

Abstract

The control of the maintenance of the industrial installations, in particular of the costs due to the implementation of the preventive policies is very interesting because of the growing importance of this service in the chains of production. The objective of this paper is to minimize the preventive maintenance costs of a production unit. For this, a state of the art on the maintenance cost models according to the policy used is first made, then a synthesis of the optimization methods is made in order to deploy the exact resolution methods and the genetic algorithms. The result of this paper is the proposal of a cost model corresponding to a periodic maintenance policy with minimal repair to the failure and the optimization of the periodicities of the partial revisions of the production unit.

Keywords: Preventive maintenance, Reliability, Optimization, Cost, Genetic algorithm

1. Introduction

Today, maintenance occupies a very important place in the production chain because the failure of a system during production can have direct and indirect consequences that are extremely detrimental for the system and for other business functions. The failure of a machine can generate: delays in delivery, loss of customers, larger stocks of finished products, cash flow difficulties, etc.

Sudden breakdowns are sometimes very costly and the loss of production during corrective interventions causes a loss of profit which can affect the profits of the company. Add to this safety issues, diminished production quality and possible loss of reputation for the company, it becomes clear that such failures should not be tolerated where preventive maintenance is required.

Optimizing preventive maintenance is a process of improving their performance and efficiency (of the company). This process tries to balance the requirements of preventive maintenance (legislative, economic, technical, etc.) and the resources used to carry out their program (labour, spare parts, consumables, equipment, etc.). The goal of preventive maintenance optimization is to choose the appropriate policy for each piece of equipment and the identification of the periodicity of this policy should be carried out to achieve the objectives concerning the safety, the reliability of the equipment

and the availability of the system. When preventive maintenance optimization is effectively implemented, overall preventive maintenance costs will be reduced.

2. State of the art on maintenance cost models according to the policy used

Production and service equipment constitute an important part of the capital of the majority of industries. This equipment is generally subject to degradation with use and time. For some of these systems, such as aircraft, nuclear systems, oil and chemical facilities, it is extremely important to do everything possible to avoid failure in operation because it can be dangerous. Moreover, for continuously operating units such as oil refineries, the loss of earnings is high in the event of a stoppage. Therefore, maintenance becomes a necessity to improve reliability. The growing importance of maintenance has generated an ever-increasing interest in the development and implementation of maintenance strategies for improving system reliability, preventing failures and reducing maintenance costs.

2.1. Notions on maintenance

2.1.1. Standard definitions

According to standard NF X 60-10 (December 1994), maintenance is "all activities intended to maintain or restore an item in a state or under given operating safety conditions, to accomplish a required function. These activities are a combination of technical, administrative and managerial activities".

Corrective maintenance is the set of actions carried out after detection of the failure and intended to return an item to a state in which it can perform a required function (NF EN 2001). Preventive maintenance is the set of actions carried out at predetermined time intervals or according to prescribed criteria and intended to reduce the probability of failure or the degradation of the functioning of an asset (NF EN 2001).

2.1.2. Effects of maintenance on systems

Maintenance can be characterized by its effect on the state of the system after receiving a maintenance action, as follows [1, 2]:

- Perfect repair (maintenance): Any maintenance action that brings the system back to an "as good as new" state. After perfect maintenance, the system has the same failure rate as a new system. A replacement is considered perfect maintenance. Example: complete overhaul of an engine.
- Minimal repair (maintenance): Any action that brings the failure rate of the system back to what it was just before the "As bad as old" failure. Example : changing a car tyre.
- Imperfect repair (maintenance): Any action that restores the system to a state between "as good as new" and "as bad as old". It is considered as a general case encompassing the two extreme cases, perfect repair (maintenance) and minimal repair (maintenance). Example: development of an engine.

In Summaries and classifications of possible causes for imperfect maintenance are systematically given [3, 4, 5, 6, 7].

2.2. Concepts on reliability

The evolution The AFNOR X606500 standard defines reliability as "the ability of an entity to perform a required function, under given conditions, during a given time interval".

It is defined by: $R(t) = P (E \text{ not failing during the duration } [0, t] \text{ assuming that it is not failing at the moment } t = 0)$.

2.2.1. Weibull model

This mathematical model covers quite a large number of lifetime distributions. It was first used in the study of material fatigue, it has been useful in the study of failure distributions of vacuum tubes, and is now in almost universal use in reliability.

Its distribution function is given by the expression 1.

$$F(t) = \begin{cases} 1 - e^{-\left(\frac{t-\gamma}{\eta}\right)^\beta} & \text{si } t > \gamma \\ 0 & \text{si } t \leq \gamma \end{cases} \quad (1)$$

β , η and γ represent respectively the shape parameter, the scale parameter and the position parameter.

Its reliability function $R(t)$ is given by relation 2.

$$R(t) = e^{-\left(\frac{t-\gamma}{\eta}\right)^\beta} \quad (2)$$

Its density function $g(t) = \lambda(t).R(t)$ is given by relation 3 (where $\lambda(t)$ is the failure rate).

$$g(t) = \begin{cases} \frac{\beta}{\eta} \left(\frac{t-\gamma}{\eta}\right)^{\beta-1} e^{-\left(\frac{t-\gamma}{\eta}\right)^\beta} & \text{si } t > \gamma \\ 0 & \text{si } t \leq \gamma \end{cases} \quad \text{with } \lambda(t) = \frac{\beta}{\eta} \left(\frac{t-\gamma}{\eta}\right)^{\beta-1} \quad (3)$$

- Whether $\gamma = 0$ and $\beta = 1$, $g(t) = \frac{1}{\eta} e^{-\frac{t}{\eta}}$

This is the exponential distribution, the special case of the Weibull distribution.

- If $\beta \geq 3$, the Weibull distribution approaches the normal distribution from which it can practically not be distinguished from $\beta = 4$.

2.2.2. Proactive maintenance approaches

The idea is to compare the real distribution function with the theoretical one. We measure the difference, point by point between these two functions (relation 4).

$$D_{n_i} = |f(t_i) - F(t_i)| \quad (4)$$

The maximum difference thus obtained is given by relation 5.

$$D_{n,\max} = \max |f(t_i) - F(t_i)| \quad (5)$$

$F(t_i)$ and $f(t_i)$ denote respectively the theoretical distribution function and the real distribution function.

The theoretical distribution function is given by relation 6.

$$F(t_i) = 1 - R(t_i) = 1 - e^{-\left(\frac{t_i-\gamma}{\eta}\right)^\beta} \quad (6)$$

The real distribution function is calculated using the empirical relationships according to the size (N) of the data [8,9] (relationships 7 to 9).

- Method of median ranks ($N \leq 20$):

$$f(t_i) = \frac{\sum n_i - 0,3}{N + 0,4} \quad (7)$$

- Mean rank formula ($20 < N < 50$):

$$f(t_i) = \frac{\sum n_i}{N + 1} \quad (8)$$

- Grouping by classes with $k = \sqrt{N}$ ($N \geq 50$):

$$f(t_i) = \frac{\sum n_i}{N} \quad (9)$$

With the Kolmogorov-Smirnov table, the value of the difference $D_{n,\alpha}$ is determined at a fixed

level of significance α . There by:

- If $D_{n,\max} > D_{n,\alpha}$, then the hypothesis of the theoretical model is refused.
- If $D_{n,\max} < D_{n,\alpha}$, then the hypothesis of the theoretical model is accepted.

2.3. Maintenance policies for elementary systems

An elementary system is defined as any part that is part of a machine (screw, seal, shaft, pinion, pin, etc.) or a machine that is part of a set, such as a grinder in an infant flour production line, a turn in a mechanical production line. In this case, the reliability characteristics and any other variable of the model relate to the entire system, itself can be broken down into elementary entities.

2.3.1. Age-dependent preventive maintenance policy

According to this policy, an elementary component is replaced when it reaches the age T or failure depending on which event occurs first [7]. The average cost per unit of time is given by relation 10.

$$C(T) = \frac{C_p \cdot R(T) + [1 - R(T)]}{\int_0^T R(t) dt} \quad (10)$$

- T : Age of preventive replacement (decision variable);
- C_p : Cost of preventive replacement;
- C_c : Failure cost including replacement cost;
- $R(t)$: Reliability function;
- $C_p \cdot R(T) + [1 - R(T)]$: Total cost of the cycle;
- $\int_0^T R(t) dt$: Cycle length expectation.

Since then, several extensions or variants of this model have emerged [10,11,12,13,14,15].

2.3.2. Periodic preventive maintenance policy

In this policy, an item is preventively maintained at fixed time intervals kT ($k = 1, 2, \dots$) independent of failure history, and repaired upon failure. Another basic periodic preventive maintenance policy is "periodic replacement with minimum repair to failure" where an item is replaced at predetermined times kT ($k = 1, 2, \dots$) and failures are eliminated by minimum repairs [16]. In this class, we can also cite the block replacement policy where an element is replaced at pre-arranged times kT and on failure (generally used for multi-component systems). For this last policy, the characterized random process is a renewal process, the average cost per unit of time is given by relation 11.

$$C(T) = \frac{C_c \cdot H(T) + C_p}{T} \quad (11)$$

- $H(T)$: Average number of replacements from 0 to T ;
- C_p : Cost of the part;
- C_c : Cost caused by the failure.

The difficulty with the previous expression lies in the determination of the renewal function $H(T)$.

With the concepts of minimal repair and especially imperfect maintenance, different extensions and variants of these two policies have been proposed [17,18,19,20,21].

2.3.3. Periodic replacement policy with minimum repair

This policy is a variant of the previous one, the difference is that following a failure, the element receives a minimal repair. Therefore, failures occur following an inhomogeneous Poisson process. The average number of failures in an interval $[0 ; T]$ is given by relation 12.

$$H(T) = \int_0^T \lambda(t) dt \quad (12)$$

$\lambda(t)$ represents the failure occurrence rate. For a non-repairable component, it represents the failure rate. Relationship 11 then becomes relationship 13.

$$C(T) = \frac{C_c.H(T) + C_p}{T} = \frac{C_c \int_0^T \lambda(t) dt + C_p}{T} \quad (13)$$

2.3.4. Imperfect Periodic Maintenance Policy with Minimal Repair

Under this policy, the item is not replaced periodically but just receives imperfect maintenance. As an example, we can cite an industrial machine that periodically receives partial overhauls and after a certain number of partial overhauls, the machine receives a general overhaul. This will mean that the rate of occurrence of failures will change after each preventive maintenance action, because we recall that, imperfect maintenance makes it possible to reduce the failure rate to a level between the initial failure rate (nine) and the one just before the maintenance. In this case, the effect of each maintenance on the system must be measured. The system failure rate after each maintenance will be expressed as a function of this effect and the previous failure rate. We give Gertsbakh 's model [22] where he assumes that the effect of all preventive maintenance is constant. It varies the failure rate exponentially, by an amount equal to e^α ($\alpha > 0$). The average cost per unit time is given by relation 14.

$$C(T) = \frac{C_c.H(T)(1 + e^\alpha + \dots + e^{\alpha(K-1)}) + (K - 1) .C_p + C_{ov}}{KT} \quad (14)$$

- C_c : Minimum repair cost;
- C_p : Cost of imperfect preventive maintenance (partial overhaul);
- C_{ov} : Cost of the general overhaul;
- K : Number of partial revisions before the general revision;
- e^α : Degradation factor.

There are other maintenance policies for single-component systems whose synthesis is presented [23,24,25,26,27,28,29,30].

In view of this synthesis of the maintenance cost models according to the policy used, it appears that the imperfect periodic maintenance policy with minimal repair is the most appropriate for the maintenance department. It is therefore this model that will be developed in the Application part of this paper.

III. Optimization methods

There are many optimization methods. However, they can be classified into two main categories: exact resolution methods and stochastic methods. In the first category, we find all the methods that seek the minimum of a function based on the knowledge of a search direction, often given by the gradient of this function. In the case of multiple optima, they stop on the first encountered. Stochastic methods are an alternative to overcome this drawback. The three most popular stochastic methods are genetic algorithms, simulated annealing, and tabu search. They are able to find the global minimum of a function even in very difficult cases, but the computation time can be high.

3.1. Exact resolution methods

A few methods of the class of complete or exact algorithms are presented. These methods give a guarantee to find the optimal solution for an instance of finite size in a limited time and to prove its optimality [31]. We will give the desired idea of each method and describe in more detail the method of the golden ratio, which will be part of the methods deployed in this work.

3.1.1. Separation and evaluation method (Branch and Bound)

The separation and evaluation algorithm, better known by its English name Branch and Bound (B&B), is based on a tree method of finding an optimal solution by separations and evaluations, by representing the solution states by a state tree, with knots, and leaves [32].

3.1.2. Plane cutting method (Cutting-Plane)

The plane cut method was developed by [33]. It is intended to solve combinatorial optimization problems which are formulated in the form of a linear program.

3.1.3. Mathematical methods

To determine an optimum, the mathematical methods are based on the knowledge of a search direction often given by the gradient of the objective function with respect to the parameters.

3.1.4. Conjugate gradient method

The conjugate gradient method [34,35,36,37], is an improved variant of the steepest slope method, which involves following the opposite direction of the gradient. This method has the disadvantage of creating orthogonal search directions, which slows down the convergence of the algorithm. The method of Fletcher and Reeves [34] solves this problem by determining the new search direction from the gradient at the current and previous steps.

3.1.5. Quasi-Newton methods

Quasi-Newton methods consist in imitating Newton's method where the optimization of a function is obtained from successive minimizations of its second-order approximation. They do not calculate the Hessian but they use a positive definite approximation of the Hessian which can be obtained either by the expression proposed by Davidon -Fletcher-Powell, or by that proposed by Broyden -Fletcher- Goldfard -Shanno [38].

3.1.6 Golden ratio method

The golden ratio method (golden search method) is an optimization technique that seeks the extremum (minimum or maximum) of a function, in the case of a unimodal function, i.e. in which the global extremum sought is the single local extremum. If there are several local extrema, the algorithm gives a local extremum, without it being guaranteed that it is the absolute extremum. The steps of the method are as follows:

- 1st step: take the two points $c = a + (1-r)h$ and $d = a + rh$ the interval $[a, b]$, with $r = (\sqrt{5}-1)/2$ and $h = b - a$.
- 2nd step: if the values of $f(x)$ for these two points are almost equal ($f(a) \approx f(b)$) and the width of the interval is small enough ($h \approx 0$), then stop the iteration to exit the loop and declare $x^0 = c$ or $x^0 = d$ depending on whether $f(c) < f(d)$ or not. If not, go to step 3.
- 3rd step: if $f(c) < f(d)$, take the new upper limit of the interval $b \leftarrow d$. If not, take the new lower limit of the interval $a \leftarrow c$. Then return to the 1st step.

We note the following points concerning the procedure of the method of the golden ratio:

- At each iteration, the new width of the interval is : $b - c = b - (a + (1-r)(b-a)) = rh$ or $d - a = a + rh - a = rh$, so that it becomes r times the width of the old interval ($b - a = h$).
- The golden r ratio is fixed such that a point $C_1 = b_1 - rh_1 = b - r^2h$ in the new interval $[c, b]$ conforms with $d = a + rh = b - (1-r)h$, that is $r^2 = 1 - r$, $r^2 + r - 1 = 0$, $r = (-1 + \sqrt{1+4})/2 = (\sqrt{5}-1)/2$.

3.2. Genetic Algorithms

3.2.1. Origin and principle

Genetic algorithms (GA) are stochastic optimization algorithms based on the mechanisms of natural selection and genetics [39,40]. The researcher Reichenberg [41], is the first scientist who introduced evolutionary algorithms by publishing his work "Evolution strategies." These algorithms are broadly inspired by Darwin's theory of evolution published in 1859. Next, Holland [42] proposed the first genetic algorithms to solve combinatorial optimization problems, and they were also developed by the work of David Goldberg published in 1989 [43].

The aim of the genetic algorithm is to bring up, from one generation to another, the candidates (potential solutions) most suited to solving the problem. Each generation is made up of a defined number of individuals, these form a population, and each of them represents a point in the search space. Each individual (chromosome) has information coded in the form of a chain of characters that analogically constitutes genes. Then the passage from one generation to another is carried out based on the process of evolution by the use of evolutionary operators like selection, crossing, and mutation.

Their operating principle is quite simple. From an initial population created at random, composed of a set of individuals (chromosomes), we proceed to the evaluation of their parent qualifications to highlight the best suited, as long as the least effective are rejected. Then, the most qualified individuals are chosen by privileged selection by giving them a chance to reproduce by crossing and mutating via the two operators of crossing and mutation. Then by relaunching this process several times, the optimal solution can be refined by passing from one generation to another.

3.2.2. Description of the formalism used

The convergence of genetic algorithms has been demonstrated for many problems, although optimality cannot be guaranteed. The ability of a genetic approach to find the right solution often depends on the adequacy of the coding, the evolution operators, and the measures of adaptation to the problem being addressed. The method proposed here is based on genetic algorithms [43] and evolutionary strategies [44]. It combines the principle of survival of the ablest individuals and genetic combinations for an elitist research mechanism. The genetic method produces new solutions (children) by combining existing solutions (parents) selected from the population, or by mutation. The central idea is that parent solutions will tend to produce superior child solutions in terms of adaptation so that ultimately a solution obtained is optimal.

In this study, we used a genetic method previously defined by [45] with a definition of the chromosome and the operators of selection, combination, and mutation concerned. Unlike genetic algorithms, the genetic method used is designed to minimize and not maximize. This method, like genetic algorithms, is not limited by assumptions about the objective function and research space, such as continuity or differentiability. It uses a population of points simultaneously by contrast with usual methods using only one point. Genetic operators are elitistically improving the search process to find the global optimum. There are more complicated genetic operators, but the basic operators and their various modifications can generally be applied. The choice of these operators depends on the nature of the problem and the performance requirements. The genetic algorithm that we are going to implement is as follows, where the process is applied to iteration k :

- a. Data coding;
- b. Generation of the initial population P_0 of N individuals;
- c. Assessment of the adaptation of all individuals in the population;

- d. Selection of a proportion of the best individuals (parents for the production of new individuals);
- e. The Crossing of all individuals in the population P_k two by two with a probability P_m , we will have N children noted C_k ;
- f. Mutation of all individuals in the population, we will have N elements noted M_k ;
- g. Choice of the most suitable individuals, i.e., those who optimize the objective function;
- h. If the stop test is verified, stop, otherwise return to step a.

We will choose, as a stop test in our implementation, a finite number of iterations.

It is important to note that the stopping criterion can be several cycles of the algorithm (number of generations), the average of the adaptations of individuals, a convergence factor, etc.

An individual represents a vector of decision variable (parameters), and its adaptation is measured by the objective function. The formalism and the genetic operators are detailed below.

3.2.2.1. Data coding

The first step is to properly define and code the problem. That step associates with each point of the search space a specific data structure called a chromosome, which will characterize each individual in the population. This step is considered to be the most important step in GA because the success of these algorithms depends heavily on how individuals are coded.

There are different choices for coding a chromosome, this choice being a very important factor in the progress of the algorithm so it must be well suited to the problem being addressed:

- **Binary coding:** It is the most used coding. The chromosome is coded by a string of bits (which can take the value 0 or 1) containing all the information necessary to describe a point in space;
- **Multi-character coding:** this is often more natural. We are talking about multiple characters as opposed to bits. A chromosome is then represented by a series of numbers or characters, each representing a gene;
- **Coding in the form of a tree:** this coding in tree structure starts from a root (comprising several parts equal to the number of initial individuals), from which one or more children can be derived. The tree then builds up gradually, adding branches to each new generation.

3.2.2.2. Generation of the initial population

Each chromosome is the potential result of the optimization problem. We define a chromosome as a chain composed of genes, which are the parameters (decision variables) to find. The value of a gene is called an allele. The possible value of an allele is an integer or a real value. Each gene is created randomly, using equation 15.

$$a_j = (a_j)_l + ((a_j)_u - (a_j)_l) \times \gamma_j \quad (15)$$

Where:

- $\gamma_j \in \{0;1\}$ is chosen randomly
- $(a_j)_l, (a_j)_u$ are the minimum and maximum limits of the allele a_j . They are chosen according to the problem to be treated.

Each chromosome, called an individual in a haploid representation, can be written:

$$X_i = [a_1, \dots, a_j, \dots, a_m]$$

With:

- m is the number of genes
- $i = 1, \dots, N$ and N is the size of the population (number of individuals).

All the constraints are taken into account in the initial phase of population creation. When an individual is created, if the constraints are respected, this individual is integrated into the initial

population; otherwise, it is not. At the start of the algorithm, the initial population contains individuals.

The length of the chromosome m and the size of the population N is two of the four adjustment parameters of the genetic method.

3.2.2.3. Objective function and adaptation

We evaluate the different solutions proposed to treat them according to their relevance and to see which the best is. For this, we use the objective function.

This function measures the performance of each individual. To be able to judge the quality of an individual and thus compare him to others.

3.2.2.4. Selection of the most suitable individuals

When the entire population is assessed at generation t , individuals are ranked in ascending order of objective function. Then the selection is made. Selection helps to statistically identify the best individuals in a population and eliminate the bad ones from one generation to the next. This operator also gives a chance to the bad elements because these elements can, by crossing or mutation, generate relevant descendants compared to the optimization criterion.

The first $N \times G$ individuals (the best $N \times G$) are selected to be parents. G is the third setting parameter of the genetic method. G is called the generation gap. G makes it possible to select a part of the population to provide sufficient genetic material without decreasing the speed of convergence [43]. There are different selection techniques:

- **Selection by rank:** This selection method always chooses the individuals with the best adaptation scores, without allowing chance to intervene;
- **Selection by wheel:** For each parent, the probability of being selected is proportional to their adaptation to the problem (their score by the fitness function). This selection mode can be imaged by a casino roulette wheel, on which all the chromosomes of the population are placed, the place is given to each of the chromosomes being proportional to its adaptation value. Also, the higher an individual's score, the more likely he is to be selected. We spin the wheel as many times as we want individual sons. The best will be able to be drawn several times, and the worst never;
- **Selection by tournament:** Two individuals are chosen at random, their adaptation functions are compared, and the best suited is selected;
- **Uniform selection:** We are not interested in the adaptation value of the objective function, and the selection is made in a random and uniform manner such that each individual has the same probability $P(i) = 1/N$ as all other individuals, where N is the total number of individuals in the population;
- **Elitism:** The passage from one generation to another through the crossing and mutation operators creates a great risk of losing the best chromosomes. Therefore, elitism aims to copy the best (or first - best) chromosome (s) from the current population to the new population before proceeding to the mechanisms of crossing and mutation. This technique quickly improves the solution because it prevents the loss of the most qualified chromosome when passing from one generation to another.

3.2.2.5. Crossing

The selected population is divided into $N/2$ couples formed randomly. Two parents P_1 , and P_2 are chosen randomly from the potential parents and their genes are combined according to equation 16.

$$a_j(k) = a_j(P_1) + (a_j(P_2) - a_j(P_1)) \times \gamma_j \quad (16)$$

Where:

- γ_j is a uniform random number,
- $k = N \times G + 1, \dots, N$, the k th individual,
- $j = 1, \dots, m$.

The newly created individual is then evaluated. If its adaptation is better than that of the worst parent, it is integrated into the population to training the next generation. If it is not the case, we repeat the combination.

3.2.2.6. Mutation of all individuals in the population

The mutation operator is a process where a minor change in the genetic code is applied to an individual to introduce diversity and thus avoid falling into local optima. This operator is applied with a probability P_m generally lower than that of the crossing P_c . This probability must be low. Otherwise, the GA will turn into a random search.

3.2.2.7. Choosing the best solutions

This choice consists in retaining the solutions which have a lower value of the objective function, and putting them in the population P_{k+1} .

3.2.2.8. Stopping criterion

The stopping criterion is evaluated in the current population. If it is filled, the whole population has converged on the solution. Otherwise, the reproduction pattern will be repeated. The stopping criterion used in this method expresses that all individuals have converged on the same solution and assumes that evolution is no longer possible, that is to say, that no better solution can be found.

The whole strategy is elitist because only the best individuals are selected for survival from one generation to the next and can be the parents of new and better individuals. To ensure convergence of the algorithm, the parameters N and G must be adjusted with care. The size of the population N affects both the performance and efficiency of the algorithm [45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60]. The algorithm is less efficient with very small population sizes. Large population size may contain more interesting solutions and discourage premature convergence towards sub-optimal solutions, but requires more assessments per generation, which can lead to a low convergence rate. The generation gap G determines the proportion of the population that remains unchanged between two generations.

It is chosen to select individuals as severely as possible, without destroying the diversity of the population too much. The global strategy used assumes that all the individuals who make up the population, from generation to generation, satisfy all the constraints.

The best solution for the latest generation represents the solution to the problem by the defined criteria.

IV. Application

The objective of this application is to minimize the maintenance cost of a production unit whose operating time history is given in table 1.

Table 1: Time Between Failures (TBF) History

Operation number	Date	TBF (hours)
1	09/30/2022	4350
2	12/01/2022	2720
3	05/01/2022	9830
4	09/06/2022	2110

5	07/08/2022	1410
6	11/08/2022	1940
7	10/09/2022	2880
8	10/13/2022	2590
9	06/11/2023	2140
10	26/11/2023	1270
11	12/15/2023	2930
12	03/03/2023	5670
13	04/06/2023	3350
14	09/28/2023	6440

It is assumed that the maintenance service follows an imperfect periodic maintenance policy with minimal repair, for which the cost model given by relation 14 is valid.

4.1. Determination of the parameters of Weibull

table 2 gives the calculation of the real distribution function ($f(t)$).

Table 2: Calculation of $f(t)$

<i>TBF(hours)</i>	n_i	$\sum n_i$	$f(t)$
1270	1	1	0.04861111
1410	1	2	0.11805556
1940	1	3	0.1875
2110	1	4	0.25694444
2140	1	5	0.32638889
2590	1	6	0.39583333
2720	1	7	0.46527778
2880	1	8	0.53472222
2930	1	9	0.60416667
3350	1	10	0.67361111
4350	1	11	0.74305556
5670	1	12	0.8125
6440	1	13	0.88194444
9830	1	14	0.95138889

By drawing the straight line which passes through the pair of points ($t/100, f(t)$) on the Weibull paper and its parallel which passes through the origin (assuming the simplifying hypothesis of the model according to which $\gamma = 0$), we obtain the following values: $\eta = 33 \times 100 \text{hours} = 3300h$ and $\beta = 3$ (see figure 1).

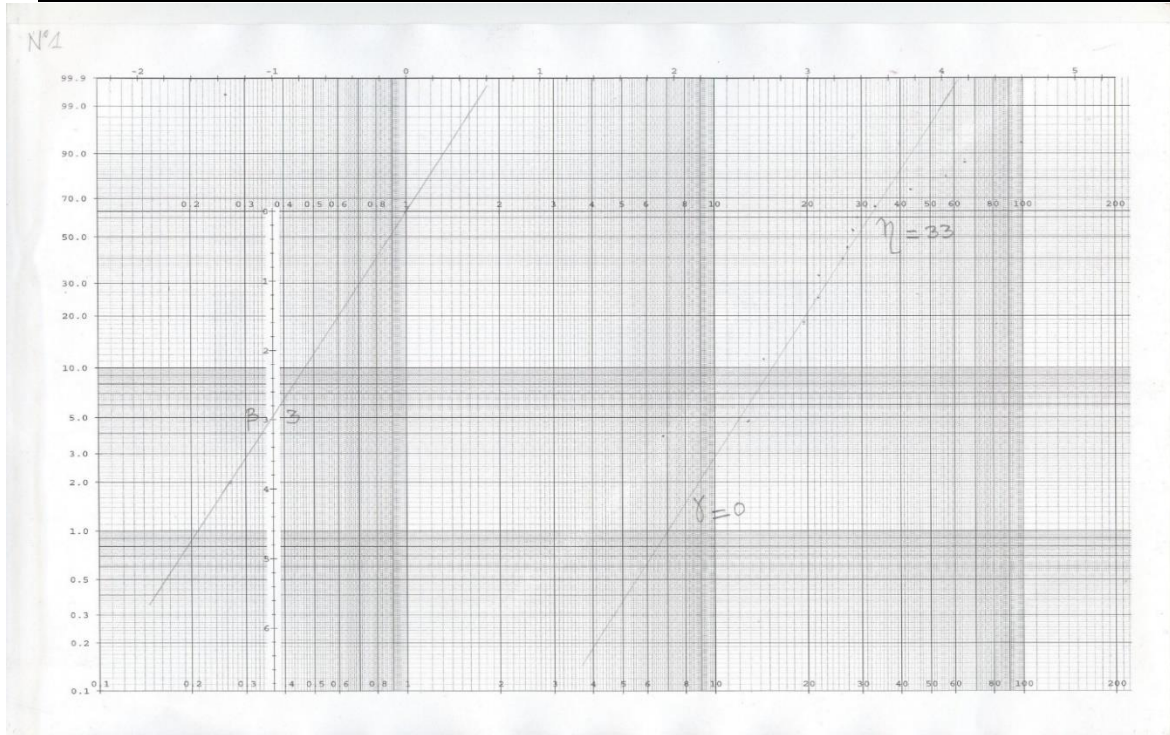


Figure 1: Estimation of Weibull parameters

4.2. Kolmogorov- Smirov validation test

table 3 gives the value of the deviation as a function of the values of $f(t)$ and of $F(t)$. The values of $F(t)$ being obtained by relation 6.

Table 3: Calculation of the gap

TBF(hours)	$F(t)$	$f(t)$	$D_{n,i}$
1270	0.13766232	0.04861111	0.08905121
1410	0.16686699	0.11805556	0.04881143
1940	0.29220549	0.1875	0.10470549
2110	0.33556924	0.25694444	0.0786248
2140	0.34330302	0.32638889	0.01691413
2590	0.45989253	0.39583333	0.0640592
2720	0.49306656	0.46527778	0.02778878
2880	0.53331059	0.53472222	0.00141163
2930	0.54539607	0.60416667	0.0587706
3350	0.64318313	0.67361111	0.03042798
4350	0.82405842	0.74305556	0.08100286
5670	0.94777262	0.8125	0.13527262
6440	0.97781660	0.88194444	0.09587216
9830	1	0.95138889	0.04861111

From table 3, $D_{n,max} = 0.13527262$.

For industrial equipment we take a risk of error $\alpha = 0.05$.

By exploiting the catalog giving the Level of significance of α , we find: $D_{n,\alpha} = 0.349$.

It can be seen that $D_{n,max} < D_{n,\alpha}$, consequently, the hypothesis according to which the times of failure follow a Weibull law is validated.

4.3. Development of the cost model

The following assumptions are made:

Under the following two assumptions:

- The element receives minimal repair following failure, so failures occur according to an inhomogeneous Poisson process.
- The system failure distribution follows a Weibull model with $\gamma = 0$.

According to relations 3 and 12 giving respectively the rate of failures and the average number of failures, we have relations 17 and 18.

$$\begin{aligned}
 H(T) &= \int_0^T \frac{\beta}{\eta} \left(\frac{t}{\eta}\right)^{\beta-1} dt = \frac{\beta}{\eta} \int_0^T \left(\frac{t}{\eta}\right)^{\beta-1} dt & (17) \\
 \Rightarrow H(T) &= \frac{\beta}{\eta} \int_0^T \left(\frac{t^{\beta-1}}{\eta^{\beta-1}}\right) dt = \frac{\beta}{\eta^\beta} \int_0^T t^{\beta-1} dt \\
 \Rightarrow H(T) &= \frac{\beta}{\eta^\beta} \left[\frac{t^\beta}{\beta} \right]_0^T = \frac{\beta}{\eta^\beta} * \left(\frac{t^\beta}{\beta} - 0 \right)
 \end{aligned}$$

From where:

$$H(T) = \frac{T^\beta}{\eta^\beta} \quad (18)$$

- By replacing 17 in 14, we obtain the relations 19 and 20.

$$\begin{aligned}
 C(T) &= \frac{C_c \cdot \frac{T^\beta}{\eta^\beta} (1 + e^\alpha + \dots + e^{\alpha(K-1)}) + (K-1) \cdot C_p + C_{ov}}{KT} & (19) \\
 \Rightarrow C(T) &= \frac{C_c \cdot T^\beta (1 + e^\alpha + \dots + e^{\alpha(K-1)})}{KT\eta^\beta} + \frac{(K-1) \cdot C_p + C_{ov}}{KT}
 \end{aligned}$$

From where:

$$C(T) = \frac{C_c \cdot T^{\beta-1} (1 + e^\alpha + \dots + e^{\alpha(K-1)})}{K\eta^\beta} + \frac{(K-1) \cdot C_p + C_{ov}}{KT} \quad (20)$$

The relation 20 represents our objective function to be minimized. To achieve this goal, two exact resolution methods will be used (the simple derivation technique and the golden ratio method), in order to compare their results with those of the genetic algorithm.

4.4. Optimization using simple derivation

The objective function $C(T)$ that we want to minimize is differentiable and in this case, it suffices to determine T_{opt} the solution of the equation $\frac{\partial C(T)}{\partial T} = 0$ which will lead us to the minimum cost (C_{min}) of $C(T)$ by taking into consideration the following conditions:

$$C_c > 0; C_p > 0; C_{ov} > 0; T > 0; K > 0; \beta > 1; \alpha > 0 \text{ and } \frac{\partial}{\partial T} \left(\frac{\partial C(T)}{\partial T} \right) \geq 0$$

By performing a simple derivation of the $C(T)$ previous expression, we obtain relations 21 and 22.

$$\frac{\partial C(T)}{\partial T} = \frac{C_c(\beta-1)T^{\beta-2} (1 + e^\alpha + \dots + e^{\alpha(K-1)})}{KT\eta^\beta} - \frac{(K-1) \cdot C_p + C_{ov}}{KT^2} \quad (21)$$

$$\Rightarrow \frac{\partial C(T)}{\partial T} = \frac{C_c(\beta-1)T^\beta (1 + e^\alpha + \dots + e^{\alpha(K-1)}) - \eta^\beta ((K-1) \cdot C_p + C_{ov})}{\eta^\beta T^2} \quad (22)$$

We deduce the expression for T_{opt} for that $\frac{\partial C(T)}{\partial T} = 0$ by relation 23.

According to relation 22, we have:

$$\begin{aligned} & \frac{Cc(\beta-1).T^\beta.(1 + e^\alpha + \dots + e^{\alpha(K-1)}) - \eta^\beta((K-1).Cp + Cov)}{\eta^\beta T^2} = 0 \\ \Rightarrow & Cc(\beta-1).T^\beta.(1 + e^\alpha + \dots + e^{\alpha(K-1)}) - \eta^\beta((K-1).Cp + Cov) = 0 \\ \Rightarrow & T_{opt} = \beta \sqrt{\frac{\eta^\beta((K-1).Cp + Cov)}{Cc(\beta-1)(1 + e^\alpha + \dots + e^{\alpha(K-1)})}} \end{aligned} \quad (23)$$

We have established a program on Matlab to calculate the value of T_{opt} in hours and days and the minimum cost (**Algorithm 1**).

The maintenance costs are proposed as well as the number of partial overhauls K and the maintenance efficiency factor α (table 4).

Table 4: Data for simulation on Matlab

Cc	Cp	Cov	K	α	β	η
170000	900000	8000000	8	0.9	3	3300

```

%*****
% Algorithm 1 Calculation code for simple derivation
%*****
%       Developed by: Dr NGNASSI DJAMI Aslain Brisco
%               Teacher-Researcher
%*****
%               Main program
%*****
clear all
close all
clc
%*****
%       Inserting data (Cc, Cov, Cp, K, beta, eta, alfa)
%*****
A0=1;
for j= 1: K -1
A0=A0+exp(j*alpha);
end
A1=( eta^beta)/(beta-1);
A2=(K- 1)* Cp+Cov ;
A3=Cc*A0;
T =[ 0:6000];
C=((T.^(beta-1) ) * Cc*A0)/(K*( eta^beta ))+(A2./(K*T));
TT=(A1*(A2/A3) )^ (1/beta)
T1=TT/24
Cmin = (((TT. ^beta-1) ) *Cc*A0)/(K*(eta^beta)))+(A2. /(K*TT))
% C(Toptimum)
plot (T, C);
%*****

```

By executing the proposed code, we obtain the results presented table 5.

Table 5: T_{opt} and C_{min} obtained by simple derivation

System	T_{opt} in hours	T_{opt} in days	C_{min} (USD)
Unit production	1181.3	49.2192	2269.8

figure 2 graphically shows the evolution of cost over time.

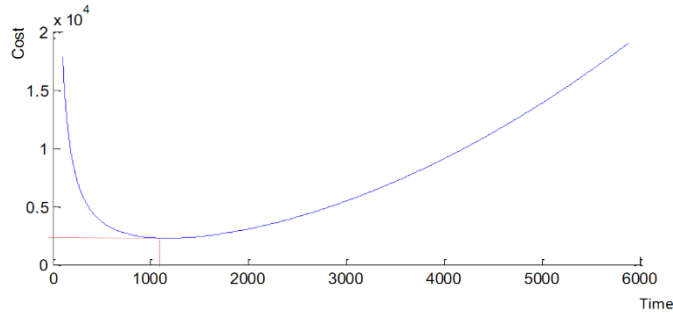


Figure 2: Cost evolution over time

The curve presented in figure 2 illustrates the result of the program. The cost begins to decrease until it reaches its minimum value of T_{opt} , then it increases over time.

4.5. Optimization using the golden ratio method

The program proposed in Matlab for the golden ratio method is given by **Algorithm 2**.

```

%*****
% Algorithm 2 Calculation code for Golden Search Method
%*****
%           Developed by: Dr NGNASSI DJAMI Aslain Brisco
%           Teacher-Researcher
%*****
%           Main program
%*****
clear all
close all
clc
%*****
% % %           Input
%*****
fx = ((x^(beta-1)) * Cc * A0) / (K * (eta^beta)) + (A2 / (K * x));
max = 100;
es = 10^-5;
r = (5^0.5 - 1) / 2;
%*****
% % %           Determine the Interval for the Initial Guess
%*****
x = [ 0:6000];
f = subs( fx,x );
a=100;
b=6000;
%*****
% % % %           Perform Golden Search
%*****
x1 = a;

```

```

xu = b;
iter = 1;
d = r*( xu -xl);
x1 = xl+d ;
x2 = xu- d;
f1 = subs( fx,x 1);
f2 = subs( fx,x 2);
if f1<f2
xopt = x1;
else
xopt = x2;
end
while ( 1)
d = r*d;
if f1<f2
x1 = x2;
x2 = x1;
x1 = xl+d ;
f2 = f1;
f1 = subs( fx,x 1);
else
xu = x1;
x1 = x2;
x2 = xu- d;
f1 = f2;
f2 = subs( fx,x 2);
end
iter = iter+1;
if f1<f2
xopt = x1;
else
xopt = x2;
end
if xopt ~=0
ea = (1 - r)* abs(( xu -xl)/ xopt )*100;
end
if ea <=es || iter >= maxit, break
end
end
Gold = xopt
%*****

```

By implementing the proposed program, we obtain the results of table 6.

Table 6: T_{opt} and C_{min} obtained by golden ratio method

System	T_{opt} in hours	T_{opt} in days	C_{min} (USD)
Unit production	1181.3	49.2192	2269.8

The results provided by the golden ratio method are identical to those of the simple derivation, which proves the effectiveness of this method in finding the extremum of a unimodal function.

4.6 Optimization using genetic algorithm

The program proposed in Matlab for the genetic algorithm is given by **Algorithm 3**.


```

%*****
%
%           Algorithm 3 Calculation code for Genetic Algorithm
%*****
%
%           Developed by: Dr NGNASSI DJAMI Aslain Brisco
%           Teacher-Researcher
%*****
%
%           Main program
%*****
clear all
close all
clcc
%*****
%
%           Parameter initialization
%*****
Np= 60;           % Population size
Pc = 0.5;        % Probability of crossing
Pm=0.01;        % Probability of mutation
Kmax = 100;      % Maximum number of iterations
lb = [100 ];    % lower bound
ub = [6000 ];   % upper bound
x0 = [1010 ];   % Approximate value
Fitness_F=@(x)((x(1) .^ (beta-1))*cc*A0)/(k*(nu^beta))+(A2./(k*x(1)) );
options= gaoptimset ('populationsize' ,60, 'generations' ,100, 'MutationFcn' ,{@
mutationadaptfeasible ,0.01}, 'crossoverfraction' ,0.5, 'initialpopulation' ,[1010], 'PopInitRange'
,[lb;ub] , 'plotfncs' ,@gaplotbestindiv)
[ xo_ga fo_ga ]= ga ( Fitness_F ,1,options);
Cmin = fo_ga
Tmin_Hours_ga = xo_ga
Tmin_days_ga = xo_ga /24
%*****

```

By running the proposed program after fifteen iterations, the results given in table 7 are obtained.

Table 7: T_{opt} and C_{min} obtained by genetic algorithm

Execution	$T_{opt-Hours}(x_0)$	$T_{opt-days}(x_0 / 24)$	$C_{min}(USD)$
1	1181.3	49.2192	2269.8
2	1181.3	49.2192	2269.8
3	1181.3	49.2192	2269.8
4	1256.4	52.3503	2278.6
5	1181.3	49.2192	2269.8
6	1188.9	49.5364	2269.9
7	1181.3	49.2192	2269.8
8	1181.3	49.2192	2269.8
9	1176.1	49.0061	2269.9
10	1181.3	49.2192	2269.8
11	1159.9	48.3273	2270.6
12	1148.8	47.8665	2271.6
13	1181.3	49.2192	2269.8
14	1181.2	49.2185	2269.8
15	1181.3	49.2190	2269.8

According to table 7, we observe a uniqueness of solution for the colored iterations. On the other hand, for the other iterations, we rather observe a diversity of solutions. However, we notice that the colored solutions correspond exactly to the optimal solution of our problem. Indeed, the minimum value of the cost and the durations of exploitations of the production unit are the same as those of the two preceding methods, therefore the method of the genetic algorithm is convergent.

4.7. Interpretation of results

The results obtained in this application show that the production unit will have to undergo a partial overhaul after 49 days of operation, and after five partial overhauls, i.e. more than eight months of operation, the production unit will receive a general revision.

5. Conclusion

Having reached the end of writing this paper, the objective of which was to minimize the preventive maintenance costs of a production unit, we first developed the cost model corresponding to an imperfect periodic maintenance policy with minimal repair, then we deployed two exact resolution methods (the simple derivation technique and the golden ratio method) and a stochastic method (the genetic algorithm), each time proposing a code on Matlab. It turns out that by implementing the different methods, a uniformity of the optimal solution is obtained, which well justifies the convergence of the genetic algorithm. Furthermore, thanks to the proposed Matlab code, we were able to determine the periodicity at which the production unit will undergo a general overhaul.

References

- [1] Sheu SH, Lin Y, Liao G. Optimum policies for a system with general imperfect maintenance. *Reliability Engineering and System Safety* 2006; 91: 362-369.
- [2] Pham H, Wang H. Imperfect maintenance. *European Journal of Operational Research* 1996; 94: 425-438.
- [3] Thomas LC A survey of maintenance and replacement models of multi-item systems. *Reliability Engineering* 1986; 16:297-309.
- [4] Valdez-Flores C, Feldman RM A survey of preventive maintenance models for hastically deteriorating single-unit systems. *Naval Research Logistics* 1989; 36: 419-446.
- [5] Cho ID, Parlar M. A survey of maintenance models for multi-unit systems. *European Journal of Operational Research* 1991; 51: 1-23.
- [6] Van Der Duyn Schouten F. Maintenance policies for multicomponent systems. In: Ozekici , S. (Ed.), *Reliability and maintenance of complex systems: And Overview. Reliability and Maintenance of Complex System* 1996: 154: 117-136.
- [7] Dekker R, Wildman RE, Van Der Duyn Schouten FA A review of multicomponent maintenance models with economic dependence. *Mathematical Methods of Operational Research* 1997; 45: 411-435.
- [8] Aupied J. *Experience feedback applied to the operational safety of equipment in operation.* Editions Eyrolles 1994.
- [9] Thomas M. *Reliability, predictive maintenance and machine vibration.* University of Quebec Press 2012.
- [10] Tahara A, Nishida T. Optimal replacement policy for minimal repair model. *Journal of Operations Research Society of Japan* 1975; 18: 113-124.
- [11] Nakagawa T. Optimal policy of continuous and discrete replacement with minimal repair at failure. *Naval Research Logistics Quarterly* 1984;31:543-550.
- [12] Sheu S, Kuo C, Nakagawa T. Extended optimal age replacement policy with minimal repair. *RAIRO: Operational Research* 1993;27:337-351.

- [13] Sheu S, Griffith W. S, Nakagawa T. Extended optimal replacement model with random minimal repair costs. *European Journal of Operational Research* 1995;83:636-649.
- [14] Block HW, Langberg NA, Savits TH Repair replacement policies. *Journal of Applied Probability* 1993: 30: 194-206.
- [15] Wang H, Pham H. Some maintenance models and availability with imperfect maintenance in production systems. *Annals of Operations Research* 1999: 91: 305-318.
- [16] Barlow RE, Hunter LC Optimum preventive maintenance policies. *Operations Research* 1960: 8: 90-100.
- [17] Liu X, Makis V, Jardine AKS A replacement model with overhauls and repairs. *Naval Research Logistics* 1995: 42:1063-1079.
- [18] Berg M, Epstein B. A modified block replacement policy. *Naval Research Logistics* 1976: 23: 15-24.
- [19] Tango T. Extended block replacement policy with used items. *Journal of Applied Probability* 1978: 15: 560-572.
- [20] Nakagawa T. A summary of periodic replacement with minimal repair at failure. *Journal of Operations Research Society of Japan* 1981a: 24:213-228.
- [21] Nakagawa T. Modified periodic replacement with minimal repair at failure. *IEEE Transactions on Reliability* 1981b: R-30 (2): 165-168.
- [22] Gertsbakh I. *Reliability Theory with applications to preventive maintenance*. Springer, Berlin 2002:34:1111-1114.
- [23] Zheng X, Fard N. A maintenance policy for repairable systems based on opportunistic failure rate tolerance. *IEEE Transactions on Reliability* 1991: 40: 237-244.
- [24] Jayabalan V, Chaudhuri D. Replacement policies: a near optimal algorithm. *IIE Transactions* 1995:27:784-788.
- [25] Nguyen DG, Murthy DNP Optimal repair limit replacement policies with imperfect repair. *Journal of Operational Research Society* 1981: 32: 409-416.
- [26] Kijima M, Nakagawa T. Replacement policies of a shock model with imperfect preventive maintenance. *European Journal of Operations Research* 1992:57: 100-110.
- [27] Hastings NA. The repair limit method. *Operational Research Quarterly* 1969: 20: 337-349.
- [28] Wang H, Pham H. Optimal maintenance policies for several imperfect maintenance models. *International Journal of Systems Science* 1996: 27: 543-549.
- [29] Nakagawa T, Osaki S. The optimum repair limit replacement policies. *Operational Research Quarterly* 1974: 25: 311-317.
- [30] Dohi T, Matsushima N, Kaio N, Osaki S. Nonparametric repair-limit replacement policies with imperfect repair. *European Journal of Operational Research* 1997: 96 (2): 260-273.
- [31] Puchinger J, Raidl. GR Combining metaheuristics and exact algorithms in combinatorial optimization: A survey and classification. In *proceedings of the first international work- coreference on the interplay between natural and artificial computation* 2005:41-53.
- [32] Land A. H, Doig AG An automatic method for solving discrete programming problems. *Econometrica* 1960:28(3):497-520.
- [33] Schrijver A. *Theory of linear and integer programming*. Wiley and Sons 1986.
- [34] Fletcher R, Reeves CM Function minimization by conjugal gradients. *Computer Journal* 1964:7: 148-154.
- [35] Fletcher R. *Practical Methods of Optimization*. John Wiley & Sons 1987.
- [36] Press WH *Numerical Recipes in C: The art of Scientific Computing*. Cambridge University Press 1992.
- [37] Culioli JC *Introduction to optimization*. Ellipsis 1994.
- [38] Minoux M. *Mathematical programming: Volume 1 Theory and algorithms*. Ed. Dunod 1983.
- [39] Tabeb M. Parallelization of a genetic algorithm for the single-machine scheduling problem with sequence-dependent setup times. *University of Quebec at Chicoutimi* 2008.

- [40] Benhaddad H, Belabbas M. Parallel genetic algorithm for flow shop scheduling. Master's thesis, University of Msila 2022.
- [41] Reichenberg I. Cybernetic solution path of an experimental problem. Library Translation No.1122, Royal Aircraft Establishment, Farnborough, UK 1965.
- [42] Holland JH. Adaptation in natural and artificial systems. University of Michigan press 1975.
- [43] Goldberg D. Genetic algorithms in search, optimization and machine learning. Addison Wesley 1989.
- [44] Schewefel HP. Numerical Optimization of computer models. Wiley Publishing 1981.
- [45] Bicking F, Fonteix C, Corriou JP, Marc I. Global optimization by artificial life: a new technique using genetic population evolution. *RAIRO-Operations Research* 1994; 28 (1): 23-36.
- [46] Digalakis JG, Margaritis KG. On benchmarking functions for genetic algorithms. *International Journal of Computer Mathematics* 2001;77(4): 481-506.
- [47] Jason GD, Konstantinos GM. An experimental study of benchmarking functions for genetic algorithms. *International Journal of Computer Mathematics* 2002; 79(4): 403-416.
- [48] Koumoussis VK, Katsaras CP. A saw-tooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance. in *IEEE Transactions on Evolutionary Computation* 2006; 10 (1): 19-28.
- [49] Vedat T, Ayse TD. An improved genetic algorithm with initial population strategy and self-adaptive member grouping. *Computers & Structures* 2008; 86: 1204-1218.
- [50] Wei C, Chi W, Yajun W. Scalable influence maximization for prevalent viral marketing in large-scale social networks. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – KDD 2010*: 10: 1029-1038.
- [51] Nasr A, Elmekawy TY. Robust and stable flexible job shop scheduling with random machine breakdowns using a hybrid genetic algorithm. *Int. J. Production Economics* 2011;132:279-291.
- [52] Hongbin D, Tao L, Rui D, Jing S. A novel hybrid genetic algorithm with granular information for feature selection and optimization. *Applied Soft Computing* 2018; 65:33-46.
- [53] Ahmad H, Khalid A, Esra'a A, Eman A, Awni H, Surya Prasath VB. Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information* 2019;10 (390):1-36.
- [54] Ailiang Q, Dong Z, Fanhua Y, Ali AH, Zongda W, Zhennao C, Fayadh A, Romany FM, Huiling C, Mayun C. Directional mutation and crossover boosted ant colony optimization with application to COVID-19 X-ray image segmentation. *Computers in Biology and Medicine* 2022:148.
- [55] Sethembiso Nonjabulo L, Akshay Kumar S. Effects of Particle Swarm Optimization and Genetic Algorithm Control Parameters on Overcurrent Relay Selectivity and Speed. in *IEEE Access* 2022; 10: 4550 – 4567.
- [56] Junfeng Z, Yanhui Z, Yubo Z, Wen-Long S, Zhile Y, Wei F. Parameters identification of photovoltaic models using a differential evolution algorithm based on elite and obsolete dynamic learning. *Applied Energy* 2022: 314.
- [57] Guo X, Wei T, Wang, Liu S, Qin S, Qi L. Multiobjective u-shaped disassembly line balancing problem considering human fatigue index and an efficient solution. in *IEEE Transactions on Computational Social Systems* 2023; 10 (4): 2061-2073.
- [58] Agushaka JO, Ezugwu AE, Abualigah L et al. Efficient initialization methods for population-based metaheuristic algorithms: a comparative study. *Arch Computat Methods Eng* 2023; 30: 1727–1787.
- [59] Asha A, Rajesh A, Poonguzhali I, Shabana U, Salem A. Optimized RNN-based performance prediction of IoT and WSN-oriented smart city application using improved honey badger algorithm. *Measurement* 2023: 210.
- [60] Yong W, Zhen L, Gai-Ge W. Improved differential evolution using two-stage mutation strategy for multimodal multi-objective optimization. *Swarm and Evolutionary Computation* 2023:78.