IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Mathematical Modelling of Engineering Problems
Vol. 11, No. 10, October, 2024, pp. 2595-2605
Journal homepage: http://iieta.org/journals/mmep

Optimal Design of Helical Springs Using Metaheuristic Techniques

Miguel Rodriguez Cabal[1*], Luis Grisales Noreña[2], Oscar Danilo Montoya[3], Diego Hincapie[1], Brandon Cortes Caicedo[4]

[1] Department of Mechatronics and Electromechanics, Instituto Tecnologico Metropolitano, Medellín 050036, Colombia
[2] Department of Electrical Engineering, Faculty of Engineering, Universidad de Talca, Curicó 3340000, Chile
[3] Facultad de Ingeniería, Universidad Distrital Francisco José de Caldas, Bogotá 110231, Colombia
[4] Facultad de Ingeniería, Institución Universitaria Pascual Bravo, Medellín 050036, Colombia

Corresponding Author Email: miguelrodriguez@itm.edu.co

https://doi.org/10.18280/mmep.111001

Received: 18 July 2024
Revised: 8 September 2024
Accepted: 14 September 2024
Available online: 31 October 2024

Keywords:
mechanical design, optimization, finite element analysis, mathematical modelling

ABSTRACT

The manufacturing industry's energy demands and the need for continuous improvement in compact, lightweight machines with mechanical resistance requirements require improvements in design. The objective is to obtain designs that occupy a smaller volume and have acceptable physical characteristics for the required function. This is achieved by implementing optimization techniques that yield the minimum value of an objective function constrained by the physical requirements of the problem. This work proposes the use of metaheuristic optimization techniques to find the dimensions that determine the smallest volume of a closed coil helical spring, which is found in industrial and automotive applications, respecting the physical and technical criteria that describe the design of these elements. The results demonstrate that the VSA identifies the optimal solution to the problem in a significantly shorter processing time than other techniques, with a volume reduction of 0.1% to 0.54%. The Inventor® software is used as a validation method of the solution, where a finite element analysis is carried out to verify that the mathematical model proposed adequately represents the problem and complies with the restrictions described in the mathematical model. Optimization algorithms together with simulation techniques are a useful tool in the design of machine elements, as they reduce computation and processing times, allowing solutions to be found that minimize production costs without neglecting the technical constraints of each problem.

1. INTRODUCTION

The design of machine elements is essential in the industry in general because, in addition to safe and reliable, these elements should be manufactured at a low cost and designed in a short time. This improves process efficiency (because less material is used) and guarantees that the final design of the element meets physical and technical requirements to work properly in the machine where it is installed. Many studies have investigated the minimization of production costs, weights, stresses, volume, and other aspects [1, 2]. Some of them have explored simulation techniques such as finite element analysis (FEA), which uses computer aided software engineering to simulate several mechanical elements in order to analyze the effects of their loads [3, 4] and the alteration that they might experience due to changes in the material or the shapes of the design [5, 6].

Such is the case presented by Yetgin et al. [7], where a failure analysis was carried out on a premature crack formation that occur in a helical spring. Authors employed nondestructive tests and simulations to determine the geometrical factors that have influence in the performance of the element in order to make the corrections in the design and avoid future failure product of the stresses presented. This case is a classical analytical method used to improve the mechanical properties of machine elements based on a real base case of study, where it is necessary to have the testing machines. In automotive industry helical springs are widely used and studied via simulation, as presented by Arslan and Genel [8], where a coil spring fractured is analyzed, authors used microscopy studies and FEA to determine the stress distribution and make a correction in the design. In this work authors found that the main cause of the failure was attained to the painting process which change the microstructure of the spring as a result of the simulation analysis carried out in ABAQUS software. To reduce the weight of springs, specialized literature includes studies on various composite materials. These studies employ finite element analysis to evaluate the stresses and deformations that these materials may experience, with the objective of optimizing the design geometry. In the work presented by Ke et al. [9], an experimental design is developed for a fatigue resistant helical spring. The geometry is initially analyzed using Abaqus software to characterize the material properties, followed by mechanical testing to validate the hypothesis. The authors describe the methodology for designing springs using

composite materials.

Simulation techniques provide data on the stresses and deformations that machine elements might suffer, as well as their lifespan and where failures that prevent their proper operation might occur. However, to run the simulations, the dimensions of the elements under analysis should be known. These dimensions are usually calculated based on the design theory established mostly by Norton [10] and Mott [11], according to which elements are modeled using a manual iterative process and considerations that depend on the designer's expertise. This results in a slow process that presents failures due to human error [12].

For this reason, optimization algorithms have been recently implemented in mechanical design. The purpose of these algorithms is to minimize or maximize an objective function to find the best values of the variables that solve the problem under analysis [13]. This is possible by using models of constraints that represent the physical and technical aspects of the design to ensure that the dimensions provided by the algorithms fulfill their purpose [14]. As the specialized literature includes multiple solution techniques, designers should explore the types of algorithms that best fit the mathematical models of their mechanical designs [15]. Since the main equations of the mathematical models that represent machine elements are nonlinear and non-convex, optimization algorithms are used to solve this type of problems. Currently, the solution methods most reported in the literature are metaheuristic optimization techniques because they focus on the exploration and exploitation of the solution space using different moving strategies (that prevent them from falling into local optima within the solution space) in order to obtain high quality solutions for the machine or part design [16-18].

In the specialized literature, various studies implement metaheuristic algorithms to solve optimization problems applied to mechanical design are reported. Among these is the work presented by Rodriguez-Cabal et al. [19], where the optimization of the weight of a drive shaft is achieved through the implementation of a Vortex Search Algorithm. This approach utilizes a nonlinear, non-convex mathematical model to properly size the shaft, with validation performed using simulation software. Another case is presented by Parouha and Tiwari [20], where two engineering problems, characterized by nonlinear mathematical models, are optimized by implementing an improved version of the Particle Swarm Optimization (PSO) algorithm. In this article, the authors demonstrate that, despite being a classical technique, this algorithm continues to yield good results for engineering optimization problems. Within the family of population-based algorithms, genetic algorithms are also found, which have been used to optimize complex mathematical models [21]. Additionally, there are studies where various algorithms are implemented to solve engineering problems using population-based algorithms [22, 23].

Helical springs are machine elements that consist of a wire curved in a cylindrical shape with a constant pitch. The cross-section of helical springs determines if they are cylindrical or square [24]. Helical springs can also be open coil or closed coil depending on the angle between the coil and the vertical axis. Open coil springs have a longer pitch and therefore fewer turns in the same length [25]. Helical springs have multiple applications: absorbing shocks, applying forces to brakes and clutches, controlling movements when they act as stabilizers, and storing energy in toys, clocks, etc. [26]. Therefore, the operation and safety of equipment that incorporates helical springs depend on the appropriate sizing of these elements.

Based on the review of specialized literature, it is found that simulation techniques are often separate from optimization techniques. The former is typically chosen for performing parametric simulations aimed at finding suitable dimensions for helical springs, leading to solutions that meet the requirements but may not be optimal in terms of volume or weight. This is why optimization techniques emerge as a tool that should be included in the design process, as they can reduce the computational cost of simulations by providing a design that meets the technical requirements established in design manuals. It is for the above that this paper proposes a novel helical spring design that minimizes the volume of this element by considering a set of constraints imposed by technical and physical criteria. Four solution methods were used here to optimize the design: Particle Swarm Optimization (PSO), the Continuous Genetic Algorithm (CGA), the Sine Cosine Algorithm (SCA), and the Vortex Search Algorithm (VSA). The latter has been employed in the specialized literature to solve this type of problems [17, 19, 27]. The selection of the solution algorithms is based on the literature review where these methods have been shown as an effective tool to optimize nonlinear non-convex mathematical models as the one who represent the design of helical springs.

The final result of this study is an optimal helical spring design that guarantees that the technical constraints associated with the element under analysis are respected. Inventor® software implementing FEA was used to simulate and validate the optimization results.

The objective of this work is to evaluate different optimization techniques for the problem at hand, while also demonstrating the validity of both the mathematical model and the optimization technique through simulation. This will establish a proper design procedure that will help reduce design time and computational cost when performing the correct sizing of such elements.

The following section outlines the structure of the remaining document. Section 1 presents the mathematical modelling to optimize the design of a closed coil helical spring. Section 2 provides a detailed account of the optimization techniques that have been implemented, accompanied by the relevant equations or parameters that are to be employed in order to solve the problem. Section 3 presents the results of the algorithms and its analysis and discussion. Subsequently, Section 4 demonstrates the manner in which the results were validated using Inventor Autodesk software. Finally, Section 5 presents the findings and proposes avenues for future research.

2. MATHEMATICAL MODEL

To design a helical spring, the physical characteristics of this element should be considered. The stresses that the spring will withstand are evaluated based on its geometry. The mathematical model of a helical spring is established by defining the spring volume based on wire diameter (d), mean coil diameter (D), and number of active coils (N_c) (see Eq. (1)). Figure 1 shows the variables in Eq. (1), which is the objective function.

$$V = \left(\frac{\pi}{2}\right)^2 (N_c + 2) D d^2 \tag{1}$$

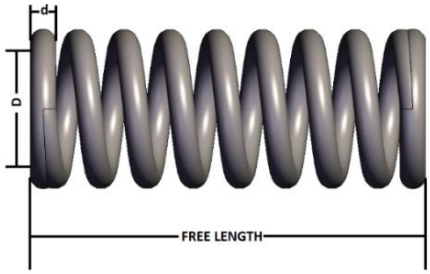

Figure 1. Identification of the variables considered in the helical spring design

2.1 Set of constraints

The geometrical parameters that define the volume in Eq. (1) satisfy the physical characteristics of the spring, i.e., it is able to withstand the load that it will be subjected to without exceeding the maximum allowable stresses and the geometrical values given by the available space and technical conditions of the design. The stress produced by the maximum applied force may not exceed the maximum allowable stress, which defines the first constraint (g_1). Eq. (2) establishes the relationship between applied stress, maximum applied force, and the geometrical parameters of the spring.

$$g_1 = \pi d^3 S - 8 C_f F_{max} D \geq 0 \tag{2}$$

where, F_{max} is the maximum applied force and C_f is the geometric parameter defined by Eq. (3).

$$C_f = \frac{4C^2 + 1.46C - 2.46}{4C(C-1)}, C = \frac{D}{d} \tag{3}$$

Note that the free length (l_f) of the spring should be shorter than the maximum length available in the space in which it will be installed, which defines the second constraint (see Eq. (4)).

$$g_2 = l_{max} - l_f \geq 0 \tag{4}$$

It is assumed that the free length is α times the solid length (see Eq. (5)).

$$l_f = \delta_l + \alpha(N_c + 2)d \tag{5}$$

where, δ_l denotes the deflection under maximum load, which is calculated using Eq. (6).

$$\delta_l = \frac{F_{max}}{K} \tag{6}$$

where, K is the spring constant, which is expressed in Eq. (7), where G is the shear modulus.

$$K = \frac{Gd^4}{8N_c D^3} \tag{7}$$

The third constraint establishes that the diameter of the spring wire should not be smaller than the specified minimum value, d_{min}, which is expressed in Eq. (8).

$$g_3 = d - d_{min} \geq 0 \tag{8}$$

In addition, the outer diameter of the coil should not surpass the specified maximum value (fourth constraint), which is expressed in Eq. (9).

$$g_4 = D_{max} - D - d \geq 0 \tag{9}$$

In order to prevent the spring from becoming too tight, it is necessary to ensure that the coil diameter is at least β times the spring diameter. This constraint is formulated in Eq. (10).

$$g_5 = C - \beta \geq 0 \tag{10}$$

Under load, the deflection of the spring should be less than a specific value. This constraint is expressed in Eq. (11).

$$\delta_p = \frac{F_p}{k} \geq 0 \tag{11}$$

This leads to the sixth constraint:

$$g_6 = \delta_{pmax} - \delta_p \geq 0 \tag{12}$$

The seventh constraint establishes that the combined deflection should be consistent with the length. This is formulated in Eq. (13).

$$g_7 = l_f - d_n \geq 0 \tag{13}$$

Finally, the preload deflection under maximum load should be equivalent to the value specified in the design, which is defined in Eq. (14), where δ_w is a constant value.

$$g_8 = F_{max} - F_P - L\delta_w \geq 0 \tag{14}$$

After these constraints have been presented and interpreted, below are Eqs. (15) to (22), which compose the set of penalties to be used in Eq. (31). Eqs. (15) to (22) are associated with each one of the previously mentioned constraints and defined by the maximum value function. When each constraint is analyzed, it takes the maximum value of the function. It takes g if the constraint is violated or 0 if it is respected.

$$p_1 = \max\{0, g_1\} \tag{15}$$

$$p_2 = \max\{0, g_2\} \tag{16}$$

$$p_3 = \max\{0, g_3\} \tag{17}$$

$$p_4 = \max\{0, g_4\} \tag{18}$$

$$p_5 = \max\{0, g_5\} \tag{19}$$

$$p_6 = \max\{0, g_6\} \tag{20}$$

$$p_7 = \max\{0, g_7\} \tag{21}$$

$$p_8 = \max\{0, g_8\} \tag{22}$$

3. SOLUTION METHODS

Four optimization algorithms are proposed here to solve the problem of minimizing a helical spring's volume. These algorithms are selected based on a review of specialized

literature, as they are used to solve optimization problems related to the minimization of non-linear, non-convex mathematical models such as the one addressed here. Additionally, some particularities of each algorithm are mentioned, which contributed to the selection of these specific methods, as detailed below. The first method, i.e., Particle Swarm Optimization (PSO), was implemented in this study because it has been used in the literature to solve nonlinear mathematical models, highlighting it as one of the most widely used classical optimization technique in the literature [28-33]. The Continuous Genetic Algorithm (CGA) and the Sine Cosine Algorithm (SCA) were also employed here because of their short processing times [34-38]. Finally, the Vortex Search Algorithm (VSA), which has been implemented in mechanical design problems [17, 19], was adopted here because of its short processing time and the repeatability of its solutions. The following subsections provide an overall explanation of these four techniques.

3.1 Particle Swarm Optimization

PSO is a type of bioinspired algorithm based on the behavior of bird flocks when they forage [39, 40]. The algorithm starts with the generation of a population within an n-dimensional space that directly depends on the number of variables that are associated with the objective function. Then, the function and the set of constraints are evaluated to find the individual with the best position. Based on this position, a social and cognitive component and a velocity are introduced. This allows the algorithm to move randomly around the point with the best previous position. At each iteration, the objective function and the set of constraints are evaluated, the particle with the best position is updated, and, after exploring and exploiting the solution space, the algorithm converges to an optimal solution. Algorithm 1 describes the programming of PSO.

Algorithm 1. PSO pseudocode

```
Data: PSO Parameters, Parameters of the mathematical model.
for t = 1: t_max do
    if t = 1 then
        Generate the particle swarm;
        Evaluate the objective function and the constraints for each individual;
        Select the best solution with its position;
    else
        Update the velocity vector;
        Update the position of the particles;
        Generate the particle swarm;
        Evaluate the objective function and the constraints for each individual;
        Evaluate the fitness function;
        Update the best solution with its position;
        if t = t_max then
            Result: Print results
            Break
        else
            Continue
        end
    end
end
```

3.2 Continuous genetic algorithm

This algorithm is based on a classical optimization technique that starts with an initial population organized in an $a \times s$ matrix, where a denotes the number of individuals or potential solutions, and s is the number of variables associated with the problem. After the creation of the initial population, the objective function is evaluated. Importantly, the CGA solves optimization problems by turning to a conditional problem a constrained problem. Therefore, at each iteration, the set of constraints is analyzed, and, if a solution violates any of the conditions proposed in the mathematical model, the algorithm penalizes the value of its objective function and said solution is considered inviable.

After the evaluation of the initial population, the descendant population is created. In this step, new potential solutions are generated so that the algorithm moves over the solution space. Mutation and crossover operators are adopted to generate this new set of individuals. Besides, classical selection is implemented so that the best solutions remain in the set of individuals to improve their position.

Finally, applying methods of combination, selection and mutation, the descendant populations are generated. This method allows to ensure that the potential solutions are different. The resulting list of potential solutions is sorted in ascending order based on the objective function of all the individuals, and the main potential solutions are selected as the new population to move to the next iterative cycle. The algorithm ends the process when the convergence criterion is achieved, or the total number of iterations is reached. Algorithm 2 shows the pseudocode that describes the programming of the CGA.

Algorithm 2. CGA pseudocode

```
Data: CGA Parameters, Parameters of the mathematical model.
for t = 1: t_max do
    m = 0;
    if t = 1 then
        Generate the initial population;
        for i = 1: a do
            Evaluate the objective function and the constraints for each individual;
            Select the best solution with its position;
        end
    else
        Generate the descending population;
        for i = 1: a do
            Evaluate the objective function and the constraints for each individual;
            Select the best solution with its position;
        end
        Determine the new population;
        if t = t_max then
            Result: Print results
            Break
        else
            Continue
        end
    end
end
```

3.3 Sine cosine algorithm

The SCA is a metaheuristic optimization technique based on population methods. This technique employs the trigonometric sine and cosine functions to improve the exploration and exploitation of the solution space [32, 33]. Its most important characteristic is a fitness function that allows the algorithm to move over the infeasible region of the solution space. The purpose of this is that the algorithm can explore the solution space better by considering unfeasible solutions that may not comply with the problem physics. At each iteration, the positions of the particles are evaluated in relation to the fitness function and the set of constraints. Subsequently, each particle advances based on trigonometric rules. To do this, it is assumed that all the individuals has been evaluated in the mathematical model. Then, the best individual in the population is named x_{best}. Based on the current best solution, a new candidate y^{t+1} or x^{t+1} is selected to replace X_t if the following conditions are satisfied (see Eqs. (23) and (24)).

$$y_i^{t+1} = y_i^t + r_2 \sin(r_3)\,|r_4 x_{best} - x_i^t|, \quad i = 1,2,\dots,n, if\ \delta \geq \frac{1}{2} \tag{23}$$

$$x_i^{t+1} = x_i^t + r_2 \sin(r_3)\,|r_4 x_{best} - x_i^t|, \quad i = 1,2,\dots,n, if\ \delta \geq \frac{1}{2} \tag{24}$$

where, r_3 and r_4 are random numbers between 0 and 1 and $-\pi$ and π, respectively; and r_2 is ensures the algorithm convergence, which can be computed using Eq. (25), where t_{max} is the maximum number of iterations.

$$r_2 = 1 - \frac{t}{t_{max}} \tag{25}$$

Note that y^{t+1} and x^{t+1} are the potential individuals who will replace X_i^t. This switch can be carried out as follows: selecting y_i^{t+1} as a potential solution if $z_f(y_i^{t+1}) < z_f(X_i^{t+1})$; selecting X_i^{t+1} as a potential solution if $z_f(x_i^{t+1}) < z_f(X_i^{t+1})$; or, otherwise, $X_i^{t+1} = X_i^t$. Also, note that, if a component of X_i^{t+1} does not satisfy the boundary constraint defined for each decision variable, this solution is adjusted employing Eq. (26) to sustain the feasibility of the current population during the optimization process. where *ij* represents the spaces assign to each individual and its variables.

$$x_{ij} = x^{min} + r_1(x_j^{max} - x_j^{min}) \tag{26}$$

Finally, the algorithm's search process ends if the convergence criterion is fulfilled. Algorithm 3 shows the pseudocode of the SCA.

Algorithm 3. SCA pseudocode

Data: Adjust the SCA parameters;
Generate the initial population X^t;
Evaluate all the individuals x_i^t and find x_{best};
for $t = t_{max}$ **do**
  for $i = 1: n$ **do**
    Generate the initial population y_i^{t+1} and x_i^{t+1};
    Evaluate y_i^{t+1} and x_i^{t+1} in the fitness function;
    Evaluate by replacing the condition of each individual and create the descending population X^{t+1};
  end
  Evaluate the number of non-consecutive improvements of z_f;
  if $k \geq k_{max}$ **then**
    Select the best solution contained in X^{t+1};
    Return the decision variables and the value of the objective function;
    Result: Print results;
    Break;
  end
end

3.4 Vortex search algorithm

This algorithm is inspired by the vortex pattern of stirred fluids. Where the solution space is represented by a set of multiple, nonconcentric hyperspherical coordinates, with the first diameter being centered on the solution space which is defined by the minimum and maximum value of each variable (see Eq. (27)).

$$\mu_0 = \frac{x^{max} + x^{min}}{2} \tag{27}$$

where, x^{max} and x^{min} denote the range limits of each variable, and μ_0 is the hypersphere radius, generating a d-dimensional space. To generate the possible solution a Gaussian distribution is implemented in the solution space (see Eq. (28)).

$$s_i^t = p(\zeta_i^t, \mu_t, v) = ((2\pi)^d |v|)^{(1/2)} e^{\left(-\frac{1(\zeta_i^t - \mu_t)^T(\zeta_i^t - \mu_t)}{2\ v}\right)} \tag{28}$$

where, ζ_t is the random variables vector; μ_t, the actual center of the hypersphere at each iteration *t*; and *v*, a matrix of covariances. The radius of the hypersphere in VSA limits the random vector ζ_t. The matrix of covariances can be simplified as is formulated in Eq. (29) [41].

$$\sigma_0 = \frac{\max\{x^{max}\} - \min\{x^{min}\}}{2} \tag{29}$$

In the next step, the possible solutions are evaluated. Then the new center of the hypersphere is updated by selecting the best solution found in the previous iteration. Where the best overall solution is found by reducing the radius in each iteration (see Eq. (30)) achieving the convergence [17].

$$r_{t+1} = \sigma_0 \left(1 - \frac{t}{t_{max}}\right) e^{\left(\alpha - \frac{t}{t_{max}}\right)} \tag{30}$$

where, α is a constant parameter that controls the stepdown speed of the hyper-sphere radius that represents the solution space.

Finally, Algorithm 4 presents the pseudocode that describes the iteration process of the VSA.

Algorithm 4. VSA pseudocode

Data: Define μ_t and r_t for the hypersphere;
Generate each potential solution candidate s_i^t;
Verify the lower and upper bounds of each s_i^t;
Evaluate s_i^t in the fitness function;

Find the best current solution s_{best}^{t};
for t =1: t_{max} **do**
  Update the center $\mu_{t+1} = s_{best}^{t}$;
  Calculate the new radius r_{t+1};
  Generate new solution candidate s_i^{t+1};
  Verify the boundaries of each s_i^{t+1};
  Evaluate s_i^{t+1} in the fitness function;
  Find the best current solution s_{best}^{t+1};
  if $k \geq k_{max}$ **then**
    Select μ_{t+1} the solution;
    Result: Print results;
    Break;
  end
end

3.5 Fitness function

In metaheuristic optimization techniques, the FF is implemented to force the algorithm to operate in the infeasible regions of the solution space. That is, the FF considers unviable or penalized solutions to be correct to avoid falling into local optima and allow the search algorithm to explore and exploit the entire solution space. In this study (about minimizing the volume of a helical spring), the FF is defined by the sum of the volume and the set of penalties associated with the problem in Eq. (31).

$$FF = V + Pen \tag{31}$$

$$Pen = (p_{1+}p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8)_{\vartheta} \tag{32}$$

where, Eq. (32) represents the sum of the penalties associated with the constraints, and ϑ is the penalty factor, whose value is 100 in this case.

3.6 Coding

The coding of the problem addressed here uses a vector of 1 row and 3 columns (1×3). The first column stores the diameter of the spring wire (d); the second, the mean diameter of the spring coil (D); and the third, the number of coils of the wire (N_c).

Furthermore, to ensure a fair comparison between the implemented algorithms, each one was tuned to maximize its performance. For the tuning a PSO algorithm was employed as is purposed by Grisales-Noreña et al. [42].

3.7 Study case

The study case described in the reference [35] is used here to evaluate the proposed mathematical model. This paper presents the optimization of the design of a closed coil helical spring subjected to a compressive load, which is the common type of spring used in automotive industry. The maximum possible length is 35.56 cm (l_{max}), the mean diameter should not exceed 7.62 cm (D_{max}), and the wire's diameter should not be greater than 1.016 cm (d_{max}). The spring will withstand a maximum force of 453.6 kg (Fmax); therefore, the stress should not exceed the allowable maximum of S = 13288.02 kgf/cm^2. Table 1 lists the parameters needed to implement the algorithm, and Table 2 presents study case parameters. Figure 1 shows a schematic representation of the spring and the variables to be optimized.

Table 1. The tuned parameters along with their value and selected range

Algorithm	Parameter	Value	Range
CGA	Number of individuals (Ni)	50	[1-200]
	Number of iterations (tmax)	1221	[1-2000]
	Mutation parameter	0.5	[0-5]
	Number of random mutations (nM)	2	[0-Nv]
PSO	Number of individuals (Ni)	120	[1-200]
	Number of iterations (tmax)	1682	[1-2000]
	Cognitive component (C1)	1.1773	[0-2]
	Social component (C2)	1.564	[0-2]
	Maximum inertia (Wmax)	0.5549	[0-1]
	Minimum inertia (Wmin)	0.4377	[0-1]
VSA	Number of individuals (Ni)	105	[1-200]
	Number of iterations (tmax)	1123	[1-2000]
	Radius reduction interval (x)	0.00655	[0-0.1]
SCA	Number of individuals (Ni)	156	[1-200]
	Number of iterations (tmax)	1658	[1-2000]

Table 2. Study case parameters

Parameter	Value	Unit
G	808543.6	kgf/cm^2
δ	3.175	cm
N_{cmax}	25	-
α	1.05	-
β	3	-
d_{min}	0.508	cm
D_{min}	1.270	cm
N_{cmin}	15	-

4. RESULTS AND DISCUSSION

This section presents the results of the solution methods applied to minimize the volume of a helical spring. These methods were compared in terms of solution quality and solution times. The solution found by the techniques described above was simulated employing Inventor® software. Said simulation was used to validate the main constraint associated with the strength of the element in terms of deflections and the volume of the element.

The four techniques (i.e., CGA, SCA, PSO, and VSA) were programmed in Matlab® running on an HP Z600 computer with 8 GB of RAM and 4 cores. All the algorithms were run one thousand times, with the same mathematical models and under the same conditions, to determine the variation of the solution from the maximum, mean, and minimum values and the mean calculation time. The standard deviation, which expresses the repeatability of the solution, was evaluated based on these values.

Table 3 shows that the minimum values found by the four techniques are very similar. Nevertheless, in terms of d and D, PSO exhibited the highest values; and the VSA, the lowest.

For a deeper analysis, Table 4 compares the minimum, maximum, and mean values of the objective function (i.e., the spring's volume), as well as its standard deviation and the mean processing time.

Table 3. Minimum values of the variables found by the four optimization techniques

Method	d	D	N_C
CGA	0.6739	2.4048	15.0012
SCA	0.6743	2.4085	15.0057
PSO	0.6746	2.4117	15.0000
VSA	0.6739	2.4042	15.0000

Table 4. Comparison of the volumes found by each one of the four algorithms

Method	Minimum [cm³]	Mean [cm³]	Maximum [cm³]	Standard deviation [%]	Mean time [s]
CGA	45.8257	46.6423	50.4851	0.6363	0.9956
SCA	45.9243	46.7141	47.7937	0.3269	2.7607
PSO	46.0478	58.7404	87.1919	6.8557	0.7951
VSA	45.8002	46.0018	48.8888	0.3797	0.1560

Figure 2 shows the minimum solutions found by the four techniques implemented in this study to solve the sizing of a helical spring. The VSA found the lowest solution, i.e., a volume of 45.8002 cm³, which is 0.1%, 0.27%, and 0.54% lower than the solutions of the CGA, SCA, and PSO, respectively. Therefore, the VSA is considered the best solution so far.

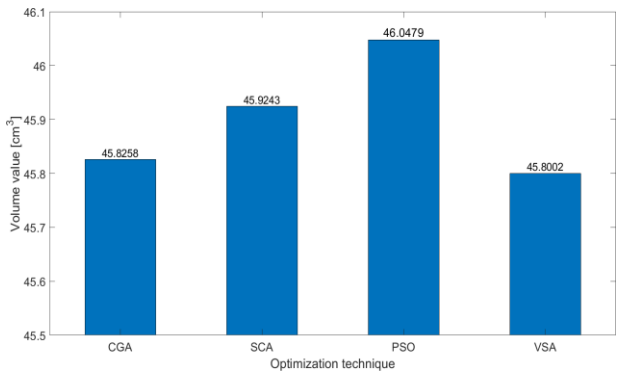

Figure 2. Minimum solution provided by the four optimization techniques implemented

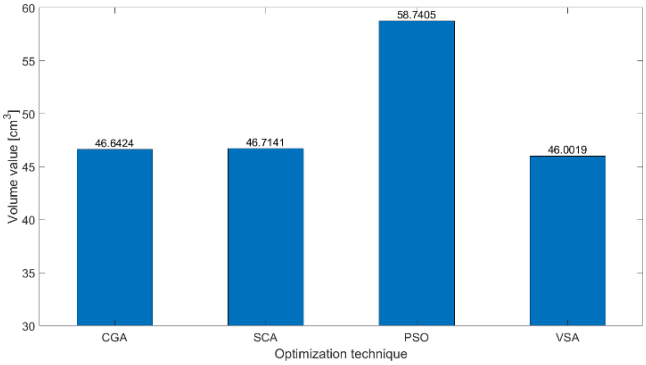

Figure 3. Average volume values found by the four optimization techniques

Figure 3 shows the average volume values found by the four optimization techniques. Note that the VSA still presents the lowest value, followed by CGA and SCA. In this case, the solution of PSO exhibits an increase of 27.5%, moving away from the minimum desired value and suggesting a high variation in the solutions found by this technique.

Figure 4 shows the maximum volume values found by the optimization techniques. The maximum value found by the SCA, the lowest among the four techniques, is 2.4% and 5.63% lower than those of the solutions provided by the VSA and the CGA, respectively.

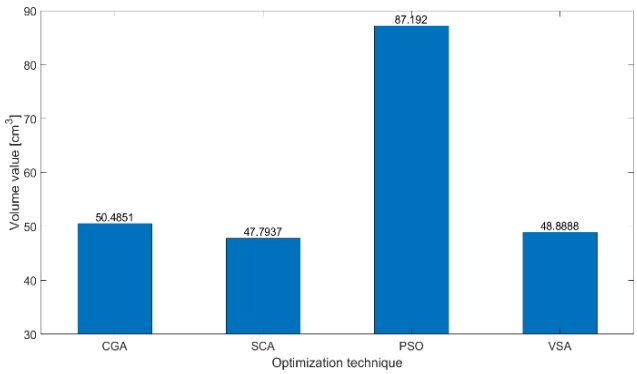

Figure 4. Maximum volume values found by the optimization techniques

The standard deviation was calculated based on the maximum and minimum values. The standard deviation of the SCA was 0.327%; and that of VSA, 0.379% (i.e., a difference of 0.052%). In this regard, the difference between these two techniques is minimal; however, the minimum value is the most important factor because it represents the most adequate solution provided by the optimization techniques. Thus, the VSA is the best technique in terms of solution quality. Although this technique did not exhibit the lowest standard deviation, it was still below 1, which guarantees the repeatability of the solution and proves that even the worst solution found by the VSA competes with those provided by the other techniques in terms of quality.

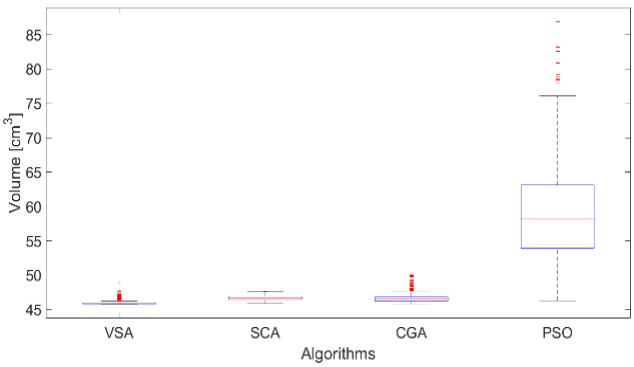

Figure 5. Boxplot fitness function for each optimization technique

In Figure 5, a boxplot is shown, in which the statistical comparison of the implemented algorithms is made. Where VSA stands out as the best-performing algorithm, exhibiting highly consistent results with minimal variability, showing volumes tightly concentrated within a narrow range of approximately 47 to 49 cm³, and only a few outliers. SCA and CGA also demonstrate consistency but to a slightly lesser extent. In contrast, PSO shows significantly higher variability, with a broader range of volumes (approximately 50 to 65 cm³)

and several notable outliers. This highlights VSA as the most reliable algorithm for achieving consistent and minimal volumes.

Regarding processing times, the VSA presented the fastest response, with an average of 0.156 s, followed by PSO, the CGA, and the SCA, in that order. The time difference between the SCA and the VSA was more than 1,000%, which means that the VSA outperforms the SCA in terms of processing time. However, the SCA provided an adequate solution with high repeatability and an acceptable minimum value; therefore, its implementation as a solution method for the problem addressed here is viable.

In general, these results suggest that the VSA exhibits the best response because it presents the minimum solution in terms of volume and complies with the set of constraints of the mathematical model, which represents a viable solution. Furthermore, in terms of processing time, the VSA is the fastest to provide a solution with high repeatability, and, although its standard deviation is not the lowest, it is still close to zero.

5. SOLUTION VALIDATION

A helical spring was designed in Inventor®, based on the data provided by the VSA technique, to verify that the value of the objective function found by the optimization algorithm matched the volume of the resulting element (see Figure 6).

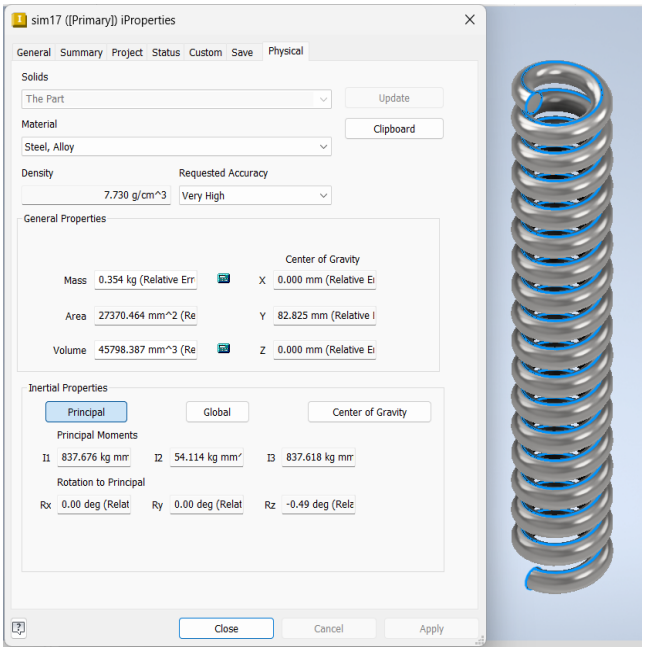

Figure 6. Verification of the volume of the helical spring

Once the dimensions and volume are confirmed to correspond with the output from the optimization algorithm, a deflection analysis is conducted to guarantee that the designed spring meets the specified conditions. This simulation is configured in Inventor's stress analysis environment. Where two plate-type elements are added to secure the component, and a fixed support constraint is applied to the lower plate to prevent axial displacement. The maximum force to which the spring will be subjected, 4445.28 N (453.6 kg), as mentioned in the problem statement, is also configured. Furthermore, the material properties, including the established stiffness modulus, are set up to ensure the design meets the conditions specified in the problem formulation.

The mesh configuration is refined, resulting in an average element size of 0.05 and a minimum element size of 0.1. These values are selected after running simulations with various parameters using an iterative convergence method. Figure 7 shows the location of the load, the movement restriction, and the mesh resulting from the parameters mentioned above.

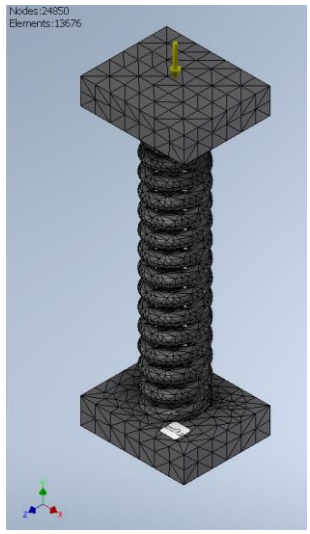

Figure 7. Configuration of the contact points and forces

Afterward, the element under the applied force was simulated to obtain the maximum deflection of the spring. Figure 8 shows this situation, where the deflection is equal to 4.54 cm; in the mathematical model, the value was 4.536 cm. This result shows that the mathematical model really represents the problem addressed, because the difference between the model and the simulation was 0.11%. Then, it was demonstrated that the element complies with the physical characteristics needed to serve its function.

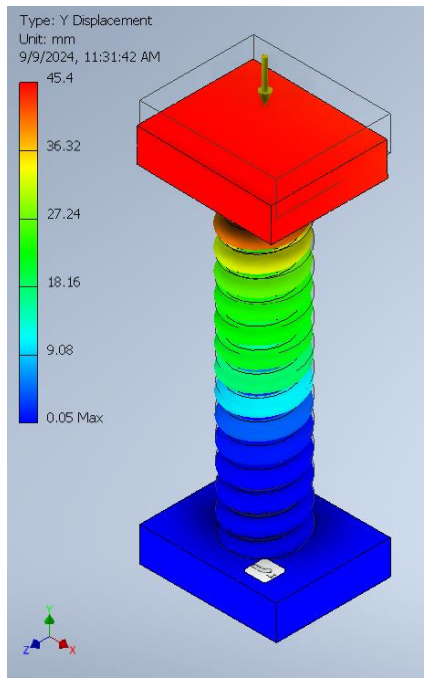

Figure 8. Validation of the deformation of the spring subjected to compression

6. CONCLUSIONS

In this study, the volume of a helical spring was minimized while meeting physical and technical requirements of the element, which were considered the set of constraints. This guarantees an adequate performance of the designed element under specific testing scenarios. The numerical simulations carried out in Matlab® showed that the mathematical models satisfactorily represented the element studied here; hence, they can be replicated to design elements under the conditions required by the industry (different forces, materials, and space requirements).

The implementation of the optimization algorithms proved that they are useful tools to solve this type of problems because they find an adequate solution in a short processing time and provide reliable designs. This means lower weights and volumes; therefore, the elements require less material for their manufacturing and implementation. Additionally, safety is guaranteed since the stresses and deformation constraints are respected, which means that the technical requirements are satisfied. Compared to the other techniques, the VSA provided the best solution in both scenarios due to the way in which it explores and exploits the solution space, which enables it to solve nonlinear non-convex optimization problems such as those addressed here. On the other hand, PSO found the worst solution to the problem due to the nature of the mathematical model, which needs more variation and independence between the decision variables. Therefore, PSO would need a different tuning for each mathematical model, which means longer preparation and testing times to provide an adequate solution in minimum processing times. Taken together, these results suggest that the VSA is the best technique to solve the mechanical design problems addressed in this paper.

Finally, the simulations performed in the specialized software provided an additional validation of the solution found by the VSA, verifying that it respected the set of constraints and met the physical and technical criteria, which demonstrated the quality of said solution. Additionally, the simulation of the applied loads proved the safety of the component. These additional results can be used to define the feasibility of designs and their improvement.

The combination of optimization techniques and simulations produces safe and reliable designs faster than conventional design methods. These tools that can be incorporated into pro- duction and manufacturing processes to improve their efficiency; avoid rework and early failures; and deliver lighter, safer, and more reliable designs. Such designs have a direct impact because they improve the efficiency of the machinery and equipment where the elements are installed and reduce their energy consumption as there are less energy losses related to power transmission.

The findings from this study provide a foundation for several avenues of future research. One area for further exploration is the refinement of the optimization algorithm to improve computational efficiency and accuracy. Moreover, expanding this research to include real-world case studies would provide valuable insights into the practical applications and limitations of the proposed methods. Where the mathematical models can be improved to consider additional physical constraints. In conclusion, the continued exploration of these topics has the potential to significantly advance the field and lead to the development of more effective optimization strategies in both academic and industrial settings.

ACKNOWLEDGMENT

This work was supported by the Instituto Tecnológico Metropolitano de Medellín (Colombia), under the research groups of Advanced Computing and Digital Design (SeCADD), which belongs to the research group of Advanced Materials and Energy (MATyER). It was also supported by Universidad Distrital Fransisco José de Caldas.

REFERENCES

[1] Dharmadhikari, N.D.K.S.R., Mahakalkar, S.G., Giri, J.P. (2013). Design and analysis of composite drive shaft using ANSYS and genetic algorithm: A critical review. International Journal of Modern Engineering Research, 3(1): 490-496. https://doi.org/10.6084/M9.FIGSHARE.1027448.V1

[2] Kim, H.S., Lee, D.G. (2005). Optimal design of the press fit joint for a hybrid aluminum/composite drive shaft. Composite Structures, 70(1): 33-47. https://doi.org/10.1016/j.compstruct.2004.08.010

[3] Sun, Z., Xiao, J., Yu, X., Tusiime, R., Gao, H., Min, W., Tao, L., Qi, L.L., Zhang, H., Yu, M. (2020). Vibration characteristics of carbon-fiber reinforced composite drive shafts fabricated using filament winding technology. Composite Structures, 241: 111725. https://doi.org/10.1016/j.compstruct.2019.111725

[4] Reddy, P.S.K., Nagaraju, C. (2017). Weight optimization and finite element analysis of composite automotive drive shaft for maximum stiffness. Materials Today: Proceedings, 4(2): 2390-2396. https://doi.org/10.1016/j.matpr.2017.02.088

[5] Jang, D., Jang, S. (2014). Development of a Lightweight CFRP coil Spring. SAE International. https://doi.org/10.4271/2014-01-1057

[6] Afolabi, S.O., Oladapo, B.I., Ijagbemi, C.O., Adeoye, A.O.M., Kayode, J.F. (2019). Design and finite element analysis of a fatigue life prediction for safe and economical machine shaft. Journal of Materials Research and Technology, 8(1): 105-111. https://doi.org/10.1016/j.jmrt.2017.10.007

[7] Yetgin, A., Karakaş, A., Acar, B., Özaslan, E. (2024). Failure analysis of a helical compression spring with relatively low spring index. Engineering Failure Analysis, 165: 108798. https://doi.org/10.1016/j.engfailanal.2024.108798

[8] Arslan, E., Genel, K. (2023). Failure analysis of automotive helical spring. Engineering Failure Analysis, 153: 107569. https://doi.org/10.1016/j.engfailanal.2023.107569

[9] Ke, J., He, J., Wu, Z., Xiang, Z. (2023). Fatigue reliability design of composite helical spring with nonlinear stiffness based on ply scheme design. Composite Structures, 319: 117119. https://doi.org/10.1016/j.compstruct.2023.117119

[10] Norton, R.L. (1999). Diseño de máQuinas (1st ed.). Prentice Hall.

[11] Mott, R.L. (2004). Machine Elements in Mechanical Design. Pearson Educación.

[12] García, G. (1985). Una teoría general de análisis en el diseño de elementos de máquinas. Ingeniería e Investigación, 13: 31-42.

[13] Shalal, O.D., Mitras, B.A. (2024). Hybridize the dwarf

mongoose optimization (DMO) algorithm to obtain the optimal solution for solve optimization problems. International Journal of Computational Methods and Experimental Measurements, 12(2): 175-184. https://doi.org/10.18280/ijcmem.120207
[14] Ghafil, H.N., Jármai, K. (2020). Dynamic differential annealed optimization: New metaheuristic optimization algorithm for engineering applications. Applied Soft Computing Journal, 93: 106392. https://doi.org/10.1016/j.asoc.2020.106392
[15] Qiao, J.F., Niu, Y.J. (2024). Thermodynamic multi-objective optimization: A deep learning and evolutionary algorithm approach. International Journal of Heat and Technology, 42(4): 1417-1426. https://doi.org/10.18280/ijht.420431
[16] Manrique, M.L., Montoya, O.D., Garrido, V.M., Grisales-Noreña, L.F., Gil-González, W. (2019). Sine-Cosine algorithm for OPF analysis in distribution systems to size distributed generators. Communications in Computer and Information Science, 1052: 28-39. https://doi.org/10.1007/978-3-030-31019-6_3
[17] Montoya, O.D., Gil-Gonzalez, W., Grisales-Norena, L.F. (2019). Vortex search algorithm for optimal power flow analysis in DC resistive networks with CPLs. IEEE Transactions on Circuits and Systems II: Express Briefs, 67(8): 1439-1443. https://doi.org/10.1109/tcsii.2019.2938530
[18] Gomez, J.F., Khodr, H.M., De Oliveira, P.M., Ocque, L., Yusta, J.M., Villasana, R., Urdaneta, A.J. (2004). Ant colony system algorithm for the planning of primary distribution circuits. IEEE Transactions on Power Systems, 19(2): 996-1004. https://doi.org/10.1109/TPWRS.2004.825867
[19] Rodriguez-Cabal, M.A., Betancur-Gómez, J.D., Grisales-Noreña, L.F., Montoya, O.D., Hincapie, D. (2021). Optimal design of transmission shafts using a vortex search algorithm. Arab Journal of Science and Engineering, 46(4): 3293-3300. https://doi.org/10.1007/s13369-020-05121-1
[20] Parouha, R.P., Tiwari, P. (2023). Updated Particle Swarm Optimization for Mechanical Engineering Design Optimization. Procedia Computer Science, 230: 830-837. https://doi.org/10.1016/j.procs.2023.12.050
[21] Tung, C.C., Lai, Y.Y., Chen, Y.Z., Lin, C.C., Chen, P.Y. (2023). Optimization of mechanical properties of bio-inspired Voronoi structures by genetic algorithm. Journal of Materials Research and Technology, 26: 3813-3829. https://doi.org/10.1016/j.jmrt.2023.08.210
[22] Rahati, A., Rigi, E. M., Idoumghar, L., Brévilliers, M. (2022). Ensembles strategies for backtracking search algorithm with application to engineering design optimization problems. Applied Soft Computing, 121: 108717. https://doi.org/10.1016/j.asoc.2022.108717
[23] Shojaei, I., Rahami, H. (2022). A heuristic swarm-based optimization method using multivariate normal distributions with self-adaptive variance matrices. Structures, 36: 372-391. https://doi.org/10.1016/j.istruc.2021.12.027
[24] Kohli, M., Arora, S. (2018). Chaotic grey wolf optimization algorithm for constrained optimization problems. Journal of Computational Design and Engineering, 5(4): 458-472. https://doi.org/10.1016/j.jcde.2017.02.005
[25] Shevale, D.V., Khaire, N.D. (2016). Analysis of helical compression spring for estimation of fatigue life. Imperial Journal of Interdisciplinary Research, 210: 2088-2093.
[26] Gaikwad, S., Kachare, P.S. (2014). Static analysis of helical compression spring. International Journal of Research in Engineering and Technology, 3(15): 835-838. https://doi.org/10.15623/ijret.2014.0315158
[27] Doğan, B., Ölmez, T. (2015). A new metaheuristic for numerical function optimization: Vortex search algorithm. Information Sciences, 293: 125-145. https://doi.org/10.1016/j.ins.2014.08.053
[28] Ben Guedria, N. (2016). Improved accelerated PSO algorithm for mechanical engineering optimization problems. Applied Soft Computing Journal, 40: 455-467. https://doi.org/10.1016/j.asoc.2015.10.048
[29] Yusup, N., Zain, A.M., Hashim, S.Z.M. (2012). Overview of PSO for optimizing process parameters of machining. Procedia Engineering, 29: 914-923. https://doi.org/10.1016/j.proeng.2012.01.064
[30] Rodriguez-Cabal, M.A., Marín, J.A., Grisales-Noreña, L.F., Montoya, O.D., Del Rio, J.A.S. (2018). Optimization of a drive shaft using PSO algorithm. WSEAS Transactions on Applied and Theoretical Mechanics, 13: 130-139.
[31] Moraes, A.O.S., Mitre, J.F., Lage, P.L.C., Secchi, A.R. (2015). A robust parallel algorithm of the particle swarm optimization method for large dimensional engineering problems. Applied Mathematical Modelling, 39(14): 4223-4241. https://doi.org/10.1016/j.apm.2014.12.034
[32] Patel, V.K., Rao, R.V. (2010). Design optimization of shell-and-tube heat exchanger using particle swarm optimization technique. Applied Thermal Engineering, 30(11-12): 1417-1425. https://doi.org/10.1016/j.applthermaleng.2010.03.001
[33] He, Q., Wang, L. (2007). An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Engineering Applications of Artificial Intelligence, 20(1): 89-99. https://doi.org/10.1016/j.engappai.2006.03.003
[34] Masoumi, A.P., Tavakolpour-Saleh, A.R. (2020). Experimental assessment of damping and heat transfer coefficients in an active free piston Stirling engine using genetic algorithm. Energy, 195: 117064. https://doi.org/10.1016/j.energy.2020.117064
[35] Jafarian, F. (2018). A modified non-dominated sorting genetic algorithm for multi-objective optimization of machining process. Journal of Engineering Science and Technology, 13(12): 4078-4093.
[36] Abdelsalam, A.M., El-Shorbagy, M.A. (2018). Optimization of wind turbines siting in a wind farm using genetic algorithm based local search. Renewable Energy, 123: 748-755. https://doi.org/10.1016/j.renene.2018.02.083
[37] Lin, C.H. (2013). A rough penalty genetic algorithm for constrained optimization. Information Sciences, 241: 119-137. https://doi.org/10.1016/j.ins.2013.04.001
[38] Bidabadi, M., Sadaghiani, A.K., Azad, A.V. (2013). Spiral heat exchanger optimization using genetic algorithm. Scientia Iranica, 20(5): 1445-1454.
[39] Mirjalili, S. (2016). SCA: A Sine Cosine Algorithm for solving optimization problems. Knowledge-Based Systems, 96: 120-133. https://doi.org/10.1016/j.knosys.2015.12.022
[40] Huang, H., Heidari, A.A., Xu, Y., Wang, M., Liang, G.,

Chen, H., Cai, X. (2020). Rationalized sine cosine optimization with efficient searching patterns. IEEE Access, 8: 61471-61490. https://doi.org/10.1109/ACCESS.2020.2983451

[41] Gupta, S., Deep, K., Moayedi, H., Foong, L.K., Assad, A. (2021). Sine cosine grey wolf optimizer to solve engineering design problems. Engineering with Computers, 37: 3123-3149. https://doi.org/10.1007/s00366-020-00996-y

[42] Grisales-Noreña, L.F., Cortés-Caicedo, B., Montoya, O.D., Bolaños, R.I., Moreno, C.A.M. (2024). Nonlinear programming for BESS operation for the improvement of economic, technical and environmental indices by considering grid-connected and stand-alone networks: An application to the territory of Colombia. Journal of Energy Storage, 98: 112856. https://doi.org/10.1016/j.est.2024.112856