International Journal of Theoretical and Applied Mechanics.
ISSN 0973-6085 Volume 12, Number 4 (2017) pp. 709-739

http://www.ripublication.com

Helical Spring Design Optimization in Dynamic Environment Based on Nature Inspired Algorithms

Godwin Raja Ebenezer N[a,*], Saravanan R[b], Ramabalan S[c], Navaneethasanthakumar S[d]

[a*]*Department of Mechanical Engineering, Loyola - ICAM College of Engineering and Technology, Chennai – Pin: 600034, Tamil nadu, India. (Corresponding Author)*

[b]*Department of Automobile Engineering, Dr.Mahalingam College of Engineering & Technology,, Pollachi-642 003. Coimbatore – District, Tamilnadu, India.*

[c]*Department of Mechanical Engineering, EGS Pillay Engineering College,Nagapattinam– 611002, Tamil nadu, India.*

[d]*Department of Mechanical and Industrial Engineering, Bahi Dar Institute of Technology, Bahir Dar, Ethiopia.*

Abstract

In this paper, nature inspired algorithms, namely, Simulated Annealing (SA), Fire fly (FA) and Cuckoo Search (CS) are proposed to get dynamic optimal solution for a helical spring design problem. The problem has four design variables and thirteen inequality constraints. To solve this problem, a dynamic model of the cylindrical helical spring having mechanical, geometric objective functions and dynamic constraints is considered. The dynamic constraints are related to the natural frequency of the spring. The sensitivity of the spring to its first natural frequency where resonance appears with large displacements is also accounted. The objective is to minimize the mass of the spring and to maximize the natural frequency of the spring. The design variables considered are: diameter of the wire, diameter of the middle helix, active number of coils and pitch of the spring. The problem is computed in MATLAB environment. Results of simulation are analysed and compared with literature.

Keywords: Helical springs; Hybrid mixed formulation; Natural frequencies; Dynamic response optimization; metaheuristics; Nature inspired Algorithms SA, FA, and CS.

1. INTRODUCTION

Springs are important members often used in machines to exert force, to absorb energy and to provide flexibility. In mechanical systems, wherever flexibility or relatively a large load under given circumstance is required, some form of spring is used. Helical spring is one of the common types of spring used in mechanical systems as shown in Figure 1. A designer must choose the right spring having greater strength and flexibility for use in the system. The designer can use synthetic tools to some extent, as they use only numerical optimization methods. Further, real time problems demand better spring designs. Hence dynamic optimal design is more desirable.

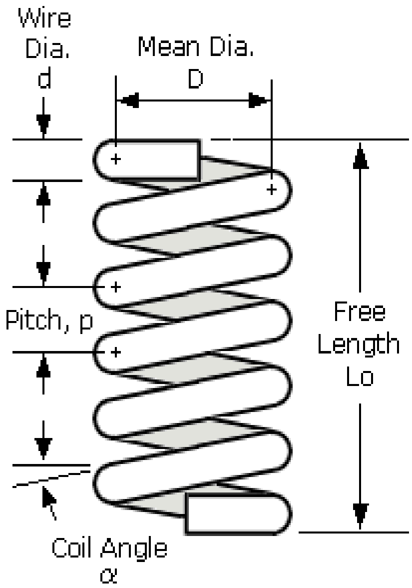

Figure 1. Cylindrical helical spring with design parameters

This paper presents, an optimal design in a dynamic way for a cylindrical helical spring using nature inspired metaheuristic algorithms. Nature inspired algorithms are characterised by easy realization, high precision, and rapid convergence. They are computationally efficient and display superiority in solving complex problems [5]. The MATLAB integrates computation, visualization, and programming in an easy-to-use environment [2]. It is a high-performance language for technical computing. In this work, dynamic model of the helical spring from literature [29] is considered for obtaining optimal values of design variables and objective function based on MATLAB. Optimization techniques have been widely employed in mechanical designs, [1], [2], [3] since many years. Researchers had solved many spring design problems using different algorithms [4]. They minimized the mass, volume, and stress distribution, movements subject to various geometric and mechanical parameters. Kulkarni and Balasubrahmanyam [11] minimized the mass, the free length and the volume of a helical spring. Yokota et al. [12] minimized the mass of a helical spring taking into account the shear stress, number of active coils, spring wire diameter and middle coil diameter using genetic algorithm. Deb and Goyal [13] and Kannan and Kramer [14] compared their results based on genetic algorithms with results of Sandregen [15] based on branch and bound approaches and augmented Lagrange method. Imaizumi et al. [16] and Hernandez [17] optimized the wire shape of spring. Xiao et al. [18] optimized helical spring based on Particle Swarm Optimization algorithm. Minimum mass of helical spring was the objective function, geometric parameters were design variables. Shear stress, maximum axial deflection, critical frequency, bucking, fatigue strength, condition of coils not touch, space and dimension were constraints. Kang and Kahraman [30] investigated dynamic behaviour of a double-helical gear pair both experimentally and theoretically. Fatih Karpat et al [31] used dynamic analysis to compare conventional spur gears with symmetric teeth and spur gears with asymmetric teeth and optimized asymmetric tooth design for minimum dynamic loads. Zheng Feng Bai and Yang Zhao [32] studied dynamic behaviour of planar mechanical systems including revolute joints with clearance using computational methodology. Letícia et al [33] applied FA algorithm to find the force and placement of dampers to control man made vibrations on foot bridge.

The following gives a brief account on dynamic optimization of helical springs: Philips and Costello [19] derived the equations of motion describing nonlinear behaviour of springs when subject to large impact of oscillations. Stokes [20] conducted analytical and experimental studies to investigate the spring radial displacement due to longitudinal impact. Mottershead [21] developed a finite element for solving differential motion equations. Yilidirim [22] developed the stiffness matrix for helical spring with circular and square sections from the linear relationship between effort and strain, taking into account the effect of transverse shear. The resolution of the modal equation was made by subspace iteration method to determine the natural frequencies of the spring. The influence of changes of parameters such as angle of the helix, middle coil diameter was studied. Forrester [23] analyzed the static and the dynamic behaviour of the spring by finite element and analytical methods to determine the stiffness and natural frequencies of the structure taking into account curvature of the spring, effects of shear and geometric effects of spring section. These methods were based on solving differential equations with boundary conditions. In the first analysis, the spring was modelled by an assembly of beam elements. In the second analysis, three-dimensional stiffness matrix of a helical spring was determined. Taktak et al. [24] developed a two node finite element with six degrees of freedom per node and modelled the behaviour of a three dimensional isotropic helical beam. Transverse shear and torsion effects and all geometric parameters were taken into account in the study of the dynamic response of the spring for harmonic excitations. Taktak et al. [29] optimized a cylindrical helical spring with dynamic constraints using Genetic Algorithm, incorporated in MATLAB code.

However, for optimal spring design, all these researchers had used either conventional techniques or less efficient techniques. Also, nature inspired algorithms which are superior and powerful are also not used. These algorithms are more desirable for solving optimization problems in real time applications [5], [6]. Hence, in this paper, it is proposed to use nature inspired algorithms, namely, Simulated Annealing (SA), Fire fly Algorithm (FA) and Cuckoo Search (CS) to obtain dynamic optimal design for a helical spring. This paper is based on [29] and the problem is solved in MATLAB environment.

This is how the paper is organised: in section 2, the method used for calculating the dynamic response of the helical spring based on the modal superposition method is presented. In section 3, the dynamic optimization method for the helical spring with the objective functions, the design variables and the constraints conditions of the problem are presented. In section 4, the proposed nature inspired optimization algorithms are presented. In section 5, numerical analysis is given. Finally, the conclusion and scope of the work is presented.

2. DYNAMIC ANALYSIS OF THE HELICAL SPRING

In this work, a cylindrical helical spring is considered. Modal Analysis is the basic dynamic analysis to calculate the natural frequencies of the helical spring. Output of the modal analysis is natural frequency and mode shapes. It helps to understand the structural behaviour of the spring in actual loading conditions. These days the method of dynamic analysis of mechanical elements with the aid of computer simulation is commonly. The present model acts as a suitable tool for performing analysis and simulation of spring dynamics.

The equation of motion of the helical spring is written as [24]:

$$[M]\{\ddot{U}\} + [C]\{\dot{U}\} + [K]\{U\} = \{F\} \tag{1}$$

$[M]$ and [K] are respectively the helical spring mass and stiffness matrices [24–26]. $[C]$ is the total damping matrix. $\{U\}$ is the global nodal displacements vector and $\{F\}$ is the vector of external forces. The description of the movement of this system with n

degrees of freedom can be made by its spatial coordinates or by its modal coordinates. The movement's equation of the structure without a second member admits a complete linear orthogonal real modes basis of the non-damped system. These eigen modes are characterized by the eigen pulsations ω_i and also by their eigen vectors$\{V_i\}$.The modal matrix is defined by [27]:

$$[\Phi] = [\{V_1\}, \{V_2\}, \dots \{V_{12}\}] \tag{2}$$

The projection of the motion's equation on the modal basis leads to build a system of n decoupled equations. The equation of the motion according to the generalized parameters is written as:

$$[M_m]\{\ddot{a}(t)\} + [C_m]\{\dot{a}(t)\} + [K_m]\{a(t)\} = \{F_m\} \tag{3}$$

where $a(t)$ is the generalized displacements vector defined as:

$$\{U(t)\} = [\Phi]\{a(t)\} \tag{4}$$

$$[M_m] = [\Phi]^T[M][\Phi] = diag(m_i) \tag{5}$$

is the generalized mass matrix.

$$[K_m] = [\Phi]^T[K][\Phi] = diag(m_i{\omega_i}^2) \tag{6}$$

is the generalized stiffness matrix.

$$[C_m] = diag(2m_i\omega_i\xi_i) \tag{7}$$

is the generalized damping matrix. $\xi_i$is the reduced modal damping coefficients [28]:

$$\{F_m\} = [\Phi]^T\{F\} = \begin{Bmatrix} f_1 \\ \vdots \\ f_2 \end{Bmatrix} \tag{8}$$

$\{F_m\}$ is generalized forces vector.

From Eq. (3) a system of the motion equations of n decoupled oscillators is obtained as follows:

$$m_i\ddot{a}_i + 2m_i\xi\omega_i\dot{a}_i + m_i{\omega_i}^2a_i = f_i \qquad i = 1, \dots, n \tag{9}$$

The modal frequency response of the variables $a_i(t)$ is simply the solution of n equations of motion transformed by Fourier is written:

$$m_i\omega_i{}^2 a_i(\omega) - m_i\omega^2 a_i(\omega) + 2jm_i\xi\omega_i\omega \dot{a_i}(\omega) = f_i\,(\omega) \tag{10}$$

The solution is:

$$a_i(\omega) = \frac{\frac{f_i\,(\omega)}{k_i}}{\left(1-\left(\frac{\omega}{\omega_i}\right)^2\right)+2j\xi_i\frac{\omega}{\omega_i}} \quad i = 1, \dots, n \tag{11}$$

where

$$k_i = m_i\omega_i{}^2 \tag{12}$$

The frequency response of the system is the product of modal variables by modes:

$$\{U(\omega)\} = \sum_{i=1}^{n} a_i(\omega)\{U_i\} = \sum_{i=1}^{n} \frac{\frac{f_i\,(\omega)}{k_i}}{\left(1-\left(\frac{\omega}{\omega_i}\right)^2\right)+2j\xi_i\frac{\omega}{\omega_i}}\{U_i\} \tag{13}$$

3. DEVELOPMENT OF THE DYNAMIC OPTIMIZATION METHOD OF THE HELICAL SPRING

Dynamic optimization is the unifying paradigm in spring analysis. Dynamic Optimization, in general sense, refers to the process of minimizing or maximizing the costs/benefits of some objective function over a period of time. Dynamic models are increasingly employed in spring optimization. Dynamic models of spring involve optimization over time. But this requires a more sophisticated theory and additional solution techniques. This optimization problem has objective functions, dynamic constraints, and choice variables as follows:

3.1. Design variables

Design variables are parameters that are chosen to describe the design of a system. Design variables are controlled by the designers. For computational design optimization of helical spring, objective function and constraints must be expressed as a function of design variables.

In this study a circular cross section spring is considered. Four geometrical proprieties of the spring are chosen as design variables, namely, the wire diameter d, the middle helix diameter D, active coil number n_a and the helix pitch P. These parameters are presented in the design parameters vector:

$$\{X\} = (x_1 \quad x_2 \quad x_3 \quad x_4)^T = (d \quad D \quad n_a \quad P)^T \tag{14}$$

The other parameters are supposed fixes.

3.2. Constraints conditions

Constraints are the conditions that must be met in the optimum design and include restrictions on design variables. These constraints define boundaries of the feasible and infeasible design space domain. Following hypotheses are assumed to be true:

1. The wire section is and remains circular.
2. The helix is slightly inclined ($\alpha < 7^0$).
3. The ends of the springs are ground and strengthened to make a perpendicular plans to the spring axis to support it without friction.

The constraints considered for the optimum design of helical spring considered are as follows:

3.2.1. Condition of shear stress

When helical spring is loaded by axial force, shear stress is exerted in the spring wire. By using superposition, the shear stress in the inside fibre of the spring can be computed. In order to avoid the structure damage, when the spring is loaded with an axial force $\vec{F}$,the condition of shear stress is, the maximum shear stress τ_{max} should be less than the allowable shear resistance R_{pg} [18]:

$$\tau_{max} = 1.66\left(\frac{d}{D}\right)^{0,16} \frac{8F_{max}D}{\pi d^3} \leq R_{pg} \tag{15}$$

$F_{max}$is the maximum axial load. So the condition of the shear stress can be expressed as follows [29]:

$$C_1(x){:}\, -R_{pg} + 4.23F_{max}\frac{x_2^{0.84}}{x_1^{2.84}} \leq 0 \tag{16}$$

3.2.2. Condition of maximum axial deflection

When a helical spring is loaded by the axial force $\vec{F}$, deflection is also exerted in spring. To obtain the equation for the deflection of a helical spring, an element of wire formed by two adjacent cross sections is considered. An element, the length of it is dx, cut from wire of diameter d. This element is on the surface of wire which is parallel to the spring axis. When loaded by force $\vec{F}$ and after deformation it will rotate through the angle and occupy the new position. The deflection of a wire element is given by [18]:

$$f = \frac{8FD^2(n_a+2)\sqrt{(\pi D)^2+P^2}}{G\pi d^4} \quad (17)$$

To ensure that the axial deflection is less than the maximum axial deflection, the maximum load that leads to the expected maximum axial deflection should be larger than the working force $F_{max} \geq F$. So the condition of the maximum axial deflection can be expressed by the following equation [29]:

$$C_2(x): \frac{G\pi f x_1^4}{8x_2^2(x_3+2)\sqrt{(\pi x_2)^2+x_4^2}} - F_{max} \leq 0 \quad (18)$$

3.2.3. Condition of fatigue strength

In shafts and many other machine members, fatigue loading in the form of completely reversed stress is quite low. But springs are always subject to high fatigue loads. In many cases, the number of cycles of required life may be small, say, several thousand for a padlock spring or a toggle-switch spring. But the valve spring of an automotive engine must sustain millions of cycles of operation without failure. So it must be designed for infinite life. The spring designed for variable stresses with a great number of cycles should be checked for resistance to fatigue. When a spring is subjected to alternating forces F_{min} to F_{max} the stresses for spring are obtained from these two equations [29]:

$$\tau_{min} = K\frac{8F_{min}D}{\pi d^3} \quad \text{and} \quad \tau_{max} = K\frac{8F_{max}D}{\pi d^3} \quad (19)$$

So the fatigue strength condition is as follows [18]:

$$S = \frac{\tau_o + 0.75\tau_{min}}{\tau_{max}} \geq \bar{S} \quad (20)$$

where $\tau_o$is the endurance limit of the spring for a zero-plus cycle. $\bar{S}$ is the safety allowable factor. If the design calculation accuracy and mechanical properties are higher, then $\bar{S} =$ 1.3– 1.7, if it is lower, then $\bar{S} =$ 1.8– 2. When the spring works with the number of cycles of the stress equal to $N = 10^6$ then $\tau_o = 0.33\sigma_B$ were σ_B is the ultimate tensile strength [18]. So the fatigue strength condition can be defined in this equation [29]:

$$C_3(x): \bar{S} - \frac{0.33\sigma_B x_1^{2.84}}{4.23F_{max}x_2^{0.84}} - 0.75\frac{F_{min}}{F_{max}} \leq 0 \quad (21)$$

3.2.4. Condition of buckling

A compression spring whose free length L_o is more than four times its mean diameter D should be checked for buckling. The length of a helical spring decreases when axial load is applied. Below critical length, some springs can flex laterally instead of continuing to decrease in length called as buckling. The corresponding length is called the buckling critical length L_k. If the spring is properly guided, such as having inside a tube over a bar, the amount of buckling can be greatly reduced. The finesse ratio $b = \frac{L_o}{D}$, with L_o as the spring free length, must not exceed the critical values b_c. The later is defined depending on the spring's ends [18]. For a helical spring, b is expressed by this equation:

$$b = \frac{L_o}{D} = \frac{n_a P + 2D}{D} \quad (22)$$

So the buckling condition is as follows [29]:

$$C_4(x): \frac{x_4 x_3 + 2x_1}{x_2} - b_c \leq 0 \quad (23)$$

3.2.5. Condition of coils not touch

When there is no load applied on a compression spring, the spring is free, and its height is L_o.As the load is applied, the coils move closer together, but do not touch. The maximum deflection exerted is $f_{max}$when the maximum load is applied. The dispersions during manufacture of the spring will make some coils touch before the theoretical length or the solid height (L_s) resulting in increased stiffness and forces. In this case, the characteristic of the spring becomes no longer linear. So it is necessary to avoid contact between the coils during operation. For this, the working length is limited to a value greater than the solid length. The minimum length of operation L_n must satisfy the following equations:

$$L_n = L_o - f_{max} \geq 0 \quad (24)$$

where

$$L_o = n_a P + 2D \quad (25)$$

In this case, to satisfy the requirement of coils not touch, following equation should be satisfied [29]:

$$n_a(d-P)+\frac{8F_{max}D^2(n_a+2)\sqrt{(\pi D)^2+P^2}}{G\pi d^4}\leq 0 \tag{26}$$

which can be expressed as follows[29]:

$$C_5(x): x_3(x_1-x_4)+0.33\frac{8F_{max}x_2^2(x_3+2)\sqrt{(\pi x_2)^2+x_4^2}}{G\pi x_1^2}\leq 0 \tag{27}$$

3.2.6. Condition of index C

Springs are manufactured either by hot-working or cold-working processes, depending upon the size of the material, the spring index $C=\frac{D}{d}$ where D is mean spring diameter, d is wire diameter, and the properties desired. Having suitable spring index is of extreme importance. The smaller the index the more difficult it is to wind the spring.

The coil ratio C is the ratio between the middle diameter of the spring and the diameter of the wire. For a large value of$C$, the wire is very little distorted inducing significant dispersions in the geometry of the spring. A small index induces strong internal stresses at the time of the manufacture and increases the stresses concentrations in usage: the spring became weakened; the wire can break at the time of the manufacture. For this, the manufacturers recommend using a value between 8 and 10 to facilitate the manufacture of the spring. The standards [18] indicate that the index must always be put between 4 and 20 to reduce internal stresses. When a spring is designed, the maximum value C_{max} and the minimal value C_{min} of spring index must be specified in such manner that: $C_{min}\leq C\leq C_{max}$.Therefore the condition of the spring index is expressed by the following equation [29]:

$$C_6(x): C_{min}\leq\frac{x_2}{x_1}\leq C_{max} \tag{28}$$

3.2.7. Condition of active coil

There are three types of ends commonly used for helical springs. In each case, when there is on load, the coils are separated. As the load is applied, the coils move closer together, but do not touch. Helical compression springs are used to exert force on mating parts. It is desirable that compression springs have as such possible with that mating parts at the ends of the springs. The number of these coils, n_a, depends on the way by which the spring ends are defined [18]. When a spring is designed, the maximum number of active coils $n_{a\,max}$ and the minimum number $n_{a\,min}$should be specified as $n_{a\,min} \leq n_a \leq n_{a\,max}$. Thus the condition of active coils is expressed in the following equation [29]:

$$C_7(x): n_{a\,min} \leq x_3 \leq n_{a\,max} \quad (29)$$

3.2.8. Condition of space and dimensions

In order to satisfy the requirements of fit and the specifications of dimensions, the wire diameter d, the spring pitch P and the middle coil diameter D must be severely limited. The conditions of space and dimensions can be expressed with these following equations [29]:

$$C_8(x): d_{min} \leq x_1 \leq d_{max} \quad (30)$$

$$C_9(x): D_{min} \leq x_2 \leq D_{max} \quad (31)$$

$$C_{10}(x) = P_{min} \leq x_4 \leq P_{max} \quad (32)$$

3.2.9. The frequency response condition

The frequency response of the helical spring is the product of the modal variables by the eigen modes as presented in Eq. (25). The amplitude of this response at the r^{th} degree of freedom excited in the q^{th} degree of freedom is:

$$A_{rq} = \sqrt{R_{rq}^2 + I_{rq}^2} \quad (33)$$

where

$$R_{rq} = \sum_{i=1}^{n} \frac{{\omega_i}^2 - \omega^2}{({\omega_i}^2 - \omega^2)^2 + 4{\xi_i}^2{\omega_i}^2\omega^2} \frac{f_i(\omega)}{m_i}\{U_i\} \quad \text{and} \quad (34)$$

$$I_{rq} = \sum_{i=1}^{n} \frac{-2\xi_i\,\omega\sqrt{\omega_i}}{({\omega_i}^2 - \omega^2)^2 + 4{\xi_i}^2{\omega_i}^2\omega^2} \frac{f_i(\omega)}{m_i}\{U_i\}$$

This amplitude is used as a dynamic constraint. Thus the condition related to the amplitude of the frequency response is expressed by the following equation [29]:

$$C_{11}(x): \sqrt{R_{rq}^2 + I_{rq}^2} - \overline{A_{rq}} \leq 0 \quad (35)$$

where $\overline{A_{rq}}$ is the desired limit of the frequency response amplitude.

3.2.10. Condition of natural frequency

If a wave is created by a disturbance at one end of a swimming pool, it will travel down the length of the pool, be reflected back at the far end, and continue this back-and-forth motion until it is finally damped out. The same effect happens to helical spring, and it is called spring surge. If one end of a compression spring is held against a flat surface and the other end is disturbed, a compression wave is created that travels back and forth from one end to the other exactly like the swimming-pool wave. When helical spring are used in applications requiring a rapid reciprocating motion, the designer must be certain that the physical dimensions of the spring are not such as to create a natural vibratory frequency close to the frequency of the applied force, otherwise resonance may occur resulting in damaging stresses, since the internal damping of spring material is quite low.

Hence, it is necessary to constrain the spring with an operating frequency different from its natural frequencies. Therefore it is proposed to increase the first natural frequency to a basic critical frequency f_b from 5 to 10 times the operating frequency f_w to avoid resonance. So the condition of the natural frequency can be expressed in this equation [29]:

$$C_{12}(x) = f_b - f_i(x) \leq 0 \tag{36}$$

where f_i is the i^{th} natural frequency of the helical spring.

3.3. The objective function

In this paper, the cylindrical helical spring is optimized in a dynamic way. Two objective functions are considered. In the design of the helical springs, getting minimum mass is the main aim. Therefore it is considered as the first objective function.

The helical spring is sensitive to its three first modes (two modes of bending and one of compression [24]. If it is excited with these frequencies, resonance occurs and the vibrations of great amplitude can damage the structure. That is why it is necessary to move these frequencies away from the excitation frequency by increasing natural frequencies helical spring. Therefore maximizing the first natural frequency is the second objective function in the dynamic optimal design of the spring considered.

The mass of the helical spring whose ends grinded and enhanced can be presented as follows [29]:

$$M = \rho_0 V - \rho_0 AL \tag{37}$$

where L is the curved length of the helical wire expressed by:

$$L = \int dL = \int ds = \rho \int_0^{2\pi(n_a+2)} d\theta = (n_a + 2)\sqrt{(\pi D)^2 + P^2} \quad (38)$$

So the spring mass is expressed by:

$$M = \rho_0 \frac{\pi d^2}{4} \sqrt{(\pi D)^2 + P^2}(n_a + 2) \quad (39)$$

The objective function, for the first case of study, can be expressed as a function of design variables as follows:

$$M = \frac{\pi}{4} \rho_0 x_1^2 (x_3 + 2)\sqrt{(\pi x_2)^2 + x_4^2} \quad (40)$$

For the second case, the first natural frequency computed by Eq. (36) is used as the objective function.

3.4. Mathematic model of helical spring dynamic optimal design

The mathematical model of dynamic optimal design of the helical spring with objective function and constraint conditions can be written as follows:

$$\min \text{ or } \max \quad f(x) \quad \text{with}$$

$$C_i(x) \leq 0 \qquad i = 1,2,\ldots,12 \quad (41)$$

$x = (d \quad D \quad n_a \quad P) = (x_1 \quad x_2 \quad x_3 \quad x_4)^T$ where $f(x)$ is the objective function which can be the mass of the spring or its first natural frequency. This dynamic optimization problem has four design variables and twelve non-linear inequalities constraints. As it is a complex problem, it is hard to solve it analytically. Further, numerical methods can only give a solution close to the optimum one. Therefore in the following section, superior nature inspired metaheuristic algorithms are used to solve the optimization problem for better solution.

4. METHODS OF SOLUTION

Nature inspired methods viz. Simulated Algorithm (SA), Firefly Algorithms (FA), Cuckoo search (CS) are proposed for solving this helical gear reducer optimization problem. These methods are discussed in the following section.

4.1 Nature inspired metaheuristic algorithms

The power of almost all modern metaheuristic algorithms comes from the fact that they imitate the best characteristics from nature, particularly biological systems evolved by natural selection for millions of years. Two very important characteristics are: selection of the most favourable species and adaptation to the environment. Numerically, it can be translated into two very important characteristics of modern metaheuristics: intensification and diversification. Intensification searches for the best current solutions, while diversification allows the algorithm to search the space efficiently.

4.1.1 Simulated Annealing (SA)

One of the earliest and most popular metaheuristic algorithms is Simulated Annealing (SA) [19], [20], [23], [24]. It was developed by Kirkpatrick, Gelatt and Vecchi in 1983. It is a trajectory-based, random search technique for global optimization. It mimics the annealing process in material processing when a metal cools and freezes into a crystalline state with the minimum energy and larger crystal size so as to reduce the defects in metallic structures. The annealing process involves the careful control of temperature and its cooling rate, often called annealing schedule. Annealing is the slow cooling of metal that produces good low energy state crystallization, whereas fast cooling produces poor crystallization. SA uses single point search method. It is a memory less search algorithm in the sense that no information is saved from previous searches. SA algorithm starts with a random initial design vector (solution) X_i and high temperature T. A second design point is created at random in the vicinity of the initial point and the difference in the function values (ΔE) at these two points is calculated as:

$$\Delta E = \Delta f = f_{t+1} - f_1 \equiv f(X_{i+1}) - f(X_i) \quad (42)$$

If the new solution's objective function value is smaller, the new solution is automatically accepted and becomes the current solution from which the search will continue. Otherwise the point is accepted with a probability $e^{\left(-\Delta E/kT\right)}$where k is the Boltzmann's constant. This completes one iteration of the SA. Due to the probabilistic acceptance of a non improving solution, SA can escape from local optima. At a certain temperature T predetermined numbers of new points are tested. The algorithm is terminated when current value of temperature is small enough or when changes in function values (Δf) are sufficiently small.

Simulated Annealing Algorithm

Objective function f (x), x = (x_1 , ..., x_p)T

Initialize initial temperature T_0 and initial guess $x^{(0)}$

Set final temperature T_f and max number of iterations N

Define cooling schedule $T \rightarrow \alpha T$, $(0 < \alpha < 1)$

while ($T > T_f$ and $n < N$)

Move randomly to new locations: $x_{n+1} = x_n + \epsilon$ (random walk)

Calculate $\Delta f = f_{n+1}(x_{n+1}) - f_n(x_n)$

Accept the new solution if better

if not improved

Generate a random number r

Accept if $p = \exp[-\Delta f / T] > r$

end if

Update the best $x*$ and $f*$

n=n+1

end while

Unlike the gradient-based methods and other deterministic search methods which have the disadvantage of being trapped into local minima, the main advantage of simulated annealing is its ability to avoid being trapped in local minima. In fact, it has been proved that simulated annealing will converge to its global optimality if enough randomness is used in combination with very slow cooling. Essentially, simulated annealing is a search algorithm via a Markov chain, which converges under appropriate conditions.

The initializing parameters and settings of SA used for this research are:

Initial temperature, T_init = 1.0;

Finial stopping temperature, T_min = 1e-10;

Min value of the function, F_min = -1e+100;

Maximum number of rejections, max_rej=500;

Maximum number of runs, max_run=150;

Maximum number of accept, max_accept = 50;

Initial search period, initial_search=500;

Boltzmann constant k = 1;

Energy norm (eg, Enorm=1e-8) Enorm=1e-5;

4.1.2 Firefly Algorithm (FA)

FA was developed by Xin-She Yang at Cambridge University in 2007 [18], [19], [21]. There are three idealized rules incorporated into the original Firefly algorithm (FA) : i) all fireflies are unisex so that a firefly is attracted to all other fireflies; ii) a firefly's attractiveness is proportional to its brightness seen by other fireflies, and so, for any two fireflies, the dimmer firefly is attracted by the brighter one and moves towards it, but if there are no brighter fireflies nearby, a firefly moves randomly; and iii) the brightness of a firefly is proportional to the value of its objective function. According to the above three rules, the degree of attractiveness of a firefly is calculated by the following equation:

$$\beta = \beta_o e^{-\gamma r^2} \tag{43}$$

where β is the degree of attractiveness of a firefly at a distance r, β_o is the degree of attractiveness of the firefly at r = 0, r is the distance between any two fireflies, and γ is a light absorption coefficient. The distance r between firefly i and firefly j located at X_i and $X_j$respectively is calculated as a Euclidean distance:

$$r = \|X_i - X_j\| = \sqrt{\sum_{k=1}^{d}\left(X_i{}^k - X_j{}^k\right)^2} \tag{44}$$

The movement of the dimmer firefly i towards the brighter firefly j in terms of the dimmer one's updated location is determined by the following equation:

$$X_{i+1} = X_i + \beta_o e^{-\gamma r^2} = \left(X_j - X_i\right) + \alpha\left(\text{rand} - \frac{1}{2}\right) \tag{45}$$

The third term in (45) is included for the case where there is no brighter firefly than the one being considered and rand is a random number in the range of [0, 1].

The Firefly algorithm

Objective function f (x),x = (x_1 , ..., x_d)T

Generate initial population of fireflies x_i (i = 1, 2, ..., n)

Light intensity I_i at x_i is determined by f (x_i)

Define light absorption coefficient γ

while (t <Max Generation)

for i = 1 : n all n fireflies

for j = 1 : n all n fireflies (inner loop)

if (I_i < I_j), Move firefly i towards j; end if

Vary attractiveness with distance r via exp[−γr]

Evaluate new solutions and update light intensity

end for j

end for i

Rank the fireflies and find the current global best g ∗

end while

Postprocess results and visualization

In firefly algorithm, it is possible to adjust the parameter γ and α so that it can outperform both the random search and PSO. Also, FA can find the global optima as well as the local optima simultaneously and electively. Another advantage of FA is that different fireflies will work almost independently and hence suitable for parallel implementation. It is even better than GAs and PSO because fireflies aggregate more closely around each optimum.

In implementation of (FA) of this work, the values of the parameters used are: 20 fireflies, Number of iterations = 250, α = 0.5, γ = 1 and β_o = 0.2. These parameters have been chosen after adjustment to suit for solving the helical gear design optimization problem.

4.1.3 Cuckoo search Algorithm (CS)

Cuckoo search (CS) is one of the latest nature-inspired metaheuristic algorithms, developed in 2009 by Xin-She Yang of Cambridge University and Suash Deb of C. V. Raman College of Engineering [19],[22]. Cuckoo Search (CS) represents a new optimization metaheuristic algorithm, which is also biologically inspired by the cuckoos' manner of looking for nests where they could lay eggs.

Cuckoos lay their eggs in other birds' nests and the host birds later take care of cuckoo chicks. Cuckoos usually choose the nest of a bird that has just laid its eggs so that they can be sure that their eggs would hatch first because cuckoo eggs hatch earlier than their host eggs birds. Some types of cuckoos have adapted to laying their eggs in other birds' nests so that their eggs are quite similar to the eggs of the host birds. When a cuckoo chick is hatched, it instinctively pushes out of the nest the host bird chicks and eggs that have not yet hatched to receive all the food brought in. A cuckoo chick can mimic the call of host chicks. If the host birds realize that a cuckoo egg has been laid in, they either remove the egg or abandon the nest.

In this optimization algorithm, each nest represents a potential solution. The cuckoo reproduction process in the algorithm is simplified by three rules:

1. Each cuckoo lays an egg in a randomly chosen nest;

2. The best nests carry over to the next generation of cuckoos;

3. The number of available host nests is fixed (limited), and the egg laid by a cuckoo is discovered by the host bird with a probability, $p_a$which ranges 0.1. Birds can detect only the worst nests so that they are losing from the population.

The initial population of nests with the size, n which are randomly distributed over the search space, is generated first. The randomly chosen initial solutions of design variables are defined in the search space by the lower and upper boundaries.

The new nest, for example i^{th}, is generated according to the following law,

$$x_i^{t+1} = x_i^{t} + \alpha \oplus levy\lambda \tag{46}$$

where $\alpha > 0$ is the step size whose value depends on the optimization problem, and t is the current generation. Step size is multiplied by the random numbers with Lévy's distribution, and such random motion is called Lévy flight. Levy flight has the step-lengths distributed according to the following probability distribution:

$$Lévy \sim u = t^{-\lambda}, 1 < \lambda \leq 3 \tag{47}$$

Lévy flight represents a variation of random walk, in which the step length is determined by Lévy distribution. Lévy flight represents one of the ways of motion used by birds for searching for food in the environment. When there is some food in the environment, animals perform motion which is analogous to Brownian motion. If they cannot find any food, animals start moving in the manner analogous to Lévy flight, i.e., by combining short and long steps in different directions thus searching a considerably larger space. The numerical algorithm proposed by Mantegna (1994), using the exponential law, was used for generation of Lévy distribution in the CS algorithm. It is recommended that the step size should be, $L/100v$ where L is the size of the space which is searched. There is a danger that Lévy flight may become too "aggressive" for large values of the step size and that new solutions may go out of the space which is searched.

Cuckoo Search via L´vy Flightse

Objective function f (x), x = (x_1 , ..., x_d)T

Generate initial population of n host nests xi

while (t <Max Generation) or (stop criterion)

  Get a cuckoo randomly/generate a solution by L´vy flightse

    and then evaluate its quality/fitness Fi

  Choose a nest among n (say, j) randomly

  if ($F_i > F_j$),

   Replace j by the new solution

  end

  A fraction (p_a) of worse nests are abandoned

    and new ones/solutions are built/generated

  Keep best solutions (or nests with quality solutions)

  Rank the solutions and find the current best

end while

Post process results and visualization

Here we have used n = 25 nests, α = 1 and p_a = 0.25 as parameters of CS for implementation. We have also tried to vary the number of host nests (or the population size n) and the probability p_a . We have used n = 5, 10, 15, 20, 30, 40, 50, 100, 150, 250, 500 and p_a = 0, 0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, 0.5. From our simulations, we found that parameters n = 25 and p_a = 0.25 are sufficient for this optimization problem.

5. NUMERICAL RESULTS

The dynamic optimization problem of the helical spring having defined objective function and constraints equations is solved using the proposed nature inspired optimization algorithms. The numeric model of the dynamic response is simulated using MATLAB. The mechanical and geometrical features of the spring are given in Table 1. The parameters used to calculate the constraints C_1 to C_{12} of the problem are presented in Table 2.

Table 1. Mechanical and geometrical features of the spring

Properties	Value
Maximum amplitude of the frequency response	$A_{33} = 200$ mm
Critical frequency	$f_b = 110$ Hz
Maximum load	$F_{max} = 100N$
Minimum load	$F_{min} = 0\ N$
Maximum deflection	$f_{max} = 10mm$
Margin actives coil number	$5 \leq n_a \leq 50$
Margin wire diameter	$1\ mm \leq d \leq 4\ mm$
Margin spring pitch	$1\ mm \leq P \leq 60\ mm$
Margin middle helix diameter	$10\ mm \leq D \leq 30\ mm$
Margin spring index	$4 \leq C \leq 12$

Table 2. The parameters of the constraints

Properties	Value
Material	Steel
Poisson coefficient	$\nu = 0.3$
Young's modulus	$E = 2.1248\ 10^{11}\ N/m^2$
Density	$\rho = 8000\ kg/m^3$
Shear modulus	$G = 83.10^9\ N/m^2$
Ultimate tensile strength	$\sigma_B = 2000\ MPa$
Allowable shear resistance	$R_{pg} = 6.10^8\ N/m^2$
Finesse ratio	$b_c = 3.7$
Safety allowable factor	$S = 1.3$

The spring considered is a clamped-free circular cross section helical spring. A harmonic axial excitation with amplitude $100N$ and vibration frequency $f_w = 25\ Hz$ is applied on the free extremity of the spring. In this paper, two parameters are optimized: the spring mass and its first natural frequency subject to constraints. In the following, the optimization problem is discussed.

5.1. Optimization of the spring mass

In this case, the objective is to minimize the mass of the helical spring. The proposed nature inspired algorithms are applied to search for optima and the simulation is done in MATLAB. Table 3 presents the results of final design variables, the optimal helical spring mass, the number of iterations and the time of running of each algorithm. The dynamic optimization problem subject to constraints is [29]:

$$\min\ M(x) = 6283.185 x_1^2 (x_3 + 2)\sqrt{(\pi x_2)^2 + x_4^2}$$

$$C_1(x): -6.10^8 + 423\frac{x_2^{0.84}}{x_1^{2.84}} \leq 0$$

$$C_2(x): 32.594.10^7 \frac{x_1^4}{x_2^2(x_3+2)\sqrt{(\pi x_2)^2+x_4^2}} - 100 \leq 0$$

$$C_3(x): 1.3 - 1.56.10^6 \frac{x_1^{2.84}}{x_2^{0.84}} \leq 0$$

$$C_4(x): \frac{x_4 x_3 + 2x_1}{x_2} - 3.7 \leq 0$$

$$C_5(x): x_3(x_1 - x_4) + 3.068.10^{-9} \frac{x_2^2(x_3+2)\sqrt{(\pi x_2)^2+x_4^2}}{x_1^4} \leq 0$$

$$C_6(x): 4 \leq \frac{x_2}{x_1} \leq 12$$

$$C_7(x): A_{33} - 0.2 \leq 0$$

$$C_8(x): 110 - f_1\,(x) \leq 0$$

$$C_9(x): 0.001 \leq x_1 \leq 0.004$$

$$C_{10}(x): 0.01 \leq x_2 \leq 0.03$$

$$C_{11}(x): 5 \leq x_3 \leq 50$$

$$C_{12}(x): 0.001 \leq x_4 \leq 0.06$$

The nature inspired algorithms used in this work converge quickly and give better results but with different speeds. From Table 3 it is observed that the Firefly Algorithm gives the best results. Optimum mass of the helical spring obtained is 7.773 g. The corresponding optimal design variables are: wire diameter d = 2mm, the middle helix diameter D = 13.5 mm; number of active coils n_a = 5 coils and the spring pitch P = 6.4 mm. Figure 3, shows a comparison of performance between FA (being the best technique) and literature.

The proposed algorithms SA, FA and CS take less time of run (3.033s, 1.125s and 10.529s) than Genetic Algorithm (42.61s) of literature. The same has been shown in the graph as well (Figure 4).The performance of the nature inspired algorithms SA, FA and CS are shown in the Figure 2.The minimum mass obtained by all the algorithms are almost the same (0.007970kg, 0.007773kg, 0.007816kg). SA gives the least wire diameter (0.0019m**)** whereas FA and CS give the same value (0.0020m).The same trend is continued in middle helix diameter value as well (0.0128m, 0.0135m, 0.0135m).The number of coils according to SA is 5.8306 and other techniques (FA and CS) are 5.Pitch of the helical spring obtained by SA, FA and CS is 0.0053m, 0.0064m, 0.0074m respectively.

Table 3. Optimized values and corresponding design values of this work (highlighted values) and literature [29] for minimum mass

	Simulated Annealing	**Firefly Algorithm**	**Cuckoo Search Algorithm**	Pattern search	Genetic Algorithm	Active set	Interior point
$\boldsymbol{M}$ (kg)	**0.007970**	**0.007773**	**0.007816**	0.007864	0.008609	0.007741	0.008207
$\boldsymbol{x_1}(m)$	**0.0019**	**0.0020**	**0.0020**	0.002	0.002	0.002	0.002
$\boldsymbol{x_2}(m)$	**0.0128**	**0.0135**	**0.0135**	0.014	0.014	0.014	0.01
$\mathbf{x_3}(spire)$	**5.8306**	**5.0000**	**5.0000**	5.04	5.696	5	10.109
$\boldsymbol{x_4}(m)$	**0.0053**	**0.0064**	**0.0074**	0.005	0.004	0.004	0.003
Number of Iterations / iteration Number	23288	5000	100000	3	7	6	151
Time of run(s)	**3.033**	**1.125**	**10.529**	03.15	42.61	02.11	09.00

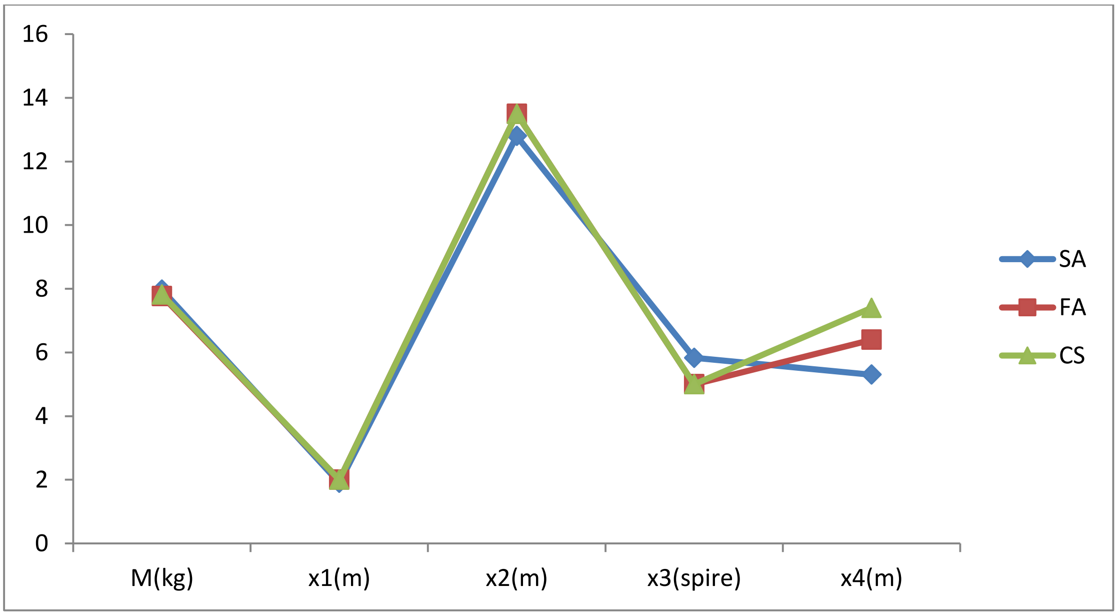

Figure 2. Graphs obtained by various techniques of this work for minimum mass

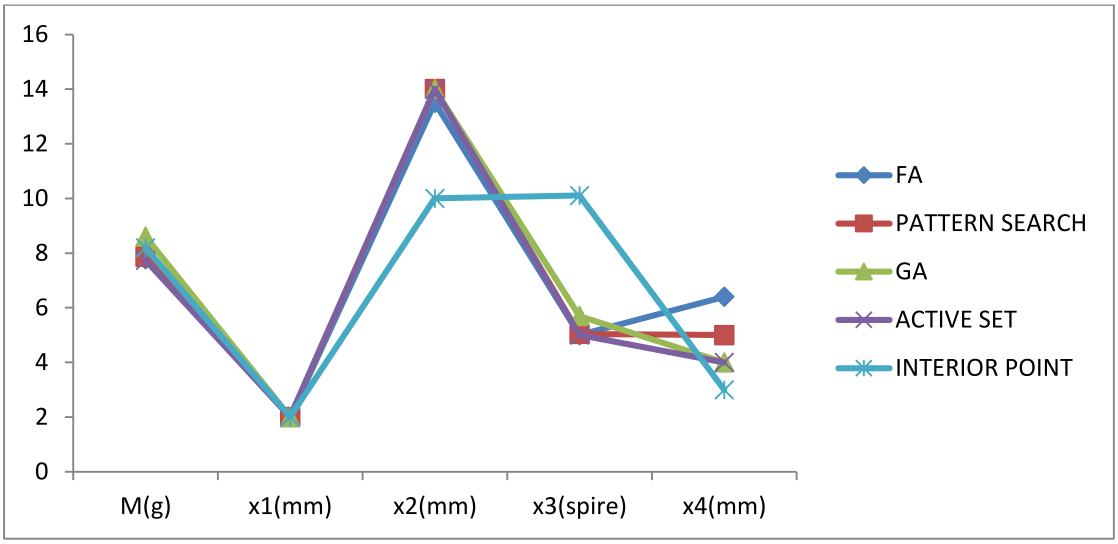

Figure 3. Graphs by FA technique of this work and literature [29] for minimum mass

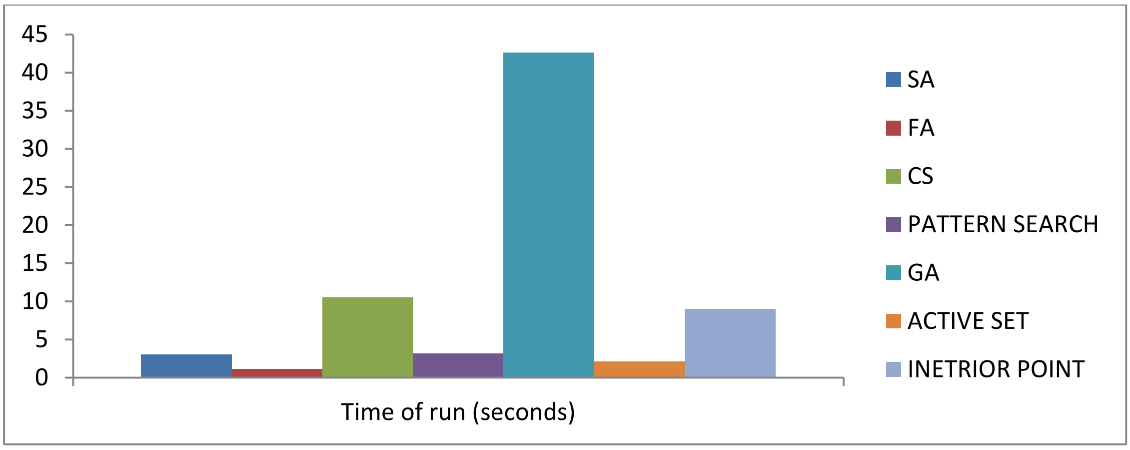

Figure 4. Computational time of all techniques of this work and literature [29] for minimum mass

To verify whether the final design variables respect the constraints including the dynamic ones, the first natural frequency of the optimized helical spring is computed analytically. This is equal to f_1 = 142.35 Hz. This value respects the constraint C_8 which indicates, the first natural frequency must be greater or equal to 110 Hz. So, the optimized helical spring has the minimum mass and also satisfies the dynamic constraint.

5.2. Optimization of the first natural frequency

The first natural frequency is chosen as the second objective function. In this case, two more constraint conditions, namely, the mass not to exceed 30 g (Constraint conditionC_{13}) and the third natural frequency not to exceed 160 Hz are added. Now optimization is carried out in the same way as in the case of the optimization of the mass of the spring, taking into account these two newer constraint conditions. The optimization problem is as follows [29]:

max $f_1(x)$ $with$

$$C_1(x): -6.10^8 + 423\frac{x_2^{0.84}}{x_1^{2.84}} \leq 0$$

$$C_2(x): 32.594.10^7 \frac{x_1^4}{x_2^2(x_3+2)\sqrt{(\pi x_2)^2 + x_4^2}} - 100 \leq 0$$

$$C_3(x): 1.3 - 1.56.10^6 \frac{x_1^{2.84}}{x_2^{0.84}} \leq 0$$

$$C_4(x): \frac{x_4 x_3 + 2x_1}{x_2} - 3.7 \leq 0$$

$$C_5(x): x_3(x_1 - x_4) + 3.068.10^{-9} \frac{x_2^2(x_3+2)\sqrt{(\pi x_2)^2 + x_4^2}}{x_1^4} \leq 0$$

$$C_6(x): 4 \leq \frac{x_2}{x_1} \leq 12 C_8(x): 110 - f_1\,(x) \leq 0$$

$$C_7(x): A_{33} - 0.2 \leq 0$$

$$C_8(x): 110 - f_1\,(x) \leq 0$$

$$C_9(x): 0.001 \leq x_1 \leq 0.004$$

$$C_{10}(x): 0.01 \leq x_2 \leq 0.03$$

$$C_{11}(x): 5 \leq x_3 \leq 50$$

$$C_{12}(x): 0.001 \leq x_4 \leq 0.06$$

$$C_{13}(x): 6283.185 x_1^2(x_3+2)\sqrt{(\pi x_2)^2 + x_4^2} - 0.03 \leq 0$$

From Table 4, it is also observed that the Firefly algorithm gives the maximum value of the first natural frequency. The optimal value of the first natural frequency is f_1 = 53.323 Hz with the corresponding final design values: wire diameter d = 2.7 mm; middle helix diameter D = 25.6 mm; number of active coils n_a = 5.163; spring pitch P = 8.9 mm. Illustrative graphs are drawn for various techniques of this work (Figure 5).As FA is found to be the best technique, a comparison graph is drawn between FA and literature [29] for the optimized values and corresponding design values for maximum first natural frequency (Figure 6).

Table 4. Optimized values and corresponding design values of this work (highlighted values) and literature [29] for maximum first natural frequency

	Simulated Annealing	**Firefly Algorithm**	**Cuckoo Search Algorithm**	Pattern search	Genetic Algorithm	Active set	Interior point
$\boldsymbol{f_1}$ (Hz)	**53.322**	**53.323**	**53.321**	80.177	80.139	81.004	80.852
$\boldsymbol{x_1}(m)$	**0.0025**	**0.0027**	**0.0026**	0.003	0.003	0.003	0.003
$\boldsymbol{x_2}(m)$	**0.0248**	**0.0256**	**0.0237**	0.023	0.022	0.023	0.022
$\boldsymbol{x_3}(spire)$	**5.037**	**5.1630**	**6.0426**	5	5.246	5.122	5.116
$\boldsymbol{x_4}(m)$	**0.0096**	**0.0089**	**0.0064**	0.005	0.005	0.005	0.005
Number of Iterations / iteration Number	13031	5000	100000	4	7	8	298
Time of run(s)	**1.740**	**1.145**	**10.665**	07.24	26.59	02.29	40.38

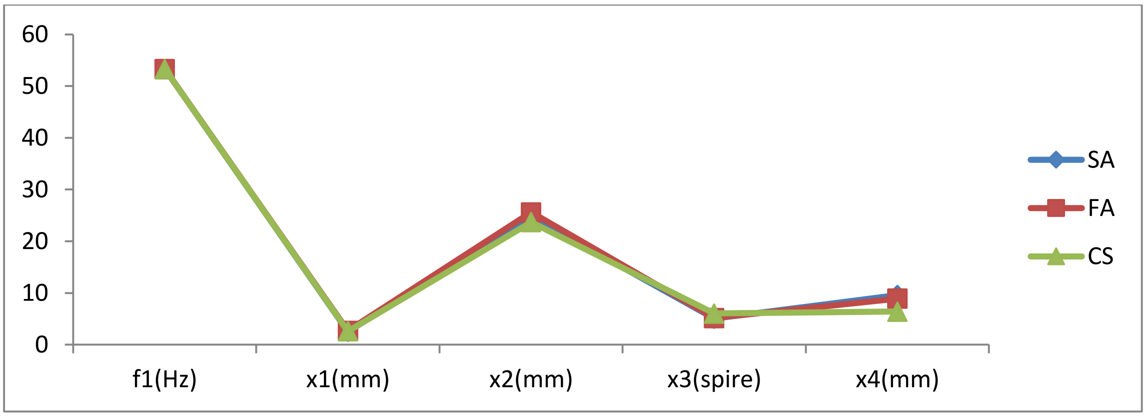

Figure 5. Graphs obtained by various techniques of this work for maximum first natural frequency

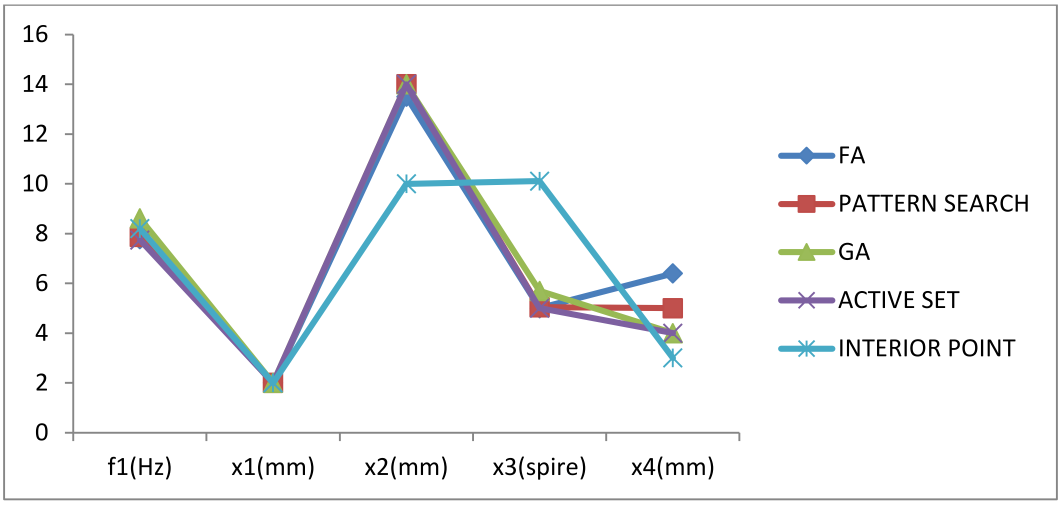

Figure 6. Graphs by FA technique of this work and literature [29] for maximum first natural frequency

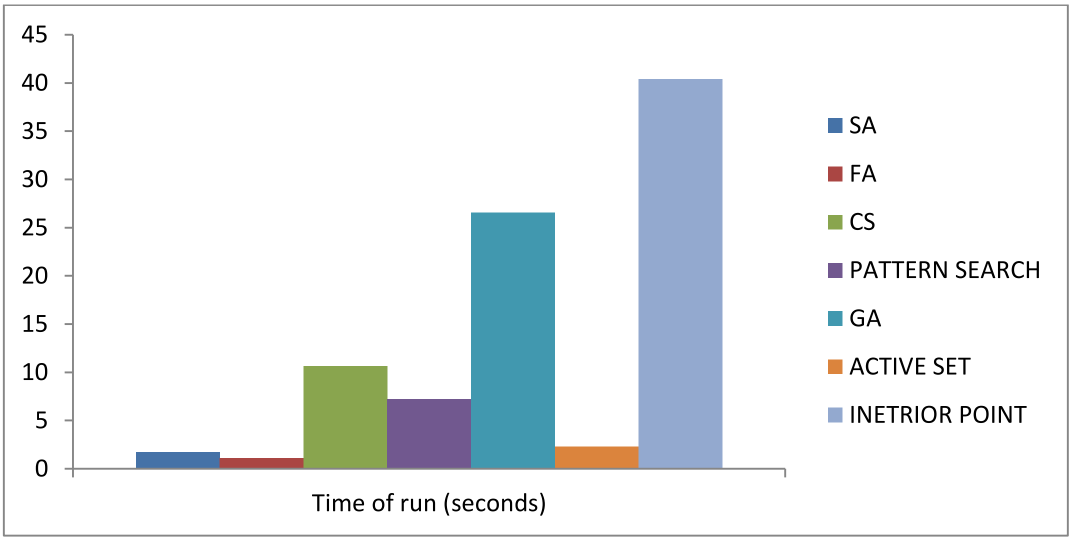

Figure 7. Computational time of all techniques of this work and literature [29] for maximum first natural frequency

Figure 5, shows a comparison of performance between FA (being the best technique) and literature. The proposed algorithms SA, FA and CS take less time of run (1.740s, 1.145s and 10.665s) than Genetic Algorithm (26.59s) of literature (Figure 7). The performance of the nature inspired algorithms SA, FA and CS for maximum first natural frequency is shown in Figure 5. The maximum first natural frequency obtained by all the algorithms are almost the same (53.322Hz, 53.323Hz, 53.321Hz). SA gives the least wire diameter (0.0025m).CS gives the least middle helix diameter

(0.0237m**).** CS also gives the highest number of coils (6.0426). Pitch of the spring obtained by SA, FA and CS is 0.0096m, 0.0089m, 0.0064m respectively.

By analytical calculation, substituting the design variables of the final result in the final objective function and the constraints, the first natural frequency f_1 got is 53.32 Hz and the third natural frequency of the spring is 155.81 Hz. From this, it is concluded that the constraints are respected and thus the results are verified. Further, the optimal helical spring gives a reasonably fair margin of frequencies band [0 – 53 Hz] for the user to work without falling into resonance. Also it is observed that the proposed natured inspired optimization techniques of this work are faster than the ones in literature. Nature inspired algorithms (SA), (FA) and (CS) of this work take less time of run (1.740s, 1.145s and 10.665s) than that of Genetic Algorithm (26.59s) of literature.

6. CONCLUSION

In this paper, optimal design for a cylindrical helical spring in a dynamic manner using nature inspired algorithms, namely, Simulated Annealing (SA), Fire fly (FA) and Cuckoo Search (CS) in MATLAB environment is got. The mass of the helical spring and its first natural frequency are optimized. All constraints are well satisfied. Three observations are made:

1. For computing optimum mass, FA gives the best result for computing optimal mass and design variables of the helical spring as follows: M =7.773 g; wire diameter d = 2mm, middle helix diameter D = 13.5 mm; number of active coils n_a = 5 coils and spring pitch P = 6.4 mm. The proposed algorithms SA, FA and CS take less time of run (3.033s, 1.125s and 10.529s) than GA (42.61s) of literature.

2. For computing optimum first natural frequency and optimal design variables, again FA gives the best result: f_1 = 53.323 Hz; wire diameter d = 2.7 mm; middle helix diameter D = 25.6 mm; number of active coils n_a = 5.163; spring pitch P = 8.9 mm. Even here, SA, FA and CS of this work take less time of run (1.740s, 1.145s and 10.665s) than GA (26.59s) of literature.

3. The proposed nature inspired algorithms SA, FA and CS are faster and computationally more efficient than GA of literature. Furthermore, this research can be used as reference for other similar mechanical element optimal designs.

REFERENCES

[1] Joseph Edward Shigley, Mechanical Engineering Design, First Metric Edit, McGraw-Hill, 1986.

[2] Yuan sicong, MATLAB Language and Its Application in Mechanical Engineering, China Machine Press, Beijing, 2008.

[3] J.S Arora, Introduction to Optimum Design, Academic press, third edition, 2012.

[4]. X. S. Yang, Firefly algorithm, L´vy flights and global optimization, in: Research and Development in Intelligent Systems XXVI, (Eds M. Bramer et al.), Springer,London, 2010 pp. 209-218.

[5]. X. S. Yang, Nature-Inspired Metaheuristic Algorithms, Luniver Press, 2008.

[6]. E. G. Talbi, Metaheuristics: From Design to Implementation, Wiley, 2009.

[7]. X. S. Yang, Firefly algorithms for multimodal optimization, in: Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Science, 2009 5792, 169-178.

[8]. Yang X. S. and Deb S,, Engineering optimization by cuckoo search, Int. J.Math. Modelling & Numerical Optimisation, 2010 1, pp.330-343.

[9]. Kirkpatrick S., Gelatt C. D., and Vecchi M. P., Optimization by simulated annealing, Science, 1983 220, No. 4598, pp.671-680.

[10]. Deb K., Optimisation for Engineering Design: Algorithms and Examples, Prentice-Hall, New Delhi, 1995.

[11] Kulkarni K, Balasubrahmanyam K. Optimal design of open coiled helical springs. J Instrum Eng (India) 1979;60:7–14.

[12] Yokota T, Taguchi T, Gen M. A solution method for optimal weight design problem of helical spring using genetic algorithms, pp. 71–76 in Proceedings of the 21st International Conference. Comput Ind Eng 1997;33:1 2.

[13] Deb K, Goyal M. A flexible optimization procedure for mechanical component design based on genetic adaptive search. J Mech Des120(2), 1998. pp.162–4.

[14] Kannan BK, Kramer SN. An augmented Lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design. J Mech Des 116(2), 1994. pp.405–11.

[15] Sandregen E. Nonlinear integer and discrete programming in mechanical design optimization. J Mech Des 1990;112(2):223–9.

[16] Imaizumi T, Ohkouchi T, Ichikawa S. Shape optimization of the wire cross section of helical springs. SAE Trans 1992;36(4):507–14.

[17] Hernandez J. Ressorts Hélicoïdaux à Section Ovoïde. Transfil Europe 1997;40:22–8.

[18] Xiao Q, Liu L, Xiao Q. The optimal design and simulation of helical spring based on particle swarm optimization algorithm and MATLAB. Wseas Trans Circ Syst 2009;8(1):84–93.

[19] Philips J, Costello G. Large deflections of impacted helical spring. J Acoust Soc Am 1972;51:967–73.

[20] Stokes VK. B''On the dynamic radial expansion of helical springs due to longitudinal impact. J Sound Vib 1974;35(1):77–99.

[21] Mottershead JE. The large displacements and dynamic stability of springs using helical finite elements. Int J Mech Sci 1982;24(9):547–58.

[22] Yilidirim V. An efficient numerical method for predicting the natural frequencies of cylindrical helical springs. Int J Mech Sci 1999;41(8):919–39.

[23] Forrester MK. Stiffness model of a die spring. Thesis of Master of Science, Virginia; 2001.

[24] Taktak M, Dammak F, Abid S, Haddar M. A finite element for dynamic analysis of a cylindrical isotropic helical spring. J Mech Mater Struct 2008;3(4):641–58.

[25] Dammak F, Taktak M, Abid S, Dhieb A, Haddar M. Finite element method for the stress analysis of isotropic cylindrical helical spring. European J Mech A/ Solid 2005;24(6):1068–78.

[26] Taktak M, Dammak F, Abid S, Haddar M. A mixed-hybrid finite element for three-dimensional isotropic helical beam analysis. Int J Mech Sci 2005;47(2):209–29.

[27] Stone RE, Tovey CA. Erratum: The simplex and projective scaling algorithms as iteratively reweighted least squares methods. SIAM Rev 1991;33(3). pp. 461– 61.

[28] Dahtt G, Touzout G. Une presentation de la method des elements finis. 2nd ed. Paris: Maloine; 1984.

[29] Mohamed Taktak, Khalifa Omheni, Abdessattar Aloui, Fakhreddine Dammak, Mohamed Haddar: Dynamic optimization design of a

cylindrical helical spring. Applied Acoustics 77, 2014. pp. 178– 183.

[30] M.R. Kang, A. Kahraman: An experimental and theoretical study of the dynamic behavior of double-helical gear sets. Journal of Sound and Vibration.Volume 350, 18 August 2015, Pages 11–29.

[31] Fatih Karpat, Stephen Ekwaro-Osire, Kadir Cavdar, Fatih C. Babalik: Dynamic analysis of involute spur gears with asymmetric teeth. International Journal of Mechanical Sciences, Volume 50, Issue 12, December 2008, Pages 1598-1610

[32] Zheng Feng Bai, Yang Zhao: Dynamic behaviour analysis of planar mechanical systems with clearance in revolute joints using a new hybrid contact force model, International Journal of Mechanical Sciences, Volume 54, Issue 1, January 2012, Pages 190-205

[33] Letícia Fleck Fadel Miguel, Leandro Fleck Fadel Miguel: A firefly algorithm for the design of force and placement of friction dampers for control of man-induced vibrations in footbridges , Optimization and Engineering, (2014) DOI 10 1007/s11081-014-9269-3