Kragujevac Journal of Mathematics
Volume 50(10) (2026), Pages 1569–1583.

A GENETIC ALGORITHM MODEL IMPROVED WITH ZECKENDORF REPRESENTATIONS FOR PREVENTIVE MAINTENANCE SCHEDULING PROBLEM

YUNUS EMRE GÖKTEPE[1], FIKRI KÖKEN[1*], AND HALIME ERGUN[1]

Abstract. This study introduces a novel Genetic Algorithm (GA) model enhanced with Zeckendorf representation and Fibonacci number-based encoding to optimize preventive maintenance scheduling (PMS) problems. Conventional maintenance scheduling methods, based on random or linear encoding techniques, often fail to optimize maintenance processes effectively.

Therefore, the proposed model aims to systematically plan maintenance periods and minimize production interruptions by encoding maintenance intervals using Zeckendorf representation.

By optimizing maintenance processes, the proposed model enhances system production continuity. Experimental analyses indicate that the proposed model enhances existing production capacity and facilitate a more balanced management of maintenance operations. The electricity and water production capacities increased by 11% and 10%, respectively, while the reserve capacity improved by 9% for electricity and 17% for water.

These results show that the proposed method is a new optimization strategy for maintenance planning by enhancing the applicability of GA in preventive maintenance scheduling problems. Optimizing maintenance scheduling with Zeckendorf representation enables systematic and balanced execution of maintenance operations, thereby ensuring more efficient planning in industrial facility maintenance processes.

Key words and phrases. Fibonacci numbers, Zeckendorf representations, genetic algorithm, preventive maintenance.

2020 *Mathematics Subject Classification.* Primary: 11B39, 68W50. Secondary: 11B83.

https://doi.org/10.46793/KgJMat2610.1569G

Received: September 25, 2024.

Accepted: February 20, 2025.

1. Introduction

Industrial facilities, power plants, and other critical infrastructures emphasize maintenance strategies due to their need for operational continuity. Maintenance processes not only extend the lifespan of equipment but also play a critical role in ensuring operational continuity and preventing sudden failures. In this context, Preventive Maintenance (PM) is a strategy that involves the planned and periodic maintenance of machines or systems. PM aims to inspect equipment at specific intervals, perform maintenance or repairs, and minimize unexpected failures and interruptions [7].

When compared to other maintenance strategies such as reactive maintenance (running to failure) and predictive maintenance (forecasting using sensor data), PM stands out for its planned and systematic nature. Preventive Maintenance Scheduling (PMS), on the other hand, involves planning when and how the specified maintenance activities will be carried out. Effective implementation of PMS optimizes maintenance processes, reduces operational costs, increases productivity, and ensures system reliability [1]. Traditional PMS approaches often employ deterministic methods, leading to the development of linear and nonlinear optimization models. Research utilizing Mixed Integer Linear Programming (MILP) and heuristic methods has been published [5, 13]. Studies have been conducted to optimize maintenance planning for power generation units, minimize maintenance costs, and create appropriate maintenance schedules [16].

However, in large-scale and complex maintenance planning problems, traditional methods are often computationally intensive and time-consuming in achieving optimal solutions. Therefore, heuristic and meta-heuristic algorithms, such as Genetic Algorithm (GA), Tabu Search (TS), and Particle Swarm Optimization (PSO), are widely preferred. In recent years, hybrid models combining these approaches have been introduced, improving solution quality and the efficiency of the optimization process [2, 3].

The rapid growth of computational optimization has led to the development of numerous algorithms aimed at solving complex scheduling and resource allocation problems. Among these, Genetic Algorithms (GAs) have emerged as versatile and powerful tools for addressing various optimization challenges due to their ability to explore large search spaces and locate near-optimal solutions [3, 18]. Traditional GAs typically rely on binary or floating-point representations to encode solutions, but these methods often struggle to balance diversity and convergence.

The methods employed to solve PMS problems remain open to further improvement. Enhancing the solution quality and success of proposed methods is both feasible and necessary. Motivated by this, the study proposes a novel preventive maintenance scheduling method by incorporating Fibonacci-based Zeckendorf and k-Zeckendorf representations into a hybrid GA-TS approach. Fibonacci-based representations aim to improve system performance by distributing maintenance durations and intervals more systematically and evenly. The effectiveness of the proposed approach is analyzed in terms of the supply-demand balance in electricity and water production.

This study introduces a novel enhancement to the standard GA framework by integrating Zeckendorf representations, a numerical encoding technique based on Fibonacci numbers. Unlike traditional representations, Zeckendorf encoding leverages the unique properties of Fibonacci numbers to provide a compact, efficient, and inherently diverse representation of solutions. This method has been applied to the preventive maintenance scheduling problem, a critical optimization challenge faced by industries such as cogeneration plants that rely on continuous operations. The scheduling problem requires balancing equipment availability, resource utilization, and demand fulfillment while minimizing operational disruptions.

The significance of this study lies in the potential to improve the performance of genetic algorithms through a structured yet adaptable coding schema. The proposed method not only achieves better optimization results by reducing redundancy and increasing the diversity of the search space, but also provides a new perspective on how numerical representations can affect algorithmic efficiency.

In the following sections of this paper, existing PMS approaches in the literature are discussed, the details of the proposed method are explained, and the performance of the method is evaluated based on experimental results.

2. Mathematical Model

This section explains how the proposed genetic algorithm approach has been developed with a mathematical approach. For this purpose, detailed explanations of Fibonacci base and Zeckendorf representations are provided, and the computation processes are demonstrated.

In number theory, there exist various alternative methods for representing a number A. These representations often depend on the properties of A and the specific mathematical framework being employed. Such methods can provide deeper insights into the structure and behavior of numbers, offering tools for analyzing their relationships and applications. Among these representations, one of the most notable is the representation using Fibonacci numbers, commonly referred to as the *Fibonacci representation*. It is well established [12] that the Fibonacci sequence, denoted by $\{F_n\}$, is defined recursively as follows:

$$F_n = F_{n-1} + F_{n-2}, \quad \text{if } n > 1,$$

with initial values $F_0 = 0$ and $F_1 = 1$. Excluding the initial values $F_0 = 0$ and $F_1 = 1$ of the Fibonacci sequence, the subsequent 12 Fibonacci numbers and their corresponding indices are summarized as follows in Table 1.

For example, every positive integer A can be written as

$$A = F_{n_1} + F_{n_2} + \cdots + F_{n_r}, \quad n_1 > n_2 > \cdots > n_r \geq 2,$$

which is called a *Fibonacci representation* of A.

TABLE 1. The F_n Fibonacci numbers for $2 \leq n \leq 13$

n	2	3	4	5	6	7	8	9	10	11	12	13
F_n	1	2	3	5	8	13	21	34	55	89	144	233

The most widely recognized method for representing non-negative integers using only the digits $\{0, 1\}$ is the binary numeral system. However, alternative methods exist for representing integers using just $\{0, 1\}$, some of which are less immediately intuitive. One notable example is *Zeckendorf's Fibonacci representation*.

Zeckendorf's theorem is a well-known result that states every positive integer can be uniquely expressed as a sum of distinct Fibonacci numbers, provided that the indices of the Fibonacci numbers satisfy the condition $n_j - n_{j-1} \geq 2$, $j = 1, 2, \ldots, r$, $n_r \geq 2$. This unique representation is referred to as *Zeckendorf's Fibonacci representation* [10, 11]. The Zeckendorf theorem offers an alternative to the binary numeral system, providing a unique representation of integers that has practical applications in areas such as data transmission and compression [17].

Zeckendorf's theorem states that every positive integer A can be uniquely expressed as the sum of one or more distinct Fibonacci numbers. This representation ensures that the sum excludes any two consecutive Fibonacci numbers [10]. Zeckendorf's Fibonacci encoding utilizes this unique representation. Specifically, any positive integer $A = (\ldots d_3 d_2 d_1)_{\text{fib}}$ can be expressed as

$$A = \sum_{i=1}^{n} d_i F_{n_i}$$

where F_{n_i}, $n_i \geq 2$ is the n_i^{th} Fibonacci number, $n_i - n_{i-1} \geq 2$, $d_i \in \{0, 1\}$, and $d_n = 1$. This encoding is referred to as *Zeckendorf's Fibonacci coding* and is denoted as $A = (d_i)_{\text{fib}}$, $i = 1, 2, \ldots$

For $A \leq 233$, a 12-bit $\{0, 1\}$ array is generated in Zeckendorf's Fibonacci representation. For instance, the representation of $A = 123$ is given by the sum $89 + 34$. In Zeckendorf's Fibonacci coding, this can be written as $123 = (1010000000)_{\text{fib}}$.

A number representation system is generally most effective when it provides a unique representation for every integer. Zeckendorf's Fibonacci encoding ensures uniqueness by prohibiting consecutive 1's in the representation. However, if this condition is relaxed, while every number can still be expressed as a sum of Fibonacci numbers, some numbers will have multiple possible sums, leading to non-unique representations.

It can be observed that the number $A = 123$ can be expressed in 7 different ways using various combinations of Fibonacci numbers. To address this non-uniqueness, let us slightly modify Zeckendorf's Fibonacci encoding to establish a new $\{0, 1\}$-based representation that ensures a single, unique form for each number.

In [4], a Zeckendorf-Wythoff sequence is introduced, which organizes positive integers into columns based on their Zeckendorf representations. The n^{th} column of the Zeckendorf-Wythoff sequence consists of integers A, listed in ascending order, whose

Zeckendorf representations terminate with F_{n+1}. The Zeckendorf-Wythoff sequence is defined such that each row corresponds to a Fibonacci sequence. If the largest Fibonacci number in the Zeckendorf representation of A is F_n, then A begins with F_n.

Each row of the sequence consists of A integers, all having Zeckendorf representations of the same structure. In particular, for any two consecutive numbers in the row, the indices of the Fibonacci numbers F_n corresponding to the digits in their Zeckendorf representations differ by exactly one [9, 10].

Let $R(A)$ represent the number of ways the non-negative integer A can be expressed as the sum of distinct Fibonacci numbers. It is given that for integers A in the odd-numbered columns of a row, the successive values of $R(A)$ form an arithmetic progression, and $R(A-1)$ remains constant. Moreover, if there exists an A such that $F_n < A < F_{n+1} - 1$ in a column of the Zeckendorf-Wythoff sequence, then $A + F_{n+k}$, $k \geq 2$, also belongs to the same column. These properties of Zeckendorf representations and Wythoff pairs are utilized to determine $R(A)$ [10, 11].

The Zeckendorf-Wythoff sequence is presented in Table 2, arranged into 14 columns, as it corresponds to the use of 8, 12, and 14-bit arrays.

TABLE 2. Zeckendorf-Wythoff sequence

$w_{(i,j)}$	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$w_{(1,j)}$	1	2	3	5	8	13	21	34	55	89	144	233	377	610
$w_{(2,j)}$	4	7	11	18	29	47	76	123	199	322	521	843	1364	2207
$w_{(3,j)}$	6	10	16	26	42	68	110	178	288	466	754	1220	1974	3194

Table 3 provides the number of distinct Fibonacci representations for each number A, organized according to the columns in Table 2.

TABLE 3. $R(A)$ values for numbers A given in Table 2

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14
$R(w_{(1,j)})$	1	1	2	2	3	3	4	4	5	5	6	6	7	7
$R(w_{(2,j)})$	1	1	3	3	5	5	7	7	9	9	11	11	13	13
$R(w_{(3,j)})$	2	2	4	4	6	6	8	8	10	10	12	12	14	14

The number of representations of A as sums of distinct Fibonacci numbers can be derived from the Zeckendorf representation of A. Specifically, if the Zeckendorf representation of A ends with F_n, where $n \geq 2$, then a constant q satisfies:

$$R(A) = R(A-1)R(F_n) - q, \quad 0 \leq q \leq R(A-1).$$

Here, $R(F_n) = \lfloor n/2 \rfloor$ and $R(F_n - 1) = 1$ for $n \geq 1$, where $\lfloor x \rfloor$ denotes the greatest integer less than or equal to a real number x.

Table 3 clearly demonstrates that the different Fibonacci representations of the numbers A in each row increase to the right, following the pattern outlined in Table 2.

Furthermore, this increase occurs at a faster rate in the lower rows. This indicates that smaller numbers in the lower rows can possess a greater number of representations. For instance, the number A can have six different representations if $A = 144$, seven different representations if $A = 123$, and eight different representations if $A = 110$.

Although $R(A)$ indicates the number of different representations of the number A, detailed information about these representations can only be explicitly expressed using the Zeckendorf representation. An alternative canonical representation can be developed to express any given number as a sum of Fibonacci numbers, while permitting the inclusion of consecutive Fibonacci numbers within the same sum. This approach is facilitated by adopting a base system where Fibonacci numbers act as placeholders, thereby allowing adjacent terms. The validity of this method stems from the property that the sum of any two consecutive Fibonacci numbers equals the next Fibonacci number. Consequently, any representation of the form $(\ldots 100 \ldots)_{\text{fib}}$ can be systematically replaced with $(\ldots 011 \ldots)_{\text{fib}}$. In this study, each instance of $(\ldots 100 \ldots)_{\text{fib}}$ was transformed into $(\ldots 011 \ldots)_{\text{fib}}$. This alternative representation not only serves as a canonical form but also provides a more compact encoding compared to Zeckendorf's Fibonacci representation.

To verify that every number can be represented in this system, one can write down the k-Zeckendorf representations of several numbers using 12 bits. By constructing such representations, it becomes evident how numbers can be uniquely expressed following the rules of the k-Zeckendorf system. For clarity, a table can be provided to illustrate these representations. Table 4 includes examples of numbers along with their corresponding k-Zeckendorf representations in 12-bit format.

TABLE 4. Representations of k-Zeckendorf using Zeckendorf's Fibonacci coding

N	Zeckendorf's code	1-Zeckendorf's code	...	6-Zeckendorf's code	7-Zeckendorf's code	8-Zeckendorf's code
68	100100000	100011000	...	011010110	-	-
123	1010000000	1001100000	...	0111011000	0111010110	-
178	10010000000	10001100000	...	01101100000	01101011000	01101010110
233	100000000000	011000000000	...	010101010110	-	-

The least number of Fibonacci numbers used in a representation corresponds to the number of 1's in Zeckendorf's Fibonacci coding. This is because Zeckendorf's theorem guarantees the use of the minimal number of Fibonacci numbers in any valid representation, which is why it is also referred to as the minimal Fibonacci representation. Consequently, any representation of the form $(...100...)_{\text{fib}}$, where a 1 is followed by two or more 0's, can be systematically replaced with $(..011..)_{\text{fib}}$ by redistributing the terms according to the Zeckendorf rules. This replacement not only adheres to the uniqueness property of Zeckendorf's representation but also ensures the representation remains minimal by avoiding consecutive Fibonacci numbers and maintaining the fewest possible 1's in the coding.

3. Genetic Algorithms for Preventive Maintenance Scheduling

Genetic algorithms (GA) are a powerful computational method inspired by natural selection and biological evolution processes. This technique is considered one of the most common applications of evolutionary computing methods used to solve complex search and optimization problems. The concept of genetic algorithms was first introduced by John Holland [8] and was further developed by his students and colleagues in the following years, making it applicable to a wide range of optimization problems.

The fundamental philosophy of GA is to evolve the population by selecting solutions with high fitness values and applying genetic operators (such as crossover, mutation, etc.) [14]. This approach enables effective exploration of the vast solution space while reducing the risk of getting stuck in local minima. It is particularly advantageous in terms of flexibility and applicability for combinatorial and constrained optimization problems.

The Preventive Maintenance Scheduling problem involves optimizing maintenance activities to ensure the efficient operation of critical systems while minimizing downtime and costs [15]. Given the combinatorial nature of PMS, exact methods become computationally infeasible for large-scale problems. Consequently, heuristic and metaheuristic approaches, particularly GAs, have garnered significant attention in the literature due to their ability to provide near-optimal solutions within reasonable computation timeframes.

GAs are population-based optimization techniques inspired by natural selection and evolutionary principles. In the context of PMS, GAs encode maintenance schedules as chromosomes that evolve over multiple generations through selection, crossover, and mutation operators. The fitness function evaluates each schedule based on criteria such as total energy production, water supply, maintenance costs, and demand-supply balance.

The application of GA to PMS problems follows specific fundamental steps [6]. First, maintenance schedules are encoded as chromosomes. In this encoding process, each gene represents the maintenance status of a unit in a specific week; that is, the unit is either operational or undergoing maintenance during that week. This transformation allows maintenance processes to be structured in a format that can be processed by the genetic algorithm.

Next, an initial population is generated. The initial population is typically produced randomly; however, certain heuristic methods suitable for the problem's nature can also be employed. This process is crucial to enable the genetic algorithm to search a wide solution set.

In each generation, the chromosomes created are evaluated based on a fitness function. The fitness value measures how well the maintenance schedules meet energy and water demands and whether they minimize production interruptions caused by maintenance. More suitable maintenance plans create a more balanced production-consumption relationship by improving the system's operational efficiency.

One of the key components of genetic algorithms is the selection process, which ensures that individuals with high fitness are chosen to reproduce in the next generation. Among the most commonly used selection methods are roulette wheel selection and tournament selection, both of which aim to pass on high-performing chromosomes through generations.

The crossover process is performed by exchanging genetic information between two selected parent chromosomes. This operation enables the creation of potentially better maintenance schedules and enhances genetic diversity, facilitating the exploration of a broader solution space.

The mutation process is applied to prevent premature convergence of the genetic algorithm and to allow further exploration of the solution space. By introducing small changes in randomly selected chromosomes, genetic diversity is preserved, and alternative solutions can be discovered.

Finally, the algorithm terminates when a predefined number of generations is reached or when the fitness function converges to a certain level. At this point, the best chromosome obtained is considered the final solution for maintenance scheduling.

4. New Models And Solutions

In this study, a novel GA model based on Zeckendorf representation and Fibonacci number decomposition is proposed to enhance the efficiency of preventive maintenance scheduling (PMS). The primary objective of the proposed approach is to optimize the scheduling of preventive maintenance activities for electricity generation and water desalination units while minimizing deviations between production and demand. Unlike traditional GA models that use direct encoding schemes, this study employs k-Zeckendorf representation to structure maintenance schedules more efficiently, thereby improving the search process and convergence speed.

The proposed model integrates three key components: (1) initial population generation using k-Zeckendorf representation, (2) a customized fitness function based on production-demand deviation, and (3) genetic operations (selection, crossover, and mutation) adapted to k-Zeckendorf-based scheduling.

Maintenance schedules for turbines and distillers are initialized using k-Zeckendorf representation, which ensures that each maintenance period is uniquely encoded in terms of Fibonacci numbers. This encoding reduces redundancy in schedule representation and enables the algorithm to explore feasible solutions more efficiently. Maintenance initiation weeks are selected based on k-Zeckendorf-decomposed values, ensuring that units undergo preventive maintenance without clustering within the same periods.

In preventive maintenance scheduling, electricity and water production are affected by maintenance periods, leading to fluctuations in meeting demand. Therefore, optimal maintenance scheduling must ensure appropriate allocation of production capacities. The fitness function evaluates the deviation between available electricity and water production and demand values, defined as follows:

$$F = \sum_{t=1}^{T} \left[w_e |P_e(t) - D_e(t)| + w_w |P_w(t) - D_w(t)| \right],$$

where F represents the fitness function, T denotes the total number of weeks, P_e and D_e are the electricity production and demand, respectively, P_w and D_w are the water production and demand, respectively, w_e represents the weight coefficient for the electricity production-demand deviation, and w_w denotes the weight coefficient for the water production-demand deviation.

Electricity production may decrease due to maintenance on turbines. The total electricity production per week is calculated as follows:

$$P_e(t) = \sum_{i=1}^{N_T} \left(A_i(t) \cdot C_{T,i} \right),$$

where N_T is the total number of turbines, $A_i(t)$ is a binary variable indicating whether the turbine is under maintenance (1 = operational, 0 = under maintenance), and $C_{T,i}$ denotes the capacity (MW) of the i-th turbine. Similarly, water production decreases due to maintenance on distillers. The weekly water production is calculated as follows:

$$P_w(t) = \sum_{j=1}^{N_D} \left(B_j(t) \cdot C_{D,j} \right),$$

where N_D is the total number of distillers, $B_j(t)$ is a binary variable indicating whether the distiller is under maintenance (1 = operational, 0 = under maintenance), and $C_{D,j}$ represents the capacity (MIGD) of the j-th distiller.

Thus, the fitness function penalizes large production deficits, especially during critical demand periods, guiding the genetic algorithm towards solutions that optimize unit maintenance intervals while maintaining supply-demand balance.

The fitness function evaluates the difference between the available electricity and water production and the corresponding demand values. It is defined as follows:

The proposed GA model is applied to a PMS problem that includes eight turbines and sixteen distillers operating under predefined demand constraints. Each unit undergoes scheduled maintenance for a predetermined period-four weeks for turbines and five weeks for distillers. The numerical values for these units are presented in Table 5, derived from literature to enable a fair and effective comparison [3].

The GA iterates over 100 generations with a population size of 50, dynamically adjusting schedules based on varying fitness values.

To evaluate the effectiveness of the proposed approach, weekly electricity and water production levels are analyzed in comparison with demand values. Additionally,

Table 5. Numerical Values of Units

Unit Type	Number of Units	Maintenance Duration (Weeks)	Production Capacity
Turbines	8	4	47,040 MW per unit
Distillers	16	5	50.4 MIGD (12 units) 40.2 MIGD (4 units)
Boilers	8	5	Support Only (No Direct Production)

total annual production, demand, and production surplus are calculated to assess system performance. The resulting schedules demonstrate that maintenance activities are well-distributed, production shortages are prevented, and resource utilization is optimized.

5. Experimental Results

This section evaluates the performance of the proposed genetic algorithm and k-Zeckendorf representation-based PMS model. The production and consumption values of electricity and water, total production surplus, and fitness function values are analyzed.

5.1. **Experimental Setup.** The proposed genetic algorithm and k-Zeckendorf representation based preventive maintenance scheduling model in this study was implemented on a computer with an Intel Core i7-12700H processor and 16 GB of RAM. The algorithm was implemented using Python 3.8 programming language, and libraries such as NumPy and Matplotlib were utilized.

To assess the model's performance, the optimization process was executed with a population of 50 individuals over 100 generations. The core parameters of the genetic algorithm are presented in Table 6.

Table 6. Key parameters of the proposed model

Parameter	Value	Description
Population Size	50	Number of individuals evaluated simultaneously
Number of Generations	100	Evolution cycles in the solution space
Crossover Rate	0.8	Probability of creating new generation individuals
Mutation Rate	0.1	Probability of applying random mutation
Selection Method	Tournament Selection	Method for selecting individuals for crossover
Encoding Type	k-Zeckendorf	Maintenance scheduling encoded using Zeckendorf representation

Preventive maintenance schedules were generated for both electricity generation units (turbines) and water treatment units (distillers). Each turbine was assigned a maintenance period of four weeks, while each distiller was assigned five weeks. Using Zeckendorf representation, maintenance intervals were systematically planned in Fibonacci-based sequences, ensuring a balanced distribution of maintenance operations across the system.

5.2. **Results and Analysis.** This section analyzes the experimental results of the proposed genetic algorithm and the k-Zeckendorf representation-based Preventive Maintenance Scheduling model. The study evaluates key performance metrics, including electricity and water production, supply-demand balance, total fitness function value, and convergence analysis. Additionally, the proposed method is compared with the conventional GA to assess the impact of Zeckendorf-based encoding on the optimization process.

To evaluate the proposed model's ability to optimize maintenance scheduling, the weekly electricity and water production values were compared with demand values. Figure 1 compares electricity production values against demand and also displays the surplus production amount.

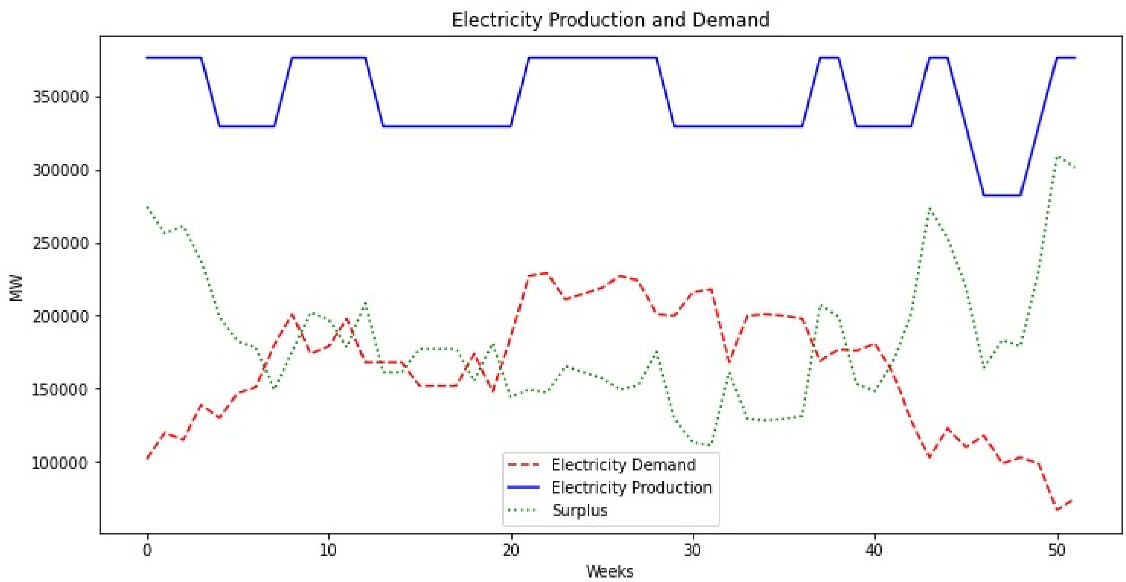

FIGURE 1. Comparison of electricity production values with demand

As seen in Figure 1, electricity production consistently meets demand; however, short-term production declines occur during maintenance periods. Nevertheless, encoding maintenance scheduling using k-Zeckendorf representation ensures a balanced distribution of maintenance periods across the system, preventing these declines from causing disruptions.

Similarly, Figure 2 presents the relationship between water production and demand. Due to the proposed maintenance planning approach, fluctuations in water production have been largely controlled outside critical maintenance periods, ensuring that demand is met throughout the year. Production deficits during maintenance processes have not led to any service interruptions.

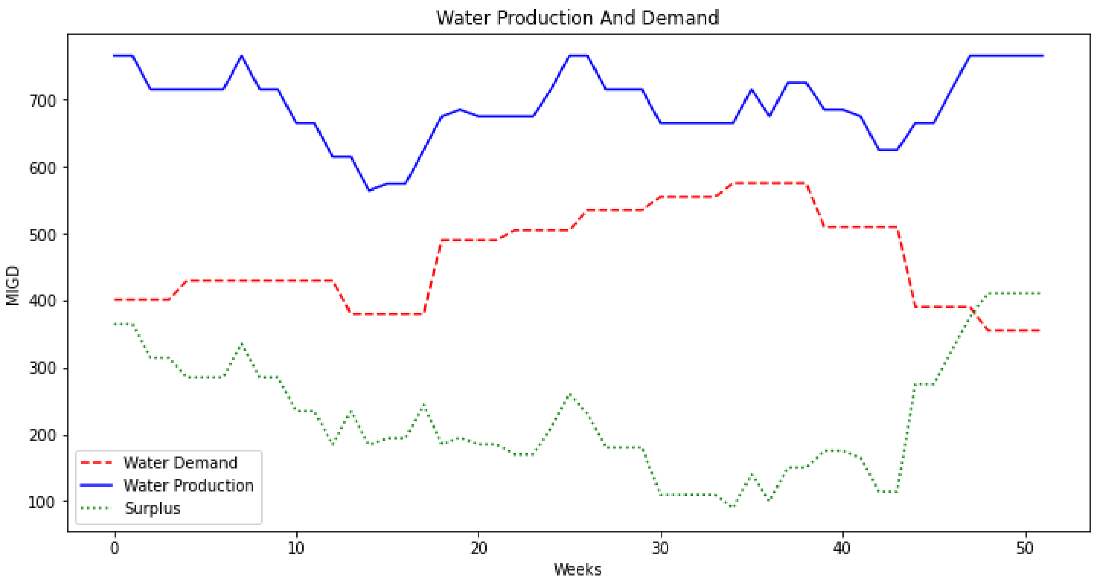

FIGURE 2. Comparison of Water Production with Demand

When the results of the proposed method are compared with studies that solve the same problem using genetic algorithms, it is observed that the proposed approach achieves superior outcomes. Table 7 provides a comparative evaluation of the results obtained by the model.

TABLE 7. Comparative Results of the Proposed Model

Models	GA+TS [3]	Proposed Model	Improvement Rate
Available Electricity (MW)	17,687,040.00	19,568,640.00	11%
Electricity Demand (MW)	8,590,000.00	8,590,000.00	-
Surplus (MW)	9,097,040.00	10,978,640.00	21%
Reserve Percentage	51%	56%	9%
Available Water (MIGD)	36,321.60	39,811.20	10%
Water Demand (MIGD)	24,130.00	24,130.00	-
Surplus (MIGD)	12,191.60	15,681.20	29%
Reserve Percentage	34%	39%	17%

The results presented in Table 7 demonstrate that the proposed *k-Zeckendorf representation*-based genetic algorithm model provides significant performance improvements compared to the conventional Hybrid GA-TS model. According to the comparisons made in terms of electricity and water production, the proposed model was observed to increase existing electricity generation by 11% and water production by 10%.

Notably, in terms of reserve capacity-i.e., the system's production capacity beyond demand-an improvement of 9% in electricity and 17% in water was achieved. This indicates that the operational flexibility of the facility has increased due to a more balanced distribution of maintenance processes and more effective management of production interruptions.

Furthermore, the 21% increase in surplus electricity production and the 29% increase in surplus water production reveal that the system possesses greater backup capacity and that maintenance processes have a reduced impact on critical production periods. This finding confirms that the proposed model minimizes production interruptions and responds more effectively to demand fluctuations.

The findings indicate that the *k-Zeckendorf representation*-based maintenance scheduling enhances system production continuity by structuring maintenance planning more efficiently. In particular, Zeckendorf encoding, which optimizes the search space within the evolutionary process of genetic algorithms, has enabled systematic rather than random scheduling of maintenance periods. Consequently, this has increased reserve capacity and strengthened production reliability.

Based on these results, it can be concluded that the proposed model offers a feasible, efficient, and high-performance solution for PMS problems.

6. Conclusion

This study proposed a genetic algorithm model incorporating Zeckendorf representation and Fibonacci-based encoding to address preventive maintenance scheduling problems. Traditional methods relying on random or linear encoding approaches often struggle to systematically plan maintenance periods, leading to production interruptions. By optimizing maintenance scheduling using Zeckendorf encoding, the proposed model effectively minimized production disruptions and improved the supply-demand balance.

Experimental results demonstrated that the proposed model increased electricity production capacity by 11% and water production capacity by 10%. Reserve capacity improved by 9% for electricity and 17% for water, reinforcing production flexibility.

The optimization process achieved faster convergence than conventional methods, reducing solution time. This study enhances the applicability of genetic algorithms for PMS problems, improving their effectiveness in maintenance scheduling. The integration of Zeckendorf representation with genetic algorithms has proven to be a more balanced and effective solution, particularly for large-scale maintenance planning problems. Future research will focus on applying this model to different energy plants, production facilities, and infrastructure systems. Further development will explore hybrid approaches and other metaheuristic optimization techniques to enhance model performance.

References

[1] E. Aghezzaf, M. Jamali and D. Ait-Kadi, *An integrated production and preventive maintenance planning model*, European Journal of Operational Research **181** (2007), 679–685.

[2] K. Alhamad, M. Alardhi and A. Almazrouee, *Preventive maintenance scheduling for multicogeneration plants with production constraints using genetic algorithms*, Adv. Oper. Res. **2015** (2015), 1–12.

[3] K. Alhamad and Y. Alkhezi, *Hybrid genetic algorithm and tabu search for solving preventive maintenance scheduling problem for cogeneration plants*, Mathematics **12** (2024), Article ID 1881.

[4] M. Bicknell-Johnson, *The Zeckendorf-Wythoff array applied to counting the number of representations of N as sums of distinct Fibonacci numbers*, in: F. T. Howard (Ed.), *Applications of Fibonacci Numbers*, Springer Netherlands, Dordrecht, 1999, 53–60.

[5] S. Chansombat, P. Pongcharoen and C. Hicks, *A mixed-integer linear programming model for integrated production and preventive maintenance scheduling in the capital goods industry*, International Journal of Production Research **57** (2019), 61–82.

[6] D. Dumitrescu, B. Lazzerini, L. C. Jain and A. Dumitrescu, *Evolutionary Computation*, CRC Press, 2000.

[7] G. G. Güner, C. T. Sakar and B. Yet, *A multicriteria method to form optional preventive maintenance plans: A case study of a large fleet of vehicles*, IEEE Transactions on Engineering Management **70** (2023), 2153–2164.

[8] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, The MIT Press, 1992.

[9] F. T. Howard (Ed.), *Applications of Fibonacci Numbers*, Springer Netherlands, Dordrecht, 1999.

[10] C. Kimberling, *The Zeckendorf array equals the Wythoff array*, Fibonacci Quart. **33** (1995), 3–8.

[11] C. Kimberling, *Edouard Zeckendorf*, Fibonacci Quart. **36** (1998), 416–418.

[12] T. Koshy, *Fibonacci and Lucas Numbers with Applications*, Vol. 2, John Wiley & Sons, 2019.

[13] R. Manzini, R. Accorsi, T. Cennerazzo, E. Ferrari and F. Maranesi, *The scheduling of maintenance. A resource-constraints mixed integer linear programming model*, Computers & Industrial Engineering **87** (2015), 561–568.

[14] M. Mitchell, *An Introduction to Genetic Algorithms*, MIT Press, 1998.

[15] F. Moinian, H. Sabouhi, J. Hushmand, A. Hallaj, H. Khaledi and M. Mohammadpour, *Gas turbine preventive maintenance optimization using genetic algorithm*, International Journal of System Assurance Engineering and Management **8** (2017), 594–601.

[16] M. Mollahassani-pour, A. Abdollahi and M. Rashidinejad, *Application of a novel cost reduction index to preventive maintenance scheduling*, International Journal of Electrical Power & Energy Systems **56** (2014), 235–240.

[17] P. Pooksombat, P. Udomkavanich and W. Kositwattanarerk, *Multidimensional Fibonacci coding*, Mathematics **10** (2022), Article ID 386.

[18] I. H. Sin and B. D. Chung, *Bi-objective optimization approach for energy aware scheduling considering electricity cost and preventive maintenance using genetic algorithm*, Journal of Cleaner Production **244** (2020), Article ID 118869.

[1]Department of Computer Engineering,
Seydisehir Ahmet Cengiz Faculty of Engineering,
Konya, Türkiye

*Corresponding Author
Email address: ygoktepe@erbakan.edu.tr
ORCID iD: https://orcid.org/0000-0002-8252-2616
Email address: fkoken@erbakan.edu.tr
ORCID iD: https://orcid.org/0000-0002-8304-9525
Email address: hboztoprak@erbakan.edu.tr
ORCID iD: https://orcid.org/0000-0003-1634-9744