Int. J. Nonlinear Anal. Appl. 11 (2020) No. 1, 367-379
ISSN: 2008-6822 (electronic)
https://dx.doi.org/10.22075/ijnaa.2020.4311

Generator maintenance scheduling using discrete firefly algorithm

Mohsen Saffarian[a], Seyed Mahmood Kazemi[a], Mohammad Baharshahi[b,*]

[a]*Department of Computer and Industrial Engineering, Birjand University of Technology, Birjand, Iran*
[b]*Department of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran*

(Communicated by Madjid Eshaghi Gordji)

Abstract

Maintenance scheduling of power generators is of particular importance for power grids from both economic and reliability aspects. In this paper, the discrete firefly algorithm is implemented for maintenance scheduling of generators. This project utilizes the combination of the discrete firefly algorithm, as the main tool, and heuristic methods, as the secondary tool, for constructing initial solutions and searching the solution space. The paper investigates a case study consisting of a 32-generator problem formulated as a mixed-integer problem. The performance of the proposed method is compared with those in the literature and its strengths and weaknesses are discussed. The obtained results showed that performance of the proposed algorithm is not highly affected by its parameters' values and is capable of providing multiple efficient maintenance schedules in a desirable time.
keyword Generator maintenance scheduling, Discrete firefly algorithm, Reliability

1. Introduction

Maintenance covers all activities associated with the functioning of the entire factory assets. Due to the stochastic nature of failure incidents, this field is appealing to mathematical modelers, and optimization techniques are employed in a broad range in this regard (Wang, 2012). About 15% to 40% of production costs are assigned to maintenance activities. The purpose of management and planning of maintenance is to mitigate the undesired effects of breakdowns. Typically, two general approaches are available to reach these objectives: Preventive Maintenance (PM) and Corrective Maintenance (CM). While the former is about decreasing the probability of failure incidents during

*Corresponding Author
Email addresses: saffarian@birjandut.ac.ir(Mohsen Saffarian), kazemi_m_s@birjandut.ac.ir (Seyed Mahmood Kazemi), m.baharshahi@yahoo.com (Mohammad Baharshahi)

Received: November 2019 *Revised:* February 2020

the system's operation, the latter seeks to lower the intensity of undesired failure impacts after its happening. The recent studies suggest that the Preventive Maintenance is more reasonable for reaching the mentioned goals (Lo & Lofsten, 2000). PM problems concerns with determining the maintenance tasks to be done and their scheduling with respect to the limited resources available. In this regard, PM problem can be considered as a resource allocation and scheduling problem. In the literature, PM is categorized into two general classes:
(a) studies that focus on inspection periods and parts replacements, and (b) studies aiming at parts improvement (Gopalakrishnan et al., 2001). Scheduling the PM is an NP-hard problem. Factors including maximizing the system's reliability, minimizing the workforce, unemployment of workers, and imposed costs, turn it into a Multi-Objective Optimization Problem (MOP) problem (Quan et al., 2007). Prajapat et al. (2017) provided a review of the recent applications of PM optimization in power plants.
From the mathematical viewpoint, Maintenance Scheduling (MS) is a non-linear, multi-constraints, stochastic optimization problem. For simplicity, the linear form of MS problems is usually formulated as Mixed Integer Programming (MIP) (Sumesh et al., 2012). Numerous intelligent methods, including neural networks, simulated annealing, tabu search algorithm, ant colony, evolutionary techniques, etc. are employed to solve such problems (Schlunz et al., 2013). These approaches are adopted to find near-optimal solutions, the solution quality of which depends on the problem's nature. In some cases, an upper bound can be offered for the difference between the general optimal solution and the one obtained with the mentioned methods (Nicolai & Dekker, 2008).
PM seems necessary in power systems to prevent the generators' early deterioration and breakdown, which leads to outages and unpredicted costs. Maintenance of generators in power systems includes the scheduling and execution of field operations. Solving this scheduling program is of great essence, for it affects short- and long-term programs of power generation commitment in power plants. In this paper, the generator maintenance scheduling (GMS) problem is addressed. The larger the studied power system, the greater the demand for its electricity (load) generation, and consequently, the more complicated obtaining a proper scheduling program. Froger et al. (2016) offered a comprehensive literature review on maintenance schedulings in the power industry.
Given the combinatorial nature of the problem, the exact solution approaches cannot reach the optimal point in reasonable CPU times. Conversely, meta-heuristic algorithms require negligible CPU times for obtaining near-optimal solutions. Thus, in this paper, a discrete firefly algorithm is developed to find high-quality solutions in desirable CPU times.
The rest of the paper is organized as follows. In section 2, a brief literature review on GMS problem and its solution methodologies is provided. Problem formulation is presented in Section 3. The proposed meta-heuristic approach based on discrete firefly algorithm is delineated in detail in Section 4. Finally, numerical results and conclusions are discussed in Sections 5 and 6, respectively.

2. Literature review

According to the literature, objective functions can be categorized into three classes in terms of economic, reliability and convenience criteria. These criteria usually conflict with each other and cause the problem to become multi-objective, if considered simultaneously (Kralj & Petrovic, 1988). From the economic perspective, minimizing the startup, production, and maintenance costs are mainly considered.Benefit maximization is another economic goal interested in today's competitive market (Canto, 2008).Smoothing the reserved load throughout the planning horizon is at the focus of reliability criteria. This goal is modeled by minimizing the reserved load (deficiency-surplus) squares. Dahal & Chakpitak (2007) and Fung et al. (2007) implemented this class of functions as a singleobjective problem. Eventually, the cornerstone of the convenience criteria, which is less addressed in the literature, is minimizing the violation of constraints or schedule interruptions (Leou, 2006). Apart from the diversity of objective functions mentioned above, a variety of solution methodologies are also developed. Commercial softwares usually adopt exact methods such as branch and bound for this purpose.Likewise, many meta-heuristic methods, including genetic algorithm (GA) (Dahal et al., 2000), simulated annealing (SA) (Saraiva et al., 2011), tabu search (TS) (El-Amin et al., 2000), ant colony optimization (ACO) (Fetanat & Shafipour, 2011; Fug et al., 2007), and particle swarm optimization (PSO) (Sumesh et al., 2012) are implemented in practice. Hybrid meta-heuristic methods have also been developed (Reihani et al., 2012).

Wang and Handschin (2000) and Baskar et al. (2003) employed genetic algorithm to solve GMS problems. Yare and Venayagamoorthy (2010) and Ekpenyong et al. (2012) used PSO algorithm for scheduling maintenance problems in generators. Samuel and Asir Rajan (2015) proposed a hybrid evolutionary algorithm consisting of GA, PSO and frog leaping algorithm for long-term maintenance of generators.

Balaji et al. (2016) introduced a differential evolution algorithm mixed with Lambda iterative method to solve the GMS problem. Froger et al.(2018) used a large-scale neighborhood search algorithm based on constrained programming to solve a wind turbine maintenance and scheduling problem.

Bali and Labdelaoui (2015) formed an optimal approach for solving the GMS problem utilizing a hybrid algorithm based on a hybrid approach. Their objective function includes both reliability and violation of constraints. To solve this problem, a new optimization strategy based on SA and ACO was recommended. Kumarappan and Suresh (2015) proposed a hybrid algorithm of SA and PSO for a transmission constrained maintenance scheduling problem.

Lakshminarayanan and Kaur (2018) developed a cuckoo search optimization (CSO) algorithm for optimal maintenance scheduling of generator units. Umamaheswari et al. (2016) adopted a new meta-heuristic algorithm called the Ant Lion Optimizer (ALO) algorithm to solve the scheduling problem of generator preventive maintenance.

Abirami et al. (2015) considered a GMS problem aiming at maximizing reliability, and developed a nature-inspired algorithm named Teaching Learning Based Optimization (TLBO) to solve their problem. Hadjaissa et al. (2016) proposed a new competitive mechanism based on a modified genetic algorithm to optimize the power system scheduling.

Tamil Selvi et al. (2018) implemented and solved the GMS problem utilizing a stepwise and comprehensive genetic algorithm. They also estimated the performance of the genetic algorithm concerning time-saving and improvement in the quality of the solutions compared to the classical methods. Table 1 and Table 2 summarize the literature based on the above discussions.

Table 1: Types of objective Functions in GMS problem

Reference	Subject
(Canto, 2008)	Objective function with cost criteria
(Dahl and Chakpitak, 2007), (Foong et al., 2007), (Schlunz, 2011), Bali and Labdelaoui (2015), Abirami et al (2015)	Objective function with reliability criteria
(Leou, 2006)	Objective function with convenience criteria
(Kralj and Petrović, 1988)	Multi-objective

Table 2: The meta-heuristic algorithms used to solve GMS problem

Reference	Subject
Dahal et al. (2000), Wang & Handschin (2000) Baskar et al. (2003), Hadjaissa et al. (2016), Tamil Selvi et al. (2018)	Genetic Algorithm
Saraiva et al. (2011)	Simulated Annealing
El-Amin et al. (2000)	Tabu Search
Foong et. al. (2007), Fetanat & Shafipour (2011)	Ant colony
Sumesh et al. (2012), Yare & Venayagamoorthy (2010), Ekpenyong et al. (2012)	Particle Swarm
Balaji et al. (2016)	Differential Evolution Algorithm
Lakshminarayanan & Kaur (2018)	Cuckoo search
Umamaheswari et al. (2016)	Ant Lion Optimization
Abirami et al. (2015)	Learning-based optimization
Froger et al. (2018)	Neighborhood Search Algorithm
Reihani et al. (2012), Samuel & Asir Rajan (2015), Bali & Labdelaoui (2015), Kumarappan & Suresh (2015)	Hybrid Meta-heuristic Methods

3. Problem formulation

Depending on the complexity of the model, assumptions, and requirements, the GMS problem comprises different constraints. One of its simplest shapes is the problem modeling with constraints of the maintenance time window (earliest and latest times) and constraints of meeting the load requirements. Moreover, extra constraints can be added to the problem in case of need.

In GMS problem, there are several objective functions to optimize and several scheduling constraints to consider. In this paper, the reliability criterion for the objective function is selected. The goal is to smooth the reserved load during the planning horizon, which is formulated as the minimization of the sum of the squares of the reserved load (Schlunz, 2011). The reserved load is defined as the available production capacity minus the load demand. Problem constraints include maintenance intervals for each unit, meeting network demand and availability of maintenance personnel.

Suppose n is the number of generating units in the system, and m is the number of periods in the planning horizon. Let's take $I = \{1, \ldots, n\}$ and $J = \{1, \ldots, m\}$ for generators and time periods, respectively. The binary variable x_{ij} is 1 if maintenance task of ith generator starts in jth period; otherwise, it is 0. Also, the auxiliary variable y_{ij} equals 1 if the ith generator is under maintenance task in jth period; otherwise, it equals 0. The earliest and latest possible times to start repairing the ith generator are also indicated by the parameters e_i and l_i , respectively. Since there is only one time to repair during the above period, the following constraint is added to the model:

$$\sum_{j=e_i}^{l_i} x_{ij} = 1, \qquad i \in I \tag{1}$$

It is also known that a generator unit cannot be repaired outside the designated time period. Therefore, equalities (2) and (3) are enforced.

$$x_{ij} = 0 \quad j < e_i \; or \; j > l_i \qquad i \in I \tag{2}$$

$$y_{ij} = 0 \quad j < e_i \; or \; j > l_i + d_i - 1 \qquad i \in I \tag{3}$$

Where, d_i is the repair duration for ith generator. Equation (4) is also valid:

$$\sum_{j=e_i}^{l_i+d_i-1} y_{ij} = d_{ij}, \qquad i \in I \tag{4}$$

Since interruptions in maintenance tasks are not allowed and it must be done in an integrated scheme, constraints (5) and (6) are imposed.

$$y_{i,j} - y_{i,j-1} \leq x_{ij}, \qquad i \in I, \quad j \in J \setminus \{1\} \tag{5}$$

$$y_{i1} \leq x_{i1}, \qquad i \in I \tag{6}$$

The load demand also limits the maintenance schedule so that the total load demand during each time period must be met. g_{ij} is generating capacity of ith generator in jth period, and D_j is the load demand in the jth duration. We offer the reliability margin S as a percentage of demand as follows.

$$\sum_{i=1}^{n} g_{ij}(1 - y_{ij}) = D_j(1 + S) + r_j. \qquad j \in J \tag{7}$$

Where, r_j is the reserved level variable, which is defined as the portion of unused power generated in jth period (except for the reliability margin). Maintenance personnel constraints concerning the availability of manpower for maintenance operations should also be considered. $m'_{p,i,j}$ represents the workforce required to repair ith unit in jth period, when repairs are started from pth period. if m_i^k represents the manpower required to repair ith generator in kth period, then parameter $m'_{p,i,j}$ is calculated as follows.

$$m'_{p,i,j} = \begin{cases} m_i^{j-p+1} & if\ j - p < d_i \\ 0 & otherwise. \end{cases} \tag{8}$$

$$\sum_{i=1}^{n} \sum_{p=1}^{j} m'_{p,i,j} x_{ip} \leq M_j, \qquad j \in J \tag{9}$$

Where, M_j indicates the number for manpower available in jth period.
Sometimes repairing several units at the same time is not permitted. A typical case for this constraint can be considered as follows: If K sets of units are excepted from the constraint of simultaneous repairs, then we denote the set of all K sets with $K = \{1, \ldots, k\}$ and denote $I_k \subseteq I$ as the kth subset of the excepted units $(k \in K)$
According to the definition, the set of exceptions will be as follows:

$$\sum_{i \in l_k} y_{ij} x_{ip} \leq K_k, \qquad j \in J, \quad k \in K. \tag{10}$$

In the above formula K_k is the maximum number of units in the subset I_k that can be serviced concurrently.
Finally, the constraints that determine variable types are given in (11 – 13):

$$x_{ij} \in \{0, 1\}, \qquad i \in I, \quad j \in J. \tag{11}$$

$$y_{ij} \in \{0, 1\}, \qquad i \in I, \quad j \in J. \tag{12}$$

$$r_j \geq 0, \qquad j \in J. \tag{13}$$

The objective function is to minimize the sum of the squares of the reserved load, which is formulated as follows:

$$\min \sum_{j=1}^{m} (D_j S + r_j)^2 \tag{14}$$

The above modeling is a mixed integer quadratic program.

4. Solution methodology

Firefly algorithm (FA), derived from the behavior of fireflies in nature, was firstly introduced in the work of Yang (2008). According to the simulation results presented by Yang on many test problems, this algorithm showed higher efficiency and success rates compared to GA and PSO (Yang & Press, 2010). Since this algorithm works well in continuous optimization problems, it can be expected to perform well in discrete optimization problems too. Consequently, after introducing the general form of this algorithm, its discrete version is also tested. In the following, we address some of the parameters and variables applied in the FA algorithm:

Attractiveness. The attractiveness of any firefly is a relative value that is determined by distance. The general structure of the attraction function by distance $\beta(r)$ is as follows:

$$\beta(r) = \beta_0 e^{-\gamma r^m} \qquad m \geq 1 \tag{15}$$

In the above function r is the distance between two fireflies, β_0 is the attraction rate at $r = 0$, and γ is the constant coefficient of light attraction by the environment.

Distance. The distance between two fireflies i and j, which are in coordinates X_i and X_j , is computed using equation (16).

$$r_{ij} = \|X_i - X_j\| = \sqrt{\sum_{k=1}^{d} (x_{ik} - x_{jk})^2} \tag{16}$$

where, x_{ik} is kth component of X_i vector.

Movement. The movement of ith firefly toward jth more luminous firefly is calculated by equation (17):

$$X_i = X_i + \beta_0 e^{-\gamma r_{ij}^2} (X_i - X_j) + \alpha(rand - \frac{1}{2}) \tag{17}$$

The third sentence in the above relation leads to divergence. α is a random parameter and rand is a random number uniformly distributed in $[0, 1]$.

4.1. Discrete Firefly Optimization Algorithm

To solve the maintenance scheduling problem addressed in the previous section, the firefly algorithm, which is of a continuous nature, needs to be manipulated for problems with integer variables.

A discrete firefly algorithm (DFA) has been developed by Jati and Manurung (2013) to solve a traveling salesman problem with Hamming's distance instead of Euclidean distance. Hamming's distance for two series of numbers (two different solutions) is determined by the number of mismatches of the solution components. Also, in 2008, a modified discrete particle swarm algorithm was adopted to solve GMS problem, in which it rounds the new particle location (which is not necessarily an integer vector) to the closest integer after calculating the new particle. Since the firefly algorithm and the particle swarm have fundamental similarities, the same approach has been adopted in this project to implement the firefly algorithm in the GMS problem (Aliyu, 2008). In this case, the equation related to the movement of firefly is as follows:

$$X_i = X_i + Round(\beta_0 e^{-\gamma r_{ij}^2}(X_i - X_j) + \alpha(rand - \frac{1}{2})) \tag{18}$$

In equation (18), the new location vector (on the left side of the equation) is computed by sum of two integers on the right side. Other parts of the algorithm remain unchanged.

4.2. Firefly Algorithm implementation

Firstly, an initial population is generated. Then, over several steps, the solution space is searched, and the algorithm is led to better desirable solutions. The way to apply this algorithm to the problem is described in the following.

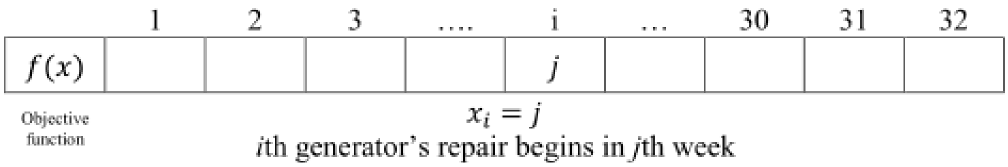
4.2.1. Pseudocode

The discrete firefly pseudocode for problem-solving is presented below.

Define objective function $f(x)$ where $x = (x_1, \ldots, x_d)^T$
Generate initialize a population of n fireflies x_i $(i = 1, 2, \ldots, n)$ by heuristic method
Light intensify l_i at x_i is determined by $f(x_i)$
Define light absorbtion coefficient γ
While (t < MaxGeneration)
    choose P percent of population
        For $i = 1 : n \times p(n \times \%$ P fireflies)
            For $j = 1 : i$
                If $(l_j > l_i)$
                    Move firefly i towards j in d-dimension by using movement equation
                End if
                Attractiveness varies with Euclidian distancer via $\exp(-\gamma r^2)$
                Evaluate new solutions and update light intensify
            End for j
        End for i
        Choose remain $(100 - P)$ Percent of population
        For $i = 1 : n \times (1 - P)(n \times \%(1 - P)$ firefly)
            If $(l_j > l_i)$
                Move firefly i towards j in d-dimension by using heuristic methods
            End if
            Attractiveness varies with Euclidian distance r
            Evaluate new solutions and update light intensify
        End for i
        Rank the fireflies and find the current best
End While
Post process result and visualization

4.2.2. Solution representation

In this algorithm, a solution is represented by a vector of length 32 + 1; The first element indicates the objective function value while the subsequent elements show the maintenance time of each generator as an integer number.



4.2.3. The main part

The algorithm's input data are taken from an Excel file which contains three input matrices. The first matrix contains the production amount, maintenance time window (earliest and longest possible), length of maintenance period, and number of workers required for 32 generators. The second matrix includes data related to the constraints of simultaneous maintenance of generators. The third matrix gives the demand for energy over 52 weeks. Afterwards, the parameters of the problem, including population size, light absorption coefficient (γ) , random motion parameter, and the number of generations to be formed, are defined.
In the next step, the solution-making function is called to construct the initial population. The population is firstly sorted by desirability and is divided into two sections, subsequently. The first 20% of the population are utilized by the discrete firefly algorithm to search the solution space. The remaining 80% is broken into four equal sections on each of which a specific scenario is implemented. These scenarios are innovatively developed by the authors and are as follows:

(A) Comparing the offspring with the best solution produced in all past generations and moving toward it through averaging maintenance times
(B) Choosing a random number from the population, selecting two generators at random and swapping their maintenance times
(C) Shifting generators' maintenance times randomly within the time window
(D) Replacing the least desirable solutions with new randomly generated solutions.

After the new generation is produced, the solution-correcting function is called to correct some of the violated constraints and determine the objective function value of the solution. In each generation, by recalling the solution-making function, the duplicated solutions are replaced by new ones and the entire population is sorted by the objective function. Finally, the best results are stored in each generation, and the next generation is created.

4.2.4. Solution-making function

This function has two inputs, including the first and the third matrices defined previously. Each calling of this function generates an initial solution such that model constraints are maintained except for limitations on simultaneous repair of generators. Additionally, a maximum of five weeks out of 52 weeks is permitted to violate the positive constraint of the reserved amount. That is, every time this function is executed, a solution is generated for which a maximum of 12 constraints out of 175 constraints can be violated. This number is chosen so that a good balance can be established between the speed of solution production and its quality.

4.2.5. Solution-correcting function

This function modifies a part of the selected solutions and entails four main inputs: three problem data matrices and one solution vector. The output of this function is a response vector balanced by the value of the given function. By this function, the violated constraints in the solution are identified and counted. Some of these constraints, which relate to the maximum available workers, are corrected. For other violations, proportionate to their number, a penalty is imposed on the objective function. This approach is adopted because correcting the solution vector with respect to all 175 constraints is exhaustive and time-consuming.
The new form of the objective function by adding the mentioned penalties is as follows:

$$f(x) = f(x) + w_l P_l + w_e P_e$$

P_l and P_e are respectively the violated constraints of the reserved load, and the constraint of concurrent repair prohibition, for which weights w_l and w_e are considered proportionally.

4.3. An example

The algorithm was implemented 20 times by setting the parameter values as the following:
Generation production number: 50
Number of fireflies: 20
Light absorption coefficient (γ) : 0.001
Random Motion Parameter (α) : 1
Here is an example answer to the problem:

Table 3: Generator Maintenance Problem Sample

Generator No.	1	2	3	4	5	6	7	8	9	10	11
Starting week for maintenance	10	6	13	27	8	27	23	49	3	46	24
Generator No.	12	13	14	15	16	17	18	19	20	21	22
Starting week for maintenance	15	11	34	10	10	2	1	17	20	31	1
Generator No.	23	24	25	26	27	28	29	30	31	32	
Starting week for maintenance	45	29	32	51	40	18	35	19	42	37	

The objective function for the above answer is $3.4510^7 MW^2$, which is obtained in 172 seconds which is a satisfactory performance compared to the literature.

4.4. Combining with heuristic local search

Heuristic neighborhood production structures have also been employed to produce more diverse solutions. To this end, the solution population is subdivided into subgroups. In each of these subgroups, a variety of heuristics are used to search for local solutions. Comparison of the produced solution with the best one and moving toward it, pair replacements, and shifting repair times are some of these heuristic methods utilized in the local search.

5. Computational results

The data of a benchmark problem was taken from the literature to examine the performance of the proposed algorithm. Parameters were adjusted using a trial and error approach. Parameter of light absorption coefficient (γ) was set to 0.001. α was chosen to be 3 according to the response vector scale. Noteworthy, the firefly algorithm showed insignificant sensitivity regarding its parameters' values and its performance was only affected by remarkable changes in parameters' values. Running this algorithm on several test problems indicated that the algorithm has a high convergence rate at the initial steps and obtains desirable solutions.

5.1. Test problem

The 32-unit GMS problem is originated from the IEEE 1979 Reliability Testing System, which intends to minimize the reserved load over the planning horizon by considering maintenance time window constraints, satisfying demand reliably, labor constraints, and a set of concurrent repair prohibition constraints. The theoretical lower bound for this problem is $33363252 MW^2$.
The developed discrete firefly algorithm was coded in MATLAB version 7.8.0 ($R2009a$) and run on a computer with an Intel Pentium $P6200$ 2.13GHz processor with 2GB of RAM and Windows 7 Ultimate 64-Bit as operating system.
In Fig.1, the minimum and average values of the objective function obtained from the implementation of the discrete firefly algorithm were compared with four simulated annealing temperature reduction functions (Geo-Huang-VanL-Triki). The best solution obtained by the discrete firefly algorithm was 34509254 MW^2, which is only 3.43% higher than the theoretical lower bound.

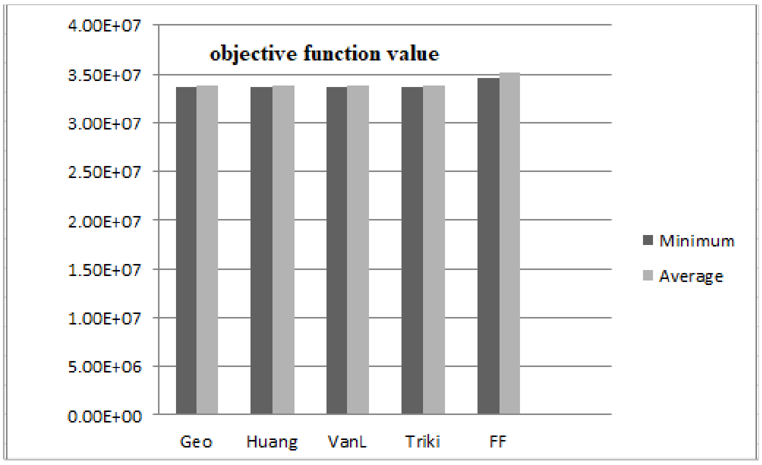


Figure 1: Discrete firefly algorithm versus the four simulated annealing temperature reduction functions

In Table 4 the mean and standard deviation of computing times in the discrete firefly algorithm is compared with the four simulated annealing temperature reduction functions. It can be clearly seen that DFA outperforms Triki on both measures. Moreover, since the computation capabilities of the Computer used in this paper was weaker than that of the benchmark paper, it can be claimed that its performance was also acceptable regarding other SA settings.

Table 4: Comparison of the solution time of the discrete firefly algorithm and the simulated annealing

	SA				DFA
	Geo	Huang	VanL	Triki	
Average (seconds)	78.95	13.78	119.89	286.01	158.83
Standard deviation	3.49	1.19	6.74	1008.9	22.12

Table 5 compares the performance of a discrete firefly algorithm with the four simulated annealing temperature reduction functions. After 50 times of the algorithm execution, the number of solutions was improved, and the average of the solutions was obtained every time the algorithm was executed and its maximum improvement was obtained. Since the discrete firefly algorithm in the early steps usually yields good results, the number of improved solutions in the 50 iterations of the algorithm was smaller than that of the simulated annealing.

Table 5: Comparison of the performance of the discrete firefly algorithm and the simulated annealing

	SA				DFA
	Geo	Huang	VanL	Triki	
number of improved solutions in 50 iterations	27	28	41	29	6
Average improvement (%)	0.03	0.05	0.03	0.9	0.06
Maximum improvement (%)	0.22	0.42	0.18	0.75	0.11

6. Conclusions

In this project, a discrete firefly algorithm was executed on a generator maintenance problem, with the aim of minimizing the square of the reserved load over the planning horizon by considering the time window of maintenance, meeting the demand with reliability margin, labor constraints, and the set of constraints on concurrent repairs. The results of this implementation revealed that the firefly algorithm was able to obtain suitable solutions for this problem at acceptable CPU times.

Moreover, by embedding multiple heuristic structures in the DFA to search more efficiently in the solution space, its convergence performance was enhanced significantly.

References

[1] Abirami, M., R. Anandhakumar, S. Ganesan and S. Subramanian (2015). ”Reliability based source maintenance scheduling using nature inspired algorithm.”
International Journal of Quality & Reliability Management 32(1): 81-96.

[2] Aliyu, Y. Y. G. K. V. U. O. (2008), “Optimal generator maintenance scheduling using a modified discrete PSO,” vol. 2, no. 6, pp. 834–846.

[3] Balaji, G., R. Balamurugan and L. Lakshminarasimman (2016). ”Mathematical approach assisted differential evolution for generator maintenance scheduling.”
International Journal of Electrical Power & Energy Systems 82: 508-518.

[4] Bali, N. and H. Labdelaoui (2015). ”Optimal generator maintenance scheduling using a hybrid metaheuristic approach.” International Journal of Computational Intelligence and Applications 14(02): 1550011.

[5] Baskar, S., P. Subbaraj, M. V. C. Rao and S. Tamilselvi (2003).
”Genetic algorithms solution to generator maintenance scheduling with modified genetic operators.”
IEE Proceedings - Generation, Transmission and Distribution 150(1): 56-60.

[6] Canto, S. P. (2008), “Application of Benders' decomposition to power plant preventive maintenance scheduling,” Eur. J. Oper. Res., vol. 184, no. 2, pp. 759–777.

[7] Dahal, K. P., McDonald, J. R.and Burt, G. M. (2000), “Modern heuristic techniques for scheduling generator maintenance in power systems,” Trans. Inst. Meas. Control, vol. 22, no. 2, pp. 179–194.

[8] Dahal, K. P.and Chakpitak N. (2007),
“Generator maintenance scheduling in power systems using metaheuristic-based hybrid approaches,” Electr. Power Syst. Res., vol. 77, no. 7, pp. 771–779.

[9] Ekpenyong, U. E., J. Zhang and X. Xia (2012). ”An improved robust model for generator maintenance scheduling.” Electric Power Systems Research 92: 29-36.

[10] El-Amin, I., Duffuaa, S. and Abbas, M. (2000), “A Tabu search algorithm for maintenance scheduling of generating units,” Electr. Power Syst. Res., vol. 54, no. 2, pp. 91–99.

[11] Fetanat, A. and Shafipour, G. (2011), “Generation maintenance scheduling in power systems using ant colony optimization for continuous domains based 0–1 integer programming,”
Expert Syst. Appl., vol. 38, no. 8, pp. 9729–9735.

[12] Foong, W. K., Simpson, A. R., Maier, H. R. and Stolp, S. (2007), “Ant colony optimization for power plant maintenance scheduling optimization—a five-station hydropower system,”
Ann. Oper. Res., vol. 159, no. 1, pp. 433–450.

[13] Froger, A., M. Gendreau, J. E. Mendoza, É. Pinson and L.-M. Rousseau (2016). ”Maintenance scheduling in the electricity industry: A literature review.” European Journal of Operational Research 251(3): 695-706.

[14] Froger, A., M. Gendreau, J. E. Mendoza, E. Pinson and L.-M. Rousseau (2018).
”Solving a wind turbine maintenance scheduling problem.” Journal of Scheduling 21(1): 53-76.

[15] Gopalakrishnan, M., Mohan, S. and He, Z. (2001),
“A tabu search heuristic for preventive maintenance scheduling,” Comput. Ind. Eng., vol. 40, no. 1–2, pp. 149–160.

[16] Hadjaissa, B., K. Ameur, S. M. Ait cheikh and N. Essounbouli (2016). ”Bi-objective optimization of maintenance scheduling for power systems.”
The International Journal of Advanced Manufacturing Technology 85(5): 1361-1372.

[17] Jati, G. K. and Manurung, R. (2013) ”13 Discrete Firefly Algorithm for Traveling Salesman Problem : A New Movement Scheme”. Elsevier Inc., pp. 295–312.

[18] Kralj, B. L. and Petrović, R. (1988), “Optimal preventive maintenance scheduling of thermal generating units in power systems —A survey of problem formulations and solution methods,”
Eur. J. Oper. Res., vol. 35, no. 1, pp. 1–15.

[19] Kumarappan, N. and K. Suresh (2015). ”Combined SA PSO method for transmission constrained maintenance scheduling using levelized risk method.”
International Journal of Electrical Power & Energy Systems 73: 1025-1034.

[20] Lakshminarayanan, S. and D. Kaur (2018). ”Optimal maintenance scheduling of generator units using discrete integer cuckoo search optimization algorithm.” Swarm and Evolutionary Computation 42: 89-98.

[21] Leou, R. C. (2006), “A new method for unit maintenance scheduling considering reliability and operation expense,” Int. J. Electr. Power Energy Syst., vol. 28, no. 7, pp. 471–481.

[22] Lo, H. and Löfsten, H. (2000), "Measuring maintenance performance – in search for a maintenance productivity index," Int. J. Prod. Econ., vol. 63, no. 1, pp. 47–58.
[23] Nicolai, R. R. and Dekker, R. (2008), "Optimal Maintenance of Multi-component Systems: A Review," in in Complex System Maintenance Handbook SE - 11, Springer London, pp. 263–286.
[24] Prajapat, N., A. Tiwari, X.-P. Gan, N. Z. Ince and W. Hutabarat (2017). Preventive Maintenance Scheduling Optimization: A Review of Applications for Power Plants. Advances in Through-life Engineering Services. L. Redding, R. Roy and A. Shaw. Cham, Springer International Publishing: 397-415.
[25] Quan, G., Greenwood, G. W., Liu, D. and Hu, S. (2007), "Searching for multiobjective preventive maintenance schedules: Combining preferences with evolutionary algorithms," Eur. J. Oper. Res., vol. 177, no. 3, pp. 1969–1984.
[26] Reihani, E., Sarikhani, A., and Davodi, M. (2012),"Reliability based generator maintenance scheduling using hybrid evolutionary approach," Int. J. Electr. Power Energy Syst., vol. 42, no. 1, pp. 434–439.
[27] Samuel, G. and Asir Rajan, C. (2015). "Hybrid: Particle Swarm Optimization–Genetic Algorithm and Particle Swarm Optimization–Shuffled Frog Leaping Algorithm for long-term generator maintenance scheduling." International Journal of Electrical Power & Energy Systems 65: 432-442.
[28] Saraiva, J. T., Pereira, M. L., Mendes, V. T. and Sousa, J. C. (2011), "A Simulated Annealing based approach to solve the generator maintenance scheduling problem," Electr. Power Syst. Res., vol. 81, no. 7, pp. 1283–1291.
[29] Schlunz, E. (2011), "Decision support for generator maintenance scheduling in the energy sector,".
[30] Schlünz, E. B. B. and van Vuuren, J. H. H. (2013),"An investigation into the effectiveness of simulated annealing as a solution approach for the generator maintenance scheduling problem," Int. J. Electr. Power Energy Syst., vol. 53, no. 0, pp. 166–174.
[31] Sumesh, E. P., Suresh, K. and Kumarappan, N. (2012), "Particle swarm optimization based generation maintenance scheduling using probabilistic approach," Procedia Eng., vol. 30, pp. 1146–1154.
[32] Tamil Selvi, S., S. Baskar and S. Rajasekar (2018). Chapter 5 - An Intelligent Approach Based on Metaheuristic for Generator Maintenance Scheduling. Classical and Recent Aspects of Power System Optimization. A. F. Zobaa, S. H. E. Abdel Aleem and A. Y. Abdelaziz, Academic Press: 99-136.
[33] Umamaheswari, E., S. Ganesan, M. Abirami and S. Subramanian (2016). Deterministic reliability model based preventive generator maintenance scheduling using Ant Lion Optimizer. 2016 International Conference on Circuit, Power and Computing Technologies (ICCPCT).
[34] Wang, W.(2012), "An overview of the recent advances in delay-time-based maintenance modelling," Reliab. Eng. Syst. Saf., vol. 106, pp. 165–178.
[35] Wang, Y. and E. Handschin (2000). "A new genetic algorithm for preventive unit maintenance scheduling of power systems." International Journal of Electrical Power & Energy Systems 22(5): 343-348.
[36] Yang, X. S. (2008), Nature-Inspired Metaheuristic Algorithms. Luniver Press. ISBN 978-1-905986-10-1.
[37] Yang, X. and Press, L. (2010), Nature-Inspired Metaheuristic Algorithms Second Edition. pp. 81–89.
[38] Yare, Y. and G. K. Venayagamoorthy (2010). "Optimal maintenance scheduling of generators using multiple swarms-MDPSO framework." Engineering Applications of Artificial Intelligence 23(6): 895-910.