THE OPTIMAL DESIGN AND SIMULATION OF HELICAL SPRING BASED ON PARTICLE SWARM ALGORITHM AND MATLAB

XIAO QIMIN[1], LIU LIWEI[2], XIAO QILI[3]
[1]Fundamental Department, The First Aviation College of Air Force
Xinyang, Henan, 464000 P. R. China
[2]Physical Department, Xinyang Normal University
Xinyang, Henan, 464000 P. R. China
[3] School of Computer and Information, Zhejiang Wanli University
Ningbo, Zhejiang, 315140 P. R. China

Abstract: - Optimal problem is often met in engineering practice. The method to solve complex optimal problem is always studied by people. Springs are important mechanical members which are often used in machines to exert force, to provide flexibility, and to store or absorb energy. Helical spring is the most popular type of springs. The method of helical spring optimization is a typical one which can be used to solving other mechanical optimal design problem. Particle Swarm Optimization algorithm is a good method in solving optimal problem. MATLAB is a high-performance language for technical computing and is an easy tool for us to simulate the optimization. In this paper, we mainly introduce the optimization of helical spring based on particle swarm algorithms and simulation in MATLAB. Directed by the theory of Particle Swarm Optimization algorithm, with the minimum weight of helical spring as objective function, with d, D_2 and n as design variables, with shear stress, maximum axial deflection, critical frequency, bucking, fatigue strength, coils not touch, space and dimension as constraint conditions, the complex helical spring optimal design mathematics model with three design variables and fourteen inequality constraints conditions is established. When the model is simulated in MATLAB the minimal optimal value of variables and the minimal weight of helical spring can be obtained. Simulating Result shows that Particle Swarm Optimization is practical in solving complicated optimal design problems and effectively on avoiding constraint of solution. The fundamental idea, the method of establishing mathematic model, the simulation process in MATLAB of helical spring can be used for reference to other similar mechanical optimal design.

Key-Words: - Particle Swarm Optimization (PSO), fitness value, local best value, global best value, helical spring, optimal design, mathematic model , objective function, design variables, constraints condition, shear stress, deflection, critical frequency, bucking, fatigue strength.

1 Introduction

Computer Aided Design (CAD) is widely used in engineering practice particularly in mechanical design, analysis, optimize and drawing [1]. The automation design of component [2] and machine [3] can be realized by applying CAD. Optimal design problem is often met in industry design. Many optimal algorithms are promoted to solve optimal problem. Particle Swarm Optimization (PSO) is a population based stochastic optimization technique, inspired by social behavior of bird flocking or fish schooling [4]. This algorithm with characteristics of easy realization, high precision and rapid convergence arouses the attention of academics and displays the superiority in solving practical optimal problem especially a complex one. This algorithm is widely used in mechanical optimization. MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include Math and computation Algorithm development Data acquisition Modeling, simulation. Simulated the mathematic model of mechanical element or machines in MATLAB, the fitness optimal value of design variables and objective function can be easily obtained. In this paper, we will focus on the optimal design of helical spring based on PSO and its simulation in MATLAB. Section 2 outlines fundamental idea of PSO searching for optima and its realization steps. Section 3 established the optimal design mathematical model of helical spring directed by theory of PSO algorithm. In Section 4 the simulation process and simulation results and analysis are given. Finally, conclusions are given.

2 PSO and Its Realization Steps

PSO is a population based stochastic optimization technique developed by Dr. Eberhart and Dr. Kennedy in 1995, inspired by social behaviour of bird flocking or fish schooling. This algorithm with characteristics of easy realization, high precision and rapid convergence arouses the attention of academics and displays the superiority in solving practical optimal problem especially complex optimal problem.

2.1 The Theory of Particles Search for Optima

The foundation idea [4] of PSO is as following. In PSO, each single solution is called particle. Each particle has a velocity and a position that are evaluated by the fitness function to be optimized. Each particle searches for optima by updating its velocity and position. The detail searching process is as following. PSO is initialized with a group of random particles and then searches for optima by updating generations. In each iteration each particle is updated by following two best values. The first one is the fitness it has achieved so far. This value is called p_{best}. Another best value that is tracked by the particle swarm optimizer is the best value obtained so far by any particle in the population. This best value is a global best and called g_{best}. When finding the two best values, particles update its velocity and position. In n dimensional space, if $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})$ represents the position of the i-th particle, $v_i = (v_{i1}, v_{i2}, \ldots, v_{in})$ represents the velocity of the i-th particle, $p_{\text{best}i} = (p_{i1}, p_{i2}, \ldots, p_{in})$ represents the fitness value of the i-th particle, $g_{\text{best}i} = (g_{i1}, g_{i2}, \ldots, g_{in})$ represents the global best value of whole particle swarm, then the velocity and particle position of the particle are updated with following equation (1) and equation (2) [5].

$$v_{id}(k+1)=wv_{id}(k)+c_1r_1(p_{\text{best}}(k)-x_{id}(k)) + c_2r_2(g_{\text{best}}(k)-x_{id}(k)) \quad (1)$$

$$x_{id}(k\text{-}1)= x_{id}(k) + v_{id}(k+1) \quad (2)$$

where $i=1,2,\ldots,m$(m is the number of particle in particle swarm), $d=1,2,\ldots,n$(n is the dimension of solution), c_1 and c_2 are learning factors, r_1 and r_2 are random number between (0, 1), w is inertia weight, used to control the effect on velocity of the older generation to the next generation. k is the number of iterations [6].

The iteration process should be gradually narrowed in order to ensure the particles gradually convergence closely to the optimal solution and not oscillation back and forth in solution space.

To solve this question, inertia weight should be obtained from equation (3) or equation (4),

$$w(k+1)=\mathrm{a}\cdot w(k) \quad \mathrm{a}\in(0,1) \quad (3)$$

$$w(k)= w_{\max}-\frac{w_{\max}-w_{\min}}{k_{\max}}\cdot k \quad (4)$$

where $w_{\max}$ and $w_{\min}$ are the maximum and minimum of inertia weight, $k_{\max}$ is the maximum number of iteration.

Particles update and iterate its position and velocity by equation (1) to (4) until get an optimum or achieve to the specified largest number of iteration.

2.2 PSO Parameter Control

There are two key steps when applying PSO to optimization problems, the representation of the solution and the fitness function. One of the advantages of PSO is that PSO take real numbers as particles. For example, we try to find the solution for $f(x) = x_1^2 + x_2^2 + x_3^2$, the particle can be set as (x_1, x_2, x_3), and fitness function is $f(x)$. Then we can use the standard procedure to find the optimum. The searching is a repeat process, and the stop criteria are that the maximum iteration number is reached or the minimum error condition is satisfied.

There are not many parameter need to be tuned in PSO. Here is a list of the parameters and their typical values.

The number of particles, the typical range is about 20 to 40. Actually for most of the problems 10 particles is large enough to get good results. For some difficult or special problems, one can try 100 or 200 particles as well.

Dimension of particles, it is determined by the problem to be optimized.

Range of particles, it is also determined by the problem to be optimized, we can specify different ranges for different dimension of particles.

Learning factors, c1 and c2 usually equal to 2.

The stop condition, the maximum number of iterations the PSO execute and the minimum error requirement.

2.3 The Implement Steps of PSO

As stated in above, the implement steps [7] of global particle swarm optimization are as following.

Firstly, initialize particle position xi, particle velocity vi, the number of iterations k, the dimension of solution n.

Secondly, calculate the current fitness value xi and pi for each particle.

Thirdly, compare the fitness value pi in step two with best fitness value pbesti. If pi < pbesti, then set current value as the new best fitness value and save the position, that is pbesti= pi and xbesti= xi.

Fourthly, compare particle with best fitness value pbesti to all particles with best fitness value gbesti. If pbesti < gbesti, then the particle with the best fitness value of all particle as gbesti, and save the current particle position, that is gbesti = pbesti, xbest= xbesti.

Fifthly calculate particle velocity according equation (1) and update particle position according equation (2). When the calculation mentioned above finished, returns to step two, and then new particle produced. The calculation is ended while the specified number of iteration is achieved. The flow chart [7] of global particle swarm optimization algorithms can be shown in Fig.1.

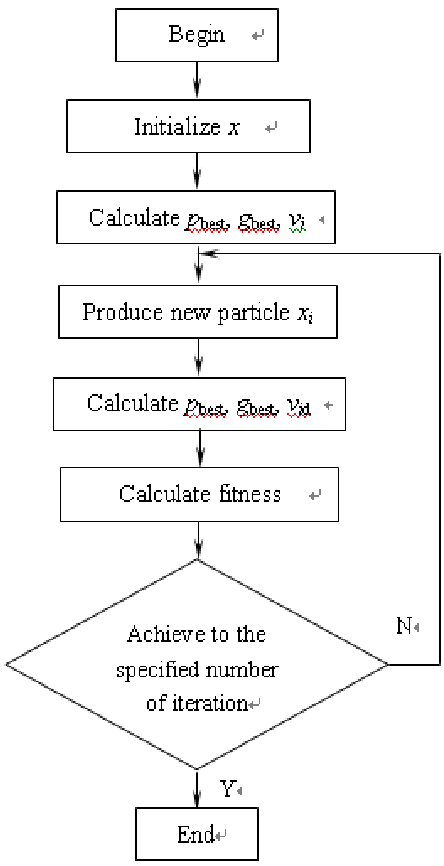

Fig.1 Flow chart of particle swarm algorithm

2.4 The Feature of PSO

In the process of iteration, being only one current optimal position needed to be returned, PSO algorithm is simple. The structure of the PSO algorithms is relatively simple and the calculate speed is fast. Just being the same reason, once the current optimal position is returned, the other particles will move closely to it quickly. If this optimal position is a local optimal position, PSO will not be able to re-search in the solution space. So PSO algorithm can easily fall into local optimum. This emergence phenomenon is called pre-mature phenomenon.

In order to solve the pre-mature phenomenon, two measures are often adopted. First, randomly initialize the particles in accordance with a certain laws to increase the diversity of particles. Second, search for variation to the best particle. In this two-pronged approach to increase the global search capability, to overcome the shortcomings of convergence the local extreme value, but also to ensure the convergence speed and accuracy of search.

3 The Optimal Design of Helical Spring Based on PSO

Springs are mechanical members, which are designed to give a relatively large amount of elastic deflection under the action of an externally applied load. Springs are used for a variety of purposes. The following is a list of the important purposes and applications of springs [8].

Controlling of motion in machines. This category represents the majority of spring applications such as operating forces in clutches and brakes.

Reduction of transmitted forces as a result of impact or shock loading. Applications here include locomotive or automotive suspension system springs and bumper springs.

Storage of energy. Applications in this category are found in clocks and movie cameras having recoil starters.

Measurement of forces. Scales used to weigh people is a very common application for this category.

In general, springs may be classified as either wire springs, flat springs, or special-shaped springs, and there are variations within these divisions. Wire springs include helical springs of round or square wire and are made to resist tensile, compressive, or torsion loads.

3.1 The Aim of Helical Spring Optimization

The optimal design of a helical spring involves the following consideration.

Aim of helical spring optimal design. To design the helical spring with minimum weight is usually the aim of optimization. So we must establish the objective function.

Design variables which determine the weight of helical spring. The selection of these variables will be stated following.

Conditions which constraint the realization of helical spring optimal design. There are many conditions which effect on the objective function, involving space into which the spring must fit and operation, values of working forces and deflections, accuracy and reliability needed, tolerances and permissible variations in specifications environmental conditions such as a corrosive atmosphere, cost and quantities needed.

The designer uses these factors to select a material and specify suitable values for the wire size, the number of coil, the diameter and free height, the type of ends, and the spring rate needed to satisfy the working force-deflection requirements. Because of the large number of interdependent variables, the problem is not as simple as inserting numbers into a procedure that will produce a complete set of results. Because of this fact, there are a rather large number of researchers who have attempted to simplify the spring-design problem by the use of charts and monographs.

The weight of a helical spring is an important measure of quality. When spring is designed for minimal weight, the weight can be defined as objective function. The weight of the helical spring can be obtained from equation (5) [9].

$$W= (n+2)\pi D2\,(\pi d2\rho/4)=0.25\rho\pi 2\ d\ 2\ D2\ (n+2) \quad (5)$$

where W is the weight of spring, n is the number of active coils, D2 is mean spring diameter, d is wire diameter, ρ is the density of spring wire materials. For steel-made spring ρ=7.8×10—6kg/mm3. So the weight of spring can be finally expressed with equation (6).

$$W=1.92\times10^{-5}\, d^{2} D_2\,(n+2) \tag{6}$$

3.2 Specify design variables

The weight of spring is determined by d, D_2 and n. So set d, D_2 and n as design variables. Design variables can be stated as equation (7).

$$x= (x_1, x_2, x_3)^{\mathrm{T}} = (d, D_2, n)^{\mathrm{T}} \tag{7}$$

3.3 Establish Objective Function

The objective function can be stated as equation (8).

$$\begin{aligned} F(x)&=1.92\times10^{-5} d^{2} D_2\,(n+2) \\ &=1.92\times10^{-5}\, x_1^2 x_2\,(x_3+2) \end{aligned} \tag{8}$$

3.4 Specify Constraints Conditions

3.4.1 Condition of Shear Stress

When a helical spring is loaded by the axial force F, shear stress is exerted in the spring wire. By using superposition, the shear stress in the inside fiber of the spring may be computed using the equation (9).

$$\tau =8FD_2/\pi d^{3} +4F/\pi d^{2} \tag{9}$$

If define spring index C=D2/d as a measure of coil curvature. With this relation, equation (9) can be arranged to give as equation (10) shows.

$$\tau = 8FD_2(1+0.5/C)/\pi d^3 \tag{10}$$

Or designating K_s=1+0.5/C, then

$$\tau = K_s \frac{8FD_2}{\pi d^3} \tag{11}$$

where K_s is called a shear multiplication factor. For most springs, C will range from about 6 to 12. Equation (11) is quite general and applies for both static and dynamic loads.

The stress equation can be presented as equation (12).

$$\tau = K\frac{8FD_2}{\pi d^3} \tag{12}$$

where K is called the Wahl correction factor. This factor includes the direction shear, together with another effect due to curvature. Curvature of the wire increases the stress on the inside of the spring but decreases it only slightly on the outside. The value of K may be obtained from equation (13).

$$K=(4C-1)/(4C-4)+0.615/C \tag{13}$$

or

$$K \approx 1.66(d/D_2)^{0.16}\ [10] \tag{14}$$

The maximum shear stress τ_{max} should not larger than allowable shear stress $[\tau]$. That is

$$\tau_{max}=8KF_{max}D_2/\pi d^3 \le[\tau] \tag{15}$$

F_{max} is the maximum axial load. So the condition of shear stress can be expressed as equation (16).

$$g_1(x)= -[\tau]+4.23F_{max}\ x_2^{0.84}\big/x_1^{2.84} \le 0 \tag{16}$$

Materials for springs should have high elastic properties, which are stable with time. It is poor practice to make springs of low-strength materials. The mass of geometrically similar springs at a given load and elastic deflection is inversely proportional to the allowable stress. The is due to the fact that springs of weaker materials must be made with larger diameters to retain the specified rigidity and, consequently, the coils are subject to higher twisting moments than those of springs of stronger materials. The effectiveness of using high-strength materials is also associated with the lower stress concentration in springs than in other machine components and the smaller size of the cross section of the coils.

A great variety of spring materials are available to the designer, including plain carbon steels, alloy steels, and corrosion-resisting steels, as well as nonferrous materials such as phosphor bronze, spring brass, beryllium copper, and various nickel alloys [11]. Table 1 provides spring materials and allowable stress.

Table 1 Spring materials and allowable stress

Common name, Specification	Allowable shear stress [τ] (Mpa)		
	I	П	Ш
Carbon steel wires, B,C,D grade	$0.3\sigma_B$	$0.4\sigma_B$	$0.5\sigma_B$
65Mn			
60Si2Mn 60Si2MnA	480	640	800
50CrVA	450	600	750

The common materials for springs are as follows.

High-carbon steels. Being less expensive, high-carbon steels are widely used for springs from 0.8 to 14mm in diameter. Not for use above 130 °C or at subzero temperature.

Manganese steel. Manganese steel has higher mechanical properties and better harden-ability, which enable them to be applied for springs from 2 to 12 mm in diameter.

Chrome vanadium steel. Chrome vanadium steel has higher mechanical properties and an especially high endurance limit, high heat resistance and good processing properties. Also used for shock and impact loads. Widely used for aircraft engine valve springs and for devices enduring temperatures to 220 °C. Available in annealed size 0.8 to 12mm in diameter.

Chrome silicon steel. This alloy is an excellent material for highly stressed springs requiring a long life and subjected to shock loading. Rockwell hardness of C50 to C53 are quite common, and the material may be used up at 250 °C, be available from 0.8 to 12mm in diameter.

Tensile strength of carbon steel spring wires is given in table 2.

Tensile strength of 65Mn spring wires [12] is given in table 3.

Table 2 Ultimate tensile strength of carbon steel spring wires σ_B (Mpa)

Carbon steel spring wire			
Wire diameter d (mm)	Class of wire		
	B	C	D
1.00	1660~2010	1960~2360	2300~2690
1.20	1620~1960	1910~2250	2250~2550
1.40	1620~1910	1860~2210	2150~2450
1.60	1570~1860	1810~2160	2110~2400
1.80	1520~1810	1760~2110	2010~2300
2.00	1470~1760	1710~2010	1910~2200
2.20	1420~1710	1660~1960	1810~2110
2.50	1420~1710	1660~1960	1760~2060
2.80	1370~1670	1620~1910	1710~2010
3.00	1370~1670	1620~1910	1710~2010
3.20	1370~1670	1570~1860	1710~1960
3.50	1320~1620	1570~1810	1660~1910
4.00	1320~1620	1520~1760	1620~1860
4.50	1320~1570	1520~1760	1620~1860

Table 3 Ultimate tensile strength of 65Mn spring wires σ_B (Mpa)

Wire diameter d (mm)	σ_B
1~1.2	1800
1.4~1.6	1750
1.8~2	1700
2.2~2.5	1650
2.8~3.4	1600

3.4.2 Condition of maximum axial deflection

When a helical spring is loaded by the axial force F, deflection is also exerted in spring. To obtain the equation for the deflection of a helical spring, we shall consider an element of wire formed by two adjacent cross sections. 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 F and after deformation it will rotate through the angle λ and occupy the new position. From the expression of Hooke's law for torsion, we have equation (17).

$$\lambda=8FD_2/\pi d^3G \tag{17}$$

The load F has a moment arm of $D_2/2$, and so the deflection is as equation (18) shows.

$$\lambda=8F D_2^3 n/ d^4G \tag{18}$$

where n is the number of active coils of spring, $F=\lambda Gd^4/8nD_2^3$, where G is shear modulus of elasticity[8], for steel-made spring $G=8\times10^4$Mpa, for bronze-made spring $G=4\times10^4$Mpa.

To ensure the axial deflection is smaller than the maximum deflection, the maximum working load which lead to maximum axial deflection should larger than the working force, that is Fmax≥F. So the condition of maximum axial deflection can be expressed as equation (19).

$$g_2(x)=\lambda G\ x_1^4/8\ x_2^3 x_3 - F_{\max} \le 0 \tag{19}$$

3.4.3 Condition of critical frequency

If a wave is created by a disturbance at one end of a swimming pool, this wave 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. So spring manufactures have taken slow-motion movies of automotive valve-spring surge.

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 [8].

The governing equation for a spring placed between two flat and parallel plates is the classical wave equation and is

$$\frac{\partial^2 u}{\partial y^2} = \frac{m}{kl^2}\frac{\partial^2 u}{\partial t^2} \tag{20}$$

where k is spring constant, l is the length of spring between plates, m is the mass of spring, y is the coordinate along length of spring, u is motion of any particle at distance y.

The solution to this equation is well known. The natural frequency in radians per second turns out to be

$$f_b = \frac{1}{2}\sqrt{\frac{k}{m}} = \frac{d}{2\pi n D_2^2}\sqrt{\frac{G}{2\rho}} \tag{21}$$

Where k is spring constant, $k = Gd^4 / 8D_2^3 n$, ρ is the density of spring wire materials.

If the spring has one end against a flat plate and the other end free, the frequency is

$$f_b = 0.25\sqrt{k/m} \tag{22}$$

This equation also applies when one end is against a flat plate and the other end is driven with a sine-wave motion.

For a spring with fixed ends, fb=3.56×105d/n D_2^2 . For a spring with one fixed end, one pinned end fb=1.78×105d/n D_2^2 [13].

The fundamental critical frequency fb should be from 15 to 20 times the frequency of the force or motion of the spring fw to avoid resonance with the harmonics. For fb=3.56×105d/n D_2^2, so condition of critical frequency can be expressed as equation (23).

$$g_3(x)=15 f_w - 3.56\times10^5 x_1 / \ x_2^2\ x_3 \le 0 \tag{23}$$

3.4.4 Condition of buckling

A compression spring whose free length H_0 is more than four times its mean diameter D_2 should be checked for buckling. If the spring is properly guided, such as inside a tube over a bar, the amount of buckling can be greatly reduced.

Define ratio of the height H0 to diameter D2, b= H0/ D2. The critical ratio bc can be obtained like this. For the springs which has the squared and ground ends are on rigid parallel surfaces perpendicular to the spring's axis, bc=5.3. For the springs which has one end on a rigid surface, one end hinged, bc=3.7. For the springs which has both surface on which the ends rest are hinged on a pin, bc =2.6 [14].

When b is greater than the above values, buckling should be checked and satisfied with

$$b= H_0/ D_2 < b_c \tag{24}$$

For helical spring, b can be obtained from equation (25).

$$b= H_0/ D_2=0.5n+1.5(d/D_2) \tag{25}$$

So condition of buckling can be expressed as equation (26).

$$g_4(x)=1.5x_1/x_2 +0.5x_3 - b_c \le 0 \tag{26}$$

3.4.5 Condition of fatigue strength

Springs are made to be used, and consequently they are always subject to fatigue loading. In many instances 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.

In the case of shafts and many other machine members, fatigue loading in the form of completely reversed stress is quite ordinary.

Springs subject to variable stresses with a large number of cycles (for instance, number of cycles of the stress N≥103) should be checked for fatigue strength and static strength. When the springs subject to the load that alternates from Fmin to Fmax, the stresses for springs can be obtained from equation (27) and equation (28) [13].

$$\tau_{\min} = K\frac{8F_{\min}D_2}{\pi d^3} \quad (27)$$

$$\tau_{\max} = K\frac{8F_{\max}D_2}{\pi d^3} \quad (28)$$

The fatigue strength condition is

$$S= (\tau_0+0.75\tau_{\min}) /\tau_{\max}\geqslant[S] \quad (29)$$

where τ_0 is endurance limit of the spring for a zero-plus cycle, which can be found in table 4. [S] is allowable factor of safety, if accuracy of design calculation and mechanical properties is higher, then [S]=1.3~1.7, if lower, [S]=1.8~2 [14].

Table 4 Endurance limit of the spring for a zero-plus cycle

Number of cycles of the stress N	10^4	10^5	10^6	10^7
τ_0	$0.45\sigma_B$	$0.35\sigma_B$	$0.33\sigma_B$	$0.3\sigma_B$

When the spring works under the condition with number of cycles of the stress N=10^6, then τ_0=0.33σ_b. So the condition of fatigue strength can be expressed as equation (30).

$$g_5(x)=[S]-\frac{0.33\sigma_B x_1^{2.84}}{4.23F_{\max}x_2^{0.84}}-0.75\frac{F_{\min}}{F_{\max}}\le 0 \quad (30)$$

3.4.6 Condition of coils not touch

When there is no load applied on a compression spring, the spring is free, and the height of it is H_0. As the load is applied, the coils move closer together, but do not touch. The maximum deflection $\lambda_{\max}$ exerts when the maximum load applied. In this case, to satisfy the requirement of coils not touch, following equation should be satisfied.

$$H_0-\lambda_{\max} \geq H_b \quad (31)$$

where H_0 free height, for closed and ground ends, H_0=nt +(1.5~2)d, for closed ends, not ground, H_0=nt +(3~3.5)d. t is pitch, t≈(0.28~0.5)D_2. When calculating, we usually set t=0.4 D_2. $\lambda_{\max}$ is the maximum deflection under the action of the load $F_{\max}$. H_b is solid height(the minimum operating height)，when the number of inactive turn n_2=2 and the springs have plain ends，H_b≈(n+1.5)d [15]. So the condition of coils not touch can be expressed by equation (32).

$$g_6(x)=8F_{\max}\, x_2^3 x_3/G\; x_1^4 +x_1x_3-0.4x_2x_3 \quad (32)$$

3.4.7 Condition of index

Springs are manufactured either by hot-working or cold-working processes, depending upon the size of the material, the spring index C=D_2/d where D_2 is mean spring diameter, d is wire diameter, and the properties desired.

Small-size springs of wire under about 8 to 10 mm in diameter are manufactured by cold coiling, larger size, by hot coiling.

Most cold-coiled springs are made of wire that has been heat-treated before coiling so that the springs are only tempered after being coiled. All hot-coiled springs and the most critical cold-coiled ones, in particular, those of most alloy steels, are hardened in the coiled condition. Wire for critical springs, made of high carbon steel with 1% carbon, is subjected to lead patenting, i.e. immersed after being heated to a high temperature in a molten lead bath. This increases grain size and the wire, following final drawing, is work-hardened to a high degree and strengthened. This also improves the elastic properties of the material.

Helical springs are produced in quantity on special automatic spring winding machines. In other cases the spring is wound around mandrel on a lathe.

For proper the spring index is of extreme importance. The smaller the index the more difficult it is to wind the spring. Table 5 provides the values of the common spring index C [16].

Compression springs are wound so as to allow a certain clearance between the coils. Extension springs have closely wound coils. To obtain closed coils the wire is stretched during the process of winding to subject it to tensile elastic strain. When the wound spring is removed from the mandrel there occurs an elastic spring-back of the material, the spring expands in diameter and the coils are so tightly pressed against each other that an initial tension is given to the spring increasing its carrying capacity.

Table 5 Common spring index C

d (mm)	C
0.2~0.4	7~14
0.5~1	5~12
1.5~2.2	5~10
2.5~6	4~9
7~16	4~8
18~50	4~6

Springs for static or limit short-term action, as well as springs subject to variable stresses with a cycle factor close to unity, are subjected to plastic deformation, which is called pressing. This is done by holding them for 6~48 hours under the load of the same sign as the design working load, but one producing stress above the elastic limit. As a result, the external fibers have a permanent set. In the free state of the spring, the external fibers, interacting with the inner fibers, have residual stresses of a sign opposite to the working stresses. When such a spring is loaded, these stresses are subtracted from the working stresses, thereby increasing the load capacity of the spring by 20% to 25%. When we design a spring, the maximum spring index Cmax and minimum spring index Cmin must be specify, that is Cmin≤C≤Cmax. So the condition of index can be express as equation (33) and equation (34).

$$g_7(x)= C_{\min} - x_2/x_1 \le 0 \quad (33)$$

$$g_8(x)= x_2/x_1 - C_{\max} \le 0 \quad (34)$$

3.4.8 Condition of active coil

The helical spring is the most popular type of spring. There are three types of ends commonly used for compression 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 three types of ends used for compression springs are described as follows.

Squared and ground ends, YI type. This type of end enjoys the advantage of not becoming readily tangled during manufacturing. In the case, the load is transmitted in a perfectly axial direction.

Squared or closed ends, YΠ type. This type does not tend to become tangled during the manufacturing process as readily as the plain ends ground.

Plain ends ground, YШ type. This type is better than the plain ends. However, during storage, handling and assembly of spring, the ends tend to become readily tangled.

Regardless of the type of ends used, a partially dead or inactive coil nd exists at each end of the spring. This interactive coil must be subtracted from the actual total number nt of coil to find the number of active coils. The following is an approximate rule for finding the active number of coils [17].

Squared and ground ends, subtract a total of 2 turns, that is nd =2, n= nt−2.

Squared or closed ends, subtract a total of 1.5 turns, that is nd =1.5, n= nt−1.5.

Plain ends ground, subtract a total of 1 turns, that is nd =1, n= nt−1.

When we design a spring, the maximum number of coil nmax and minimum number of coil nmin must be specify, that is nmin≤n≤nmax. So the condition of active coil can be expressed as equation (35) and (36).

$$g_9(x)= n_{\min} - x_3 \le 0 \quad (35)$$

$$g_{10}(x)= x_3 - n_{\max} \le 0 \quad (36)$$

3.4.9 Condition of space and dimension

To satisfy the requirement of fit, operation and dimension specification, the wire diameter d, the outside spring diameter D must be severely limited[18], that is $d_{\min} \le d \le d_{\max}$, $D_{2\min} \le D_2 \le D_{2\max}$. So the conditions of space and dimension can be expressed as the following equations.

$$g_{11}(x)= d_{\min} - x_1 \le 0 \quad (37)$$

$$g_{12}(x)= x_1 - d_{\max} \le 0 \quad (38)$$

$$g_{13}(x)= D_{2\min} - x_2 \le 0 \quad (39)$$

$$g_{14}(x)= x_2 - D_{2\max} \le 0 \quad (40)$$

3.5 Mathematic model of helical spring optimal design

According to objective function and constraints conditions the optimal design mathematical model [19] of helical spring can be states as equation (41).

$$\begin{cases} \min F(x) = 1.92 \times 10^{-5} x_1^2 x_2 (x_3 + 2) \\ s.t. g_u(x) \le 0 \quad u = 1,2,3,\cdots,14 \\ x = (d, D_2, n)^T = (x_1, x_2, x_3)^T \end{cases} \quad (41)$$

The helical spring optimal design model is a complex one which has three design variables and fourteen inequality constraints conditions.

4 Simulation of Helical Spring Optimization in MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include Math and computation Algorithm development Data acquisition Modeling, simulation, and prototyping Data analysis, exploration, and visualization Scientific and engineering graphics Application development, including graphical user interface building MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows us to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or FORTRAN. So it is easy for us to use MATLAB to simulate the optimization of helical spring. In this paper, we simulate the optimization of helical spring with language C.

When simulating helical spring optimal design there are almost as many ways to create a spring-design program as there are programmers, and there is nothing unusual about the program presented here. It works. The program, which is for the design of compression springs, can be used as a starting point for creation of other programs. It contains of seven separate subroutines, all of which utilize the same memory location. The subroutines, which should be used in the order in which they are present, are as following. Enter and display the outside diameter. Enter and display the total number of coils. Enter and display the number of dead coils. Compute and display the number of active coils. Select a material and enter and display the exponent and coefficient. Enter and display the wire diameter. Compute and display the torsion yield strength. Enter and display the maximum torsion stress desired when the spring is closed solid. Compute and display the solid height, the free height, and the force required to compress the spring solid. Compute and display the spring constant. Enter and display any desired operating force F. Compute and display the corresponding values of the torsion stress and the spring deflection.

Separate subroutines are used in this program in order to avoid re-entering all the data when only a single parameter is to be changed. In this way it is easy to see the effect of the single change. For example, having run through the program once it may be desirable to try a different wire size. This can be done by entering the new wire size in subroutine 4 and proceeding from that point.

4.1 Example

Example, A compression spring with one end fixed and another end free, the spring material is 65Mn, maximum load F_{max}=40N, the minimum load F_{min}=0, the frequency of the force f_w=25Hz, the life of spring is 104h, the wire diameter d ranges about 1 to 4mm, the mean spring diameter D_2 ranges about 10 to 30mm, the number of active coil n ranges about 4.5 to 50, the spring index C is not smaller than 4, the working temperature is 50°C, the deflection is not smaller than 10mm. Design this helical spring for minimal weight.

4.2 Specify the parameters data

From the example above, values of parameters are known. F_{max}=40N, F_{min}=0, f_w=25Hz, d_{min}=1mm, d_{max}=4mm, n_{min}=4.5, n_{max}=50, D_{2min}=10mm, D_{2max}=30mm, C_{min}=4, C_{max}=12, λ_{min}=10mm.

As stated above, by consulting table 1 and table 2, we can get [τ]=350Mpa.To this spring, we can specify G=8 × 104Mpa, [S]=1.9, bc=3.7. So the constraints conditions can be obtained.

$$g_1(x)=-350+169.2\, x_2^{0.84}\big/x_1^{2.84}\le 0$$
$$g_2(x)=10^3\, x_1^4/8\, x_2^3\, x_3-40\le 0$$
$$g_3(x)=375-3.56\times 10^5 x_1/\ x_2^2\, x_3\le 0$$
$$g_4(x)=1.5x_1/x_2+0.5x_3-3.7\le 0$$
$$g_5(x)=1.9-2.07\, x_1^{2.84}\big/x_2^{0.84}\le 0$$
$$g_6(x)=4\times 10^{-3}\, x_2^3 x_3+x_1\, x_3-0.4\, x_2\, x_3\le 0$$
$$g_7(x)=4-x_3/x_1\le 0$$
$$g_8(x)=x_3/x_1-12\le 0$$
$$g_9(x)=4.5-x_3\le 0$$
$$g_{10}(x)=x_3-50\le 0$$
$$g_{11}(x)=1-x_1\le 0$$
$$g_{12}(x)=x_1-4\le 0$$
$$g_{13}(x)=10-x_2\le 0$$
$$g_{14}(x)=x_2-30\le 0$$

4.3 Write M file

Enter MATLAB, then write the M file [7] of objective function and constraints conditions and named as Spring.m.

```
function  [f,g]=Spring_Opti(x)
f=1.92*(10e−5)* x(1) ^2*x(2) *(x(3)+2) ;
g(1)= −350+169.2*x(2)^0.84/x(1)^2.84;
g(2)= 1*(10e+3)*x(1)^4/( x (2)^3* x(3)) −40;
g(3)= 375−3.56*10e5*x(1)/( x(2)^2 * x(3)) ;
g(4)=1.5* x(1)/ x(2)  +0.5* x(3) − 3.7;
g(5)=1.9 −2.07* x(1) ^2.84/ x(2) ^0.84;
g(6)=4*(10e−3)* x(2) ^3* x(3) / x(1) ^4 + x(1) *
        x(3)−0.4* x(2) * x(3) ;
g(7)= 4−x(2)/x(1) ;
g(8)= x(2)/x(1) −12;
g(9)= 4.5 − x(3) ;
g(10)= x(3) −50;
g(11)= 1−x(1) ;
g(12)= x(1) −4;
g(13)=10− x(3) ;
g(14)= x(3) −30;
x0=[ x1   x2  x3 ] ;
x=constr('Spring_Opti',x0)
```

4.4 Results and analysis

Write program by using MATLAB language for computing simulation of PSO, the result of simulation is obtained. They are shown as following.

$x = (x_1, x_2, x_3)^T = (1.645, 16.113, 4.501)^T$

$F(x) = 5.5\text{g}$

That is to say, when taken d=1.645, D2=16.113, n=4.501 respectively, the weight of helical spring is minimum and equals 5.5g. According to practice, these parameters can be round to the whole value or in accordance with the standard series.

As stated above, good optimization results can be obtained by using PSO.

5 Conclusion

Directed by theory of PSO algorithm the optimal design mathematical model of helical spring is established. This mathematical model with three design variables and fourteen inequality constraints conditions is a complex optimal design problem. When simulated in MATLAB, minimal weight of the helical spring can be obtained, and optimal values of variables can be obtained too. The result of simulation shows that by using PSO algorithm the weight of the helical spring can be greatly reduced, the design quality and efficiency can be improved greatly, PSO algorithms can redound the particle's capability to dap out of the Maximum trap effectively. PSO algorithm is a new method in solving the complex optimal design problem. The fundamental idea, simulation process and data-processing method of helical spring can be used for reference to other similar mechanical optimal design.

References:

[1] Ibrahim Zeid, *Theory and Practice*, McGraw-Hill, Inc., 1991.

[2] Mikell P.Groover, Emory W.Zimmers Jr., *Computer-Aided Design and Manufacturing*, Prentice-Hall Inc., 1984.

[3] C.B.Besant, *Computer-Aided Design and Manufacture*, Ellis Horwood Limitad, 1998.

[4] Eberhart, R. C. and Shi, Y, *Particle swarm optimization: developments, applications and resources*, Proc. congress on evolutionary computation 2001 IEEE service center, Piscataway, NJ., Seoul, Korea., 2001.

[5] Kennedy, J. and Eberhart R. C., *Particle Swarm Optimization*, Proceedings of the 1995 IEEE International Conference on Neural Networks, Piscataway, 1995.

[6] Shi Y., and Eberhart R. C., *Parameter selection in particle swarm optimization*, Evolutionary Programming VII, Proc. EP 98 pp, 591-600. Springer-Verlag, New York, 1998.

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

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

[9] Liang shangming, Yin guofu. *Modern Mechanical Design Optimization*, Chemical Industry Press, Beijing, 2005.

[10] Smith E.H., *Mechanical Engineer's Reference Book*, 12th edit., Butterworth-Heinemann, Oxford, 1998.

[11] Andrew D.Dimarogoneas, *Machine Design*, John Wiley & Sons,Inc,2001.

[12] Liu weixin, *Modern Mechanical Optimal Design*, Tsinghua university Press, Beijing, 2001.

[13] Kenneth S. Edwards Jr, Robert B. McKee, *Foundamentals of Mechanical Component Design*, McGraw-hill, 1991.

[14] G.M.Maitra, L.V.Prasad, *Machine Design*,TATA McGraw-Hill,1985.

[15] Robert L.Mott, *Machine Elements in Mechanical Design*, 3rd edit. , Prentice-Hall, Inc, Upper Saddle River, New Jersey, 1999.

[16] M. F. Spotts and T.E.Shoup. *Design of Machine Elements*. Prentice-Hall, 1992.

[17] G.M.Maitra, L.V. Prasad. *Mechanical Design*. TATA McGraw-Hill, 1985.

[18] Robert H.Creamer, *Machine Design*, Addison-Wesley Publishing Company, 1984.

[19] Mortenson M.E., *Geometric Modeling*, John Wiley & Sons, Inc, 1985.