Optimization of helical compression springs using simulated annealing and ant colony optimization

FARHAD KOLAHAN[1], AHMAD TAVAKOLI [2], SAEED SOHEILI, MAHDI ABACHIZADEH, GOLNAZ TADAYON
[1]Department of Mechanical Engineering
[2] Department of Management Science
Ferdowsi University of Mashhad
Mashhad
IRAN
kolahan@um.ac.ir

Abstract: In this paper, the optimization of compression helical springs is investigated. The design process involves determining wire material, end type of the spring and wire diameter. In this regards, important design constraints are taken into account. The objective is to minimize the figure of merit, which shows the relative cost of spring. Two heuristic algorithms, namely simulated annealing and ant colony optimization, are employed to optimize the spring. The results obtained by these methods are compared with each other, and with the exact solution of the problem.

Key-Words: helical compression spring, simulated annealing, ant colony optimization

1 Introduction

The optimization of structures and mechanical systems has been investigated during last decades. Optimization techniques can be applied to various types of problems such as finding the optimum frequency and modal responses of dynamical systems [1], minimal weight stress [2], cost, volume or any other mechanical and structural objectives. They can be employed when the objective function is stated in closed form, that is, there are linear or nonlinear relationships between the design variables. Since there are several local optima in these functions, the optimization technique should be capable of escaping from local maxima or minima and finding the global optimum. The common classical methods of optimization usually fail to find the global optimum, therefore; the heuristic methods have been considered and improved during the past three decades. The heuristic algorithms, such as Genetic Algorithm (GA), Simulated Annealing (SA) and Ant Colony Optimization (ACO) are inspired by natural systems in physics and biology. They can process functions with any degrees of nonlinearities and discontinuities. They can also deal with discrete, continuous and mixed variables.

Simulated annealing (SA) was first proposed by Metropolis et al. [3]. Based on the physical process of annealing, a succession of options is assumed to decrease their configuration energy by decreasing temperature, and form a low energy crystal.

Since its proposal, many attempts have been made to improve SA algorithm, and to apply it on various types of problems. Szu and Hartley [4] proposed the fast simulated annealing (FSA), which uses a Cauchy probability density to provide a random process and permits large steps rather than the Gaussian probability used by the canonical simulated annealing (CSA). This method provides an inverse linear cooling rate rather than the inverse logarithmic cooling rate. On the other hand, the SA algorithm proposed by Kiselyov et al. [5] focuses on the improvement of the quality of the solution rather than the reduction of the annealing time.

One of the first attempts to employ the simulated annealing algorithm in structural engineering problems performed in the late 1980s. Using discrete design parameters, Elperin [6] described the main ideas of Monte Carlo annealing algorithms for structural optimization. Later, Topping et al. [7] developed a search strategy by using the SA method for the optimization of planar truss structures with discrete and continuous variables.

Optimization using Ant Colony Optimization is relatively new. In the field of structural optimization, a few optimization works have been reported; mostly on simplified models. Camp et al. [8] studied the application of ACO for designing steel frames. Christodoulou [9] presented the optimal truss design using ACO. From few reported works it seems ACO is a challenging area and research on ACO is still on its early stages.

2 Problem Formulation

Helical Springs are among the most common flexible parts in mechanisms and machineries. They are widely employed in various machines to absorb shocks and to bring flexibility to the structure. They are also used to store potential energy of strain for releasing at a later time.

In this paper, the optimal design of a helical compression spring for static load service is investigated. The helical spring is expected to have certain displacement (compression) in response to a specified force. The compressed and free lengths should also be within a pre-specified range.

In order to design the spring, the design procedure presented by Shigley et al. [10] is employed. The design variables are as follows:

- Wire material: Six types of materials are proposed for the spring wire in [10] including music wire A228, hard-drawn wire A227, chrome-vanadium wire A232, chrome-silicon wire A401, stainless steel wire A313 and phosphor-bronze wire B159.
- End conditions: Four end types are available for helical springs: plain end, plain and ground end, squared end, squared and ground end.
- Wire diameter (*d*): Generally, it is set between 0.004 to 0.5 inches. However, there are some size limitations which depend on the wire material [10].

Other design parameters considered in this research include spring coefficient (*K*), spring index (*C*), number of active turns (N_a), free length of the spring (L_0) and safety factor (*SF*). In our computations, the range of these parameters is set as follows:

$$\begin{aligned}&4 \le C \le 12,\\ &3 \le N_a \le 15,\\ &SF \ge 1.2,\\ &L_0 \le (L_0)_{cr},\\ &\xi \ge 0.15\end{aligned} \qquad (1)$$

Where ξ is the fractional overrun to closure and to achieve linear behavior of the spring, it is assumed that $\xi = 0.15$ [10]. The objective is to design a spring with the highest figure of merit (fom) given by:

$$fom = (\text{relative material cost})\gamma\pi^2 d^2 N_t D / 4 \qquad (2)$$

where d and D denote wire and coil diameters respectively, γ is the density of spring material and N_t is the number of helices. This function is the representative of relative cost, and is used as the objective function for optimization.

3 Simulated Annealing (SA)

Simulated annealing is a powerful search technique for optimization of large scale problems, especially ones with global optimum hidden among many local optima [2, 11].

This method is based on the concept of metallurgical annealing of solids and metals. In annealing process, a molten metal with high temperature is slowly cooled until thermal mobility molecules can move about freely, and then a cooling process is performed until solidification. If the cooling is slow enough a perfect crystal is formed in which all the atoms are arranged in a low level lattice, and so the crystal reaches the minimum energy. As the metal cools, atoms may align in different directions. In this case, the whole regions of atoms should be reversed to escape this state of local optimum. The required energy is available as heat in the metal, and it depends on the current temperature of the system, given by the Boltzmann distribution. As the temperature is decreased, great changes become more difficult for the system. When the temperature approaches zero, movements become impossible and the state of the atoms is frozen. In this way, for the slowly cooled system, the atoms are arranged in a low energy state and produce a pure crystal. However, if the cooling process is performed too quickly, a polycrystalline or metastable amorphous is formed with higher energy state.

In simulated annealing optimization technique, the probability function is based on the Boltzmann distribution as follows:

$$P(E_r) = \frac{1}{Z(T)}\exp(-\frac{E_r - E_q}{k_b T}) \qquad (3)$$

In the above equation, *Z(T)* is the normalization factor, k_b is the Boltzmann constant, E_q denotes the system energy in a configuration *q* at time *t*, and E_r shows the energy in a new randomly generated configuration *r* at time *t+1*.

If $E_r - E_q < 0$, then the configuration *r* is accepted as the next configuration at time *t+1*. Otherwise, the acceptance of this new configuration is decided by the probability function. That is to say, the system state may be at a higher energy state with a certain probability. This acceptance rule for next configurations is referred to as the Metropolis Criterion. Simulated Annealing algorithm requires the following data:

- A definition of possible configurations, which represents the solution space.
- An objective function for minimization.

- A control parameter representing the temperature; and a cooling schedule that shows the rate of cooling.
- A generator of random changes of the system configuration that models the movement of system towards the lower energy state.

Generally, the SA algorithm can be stated as follows:

```
$T = T_0$;
while ($T > T_{freezing}$)
  do until (Equilibrium is reached)
      Alternation (State i to State j);
if ($\Delta E_{ij} < 0$) then
          accept update (State j);
      else
          r=random number [0, 1];
          if ($\exp(-\Delta E_{ij} / T) > r$) then
              accept update (State j);
          else
              refuse update (State i);
$T_n = T_n * T_f$;
```

In the above T_f is the temperature reduction rate dependent on the time or number of iterations, *n*, and is given by:

$$T_f = \frac{1}{\log(1+n)} \tag{4}$$

The algorithm consists of two main loops. The external loop controls the rate of temperature reduction. The internal loop provides the uphill movement possibilities. For the presented research the SA method is executed as the following:

First, the initial temperature (T_0), the final temperature (T_f), the coefficient of temperature decrease (*k*), and the total number of configurations (*N*) are set. Then, a set of feasible design variables (x_0) is selected and the internal loop starts with the creation of a new set of the design variables x_i. The corresponding value of the objective function $f(x_i)$ is computed. If the constraints are fulfilled, the penalty function $P(x_i)$ is set to zero; otherwise it is given sufficiently high value to force discarding the generated configuration. The function $F(x_i)$ is then evaluated as the sum of $f(x_i)$ and $P(x_i)$. The difference ΔF between the function $F(x_i)$ and $F(x_{i-1})$ is computed. The configuration is accepted, if it satisfies the acceptance criterion; otherwise it is discarded. The acceptance condition is given by:

$$\Delta F < 0 \quad or \quad rand[0,1] < \exp(-\Delta F / T) \tag{5}$$

In the first case the number of the accepted configurations *n* is updated and the internal loop is repeated until *n* equals *N*. Then, on the external loop, the temperature *T* is decreased and the procedure is repeated with the updated temperature. The external loop ends when $T = T_f$ is satisfied.

4 Ant Colony Optimization (ACO)

First proposed by Dorigo et al. [12], this algorithm is based on the nature of ants finding their paths by pheromone deposition. Ants usually prefer the path with more pheromone trail on it. Such a path is passed by more ants. In this way, the shorter paths are more desirable and have stronger pheromone trail, because it takes shorter time to march. Therefore, these paths are more frequently visited by ants. These behaviors are simulated by three rules in ACO, which can be best applied on TSP problem where it deals with finding the shortest tour. Regarding the nature of problem presented here, the definition of rules is to some extent different than those used in TSP-based formulation.

Regarding the new formulation, the design variables are presented by *i* and their divided search domains are shown by *j*. The sections of total solution are chosen in a constructive approach named as "state transition rule":

$$S = \begin{cases} \arg\max\left\{[\tau(i,j)].[\eta(i,j)]^{\beta}\right\} & if\ q \le q_0 \\ s & otherwise \end{cases} \tag{6}$$

Where $\tau(i,j)$ shows the amount of pheromone related to the *j*th element of variable *i*, and $\eta(i,j)$ is the heuristic function defined according to the problem. In this rule, *q* is a random number, and q_0 is a parameter set by the user ($0 \le q, q_0 \le 1$). If $q > q_0$, the next step is selected according to proportional distribution of probability function, like the roulette wheels, as follows:

$$s = \begin{cases} \dfrac{[\tau(i,j)].[\eta(i,j)]^{\beta}}{\sum\limits_{u \in allowed\, u}[\tau(i,u)].[\eta(i,u)]^{\beta}} & if\ j \in allowed\ j \\ 0 & otherwise \end{cases} \tag{7}$$

An important factor in this process is the amount of q_0 which defines the range of randomness and determination of state transition rule. It is clear that the higher amounts of q_0 directs the algorithm towards deterministic decisions, while the lower amounts of it generates more randomness.

To avoid stagnation of the algorithm and similar to evaporation of pheromone in real world, the amount of pheromone level is changed after finishing each evaluation by applying “the local updating rule”:

$$\tau(i,j) = (1-\rho).\tau(i,j) + \rho.\Delta\tau(i,j) \qquad (8)$$

In this equation, ρ denotes the local evaporation coefficient. According to Dorigo [12], the best performance is obtained where $\Delta\tau(i,j) = \tau_0$.

The third rule known as “the global updating rule” acts as a positive feedback and accumulates more pheromone around the best solution obtained so far:

$$\tau(i,j) = (1-\alpha).\tau(i,j) + \alpha.\Delta\tau \qquad (9)$$

Where

$$\Delta\tau = \frac{1}{f} \qquad (10)$$

and α is the global evaporation coefficient. The parameter f is the amount of objective function which is the value of *fom* here.

This process of next step evaluation and updating is repeated until the termination condition, which is usually the maximum number of cycles, is satisfied.

5 Numerical Results

The problem described in section 2 is solved with the help of ACO and SA algorithms. For illustrative purpose, we consider a helical spring which is expected to have 2 inches compression in response to 20 lb force. The compressed and free lengths should be less than 1 inch and 4 inches, respectively. The computational results are presented in Table 1. The global optimum is also obtained by direct enumeration to be the criterion for the efficiency of the algorithms.

It should be noted that the percentage of feasible solutions in this problem is much limited due to several constraint defined based on mechanical and manufacturing considerations. Therefore, the optimization algorithm should be powerful enough to direct the solution toward optimum and also feasible regions.

Regarding SA and its nature of the randomness, it is not able to jump over unfeasible zones and so the results are satisfactory only to some extent. Concerning ACO, as it is equipped with the pheromone data communication, the algorithm performs effectively in achieving much higher ratio of feasible solutions to total trials. It is also able to find the global optimum which validates the application of ACO in this type of problems.

In this procedure, the ACO employs 5 ants in 1000 iterations which are 5000 evaluations in total. The parameter setting, which is of great importance in ACO, is performed considering the nature of problem. As explained earlier, the feasible zone is strictly limited so the biased randomness should be intensified in order to provide more efficient searching of solution space. To this end, $q_0 = 0.5$ is applied that gives equal chance to deterministic and probabilistic rules. The decay parameters ρ and α are both set to 0.1, which is a common strategy in ACO applications. No heuristic function is used here, since defining them particularly in mechanical and structural problems is very intricate. Therefore, the algorithm relies on its pheromone mechanism and also the feedback by objective function evaluation and its effect on the amount of pheromone embedded in the global updating rule.

6 Conclusion

The problem of optimizing helical compression springs for minimum figure of merit (a cost index) was investigated. The design variables were the spring wire diameter, wire material and spring end condition. Simulated annealing and ant colony optimization were the two methods described and employed for the optimization process. The computational results were compared with the global optimum obtained by direct enumeration. Against the common definition of operators in ACO in problems like TSP, a modified pheromone communication was proposed. It was shown that, the results obtained by inserting these modifications in ACO method were of better quality in comparison to the SA. In summary, ACO performs better than SA in terms of solution quality as it reached the global optimum but SA failed to achieve it.

Table1. Optimization results for the helical compression spring

Optimization method	Best *fom*	Best wire diameter (in)	Best material	Best end condition
SA	0.3394	0.182	Phosphor-bronze B159	Plain
ACO	0.2254	0.097	Hard-drawn A227	Plain
Optimum Solution	0.2254	0.097	Hard-drawn A227	Plain

References:

[1] Masad, J.A., Optimization Methods with Structural Dynamics Applications, *Computers and Structures*, Vol. 62, 1996, pp. 521–525.

[2] Hwang, S.F., He, R.S., Improving Real-parameter Genetic Algorithm with Simulated Annealing for Engineering Problems, *Advances in Engineering Software*, Vol. 37, 2006, pp. 406-418.

[3] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller A., and Teller, E., Equation of State Calculation by Fast Computing Machines, *Journal of Chemical Physics*, Vol. 21, 1983, pp. 1087-1092.

[4] Szu, H., Hartley, R., Fast Simulated Annealing, *Physics Letters A*, Vol. 122, 1987, pp. 157-162.

[5] Kiselyov, B.S., Kulakov, N., Mikaelian, A.L., Modification of the Simulated Algorithm Method for Solving Combinatorial Optimization Problems, *Proceedings of SPIED Photonics for Computers, Neural Networks and Memories*, 1992, pp. 120-124.

[6] Elperin, T., Monte Carlo Structural Optimization in Discrete Variables with Annealing Algorithm, *International Journal for Numerical Methods in Engineering*, Vol. 26, 1988, pp. 815-821.

[7] Topping, B.H.V., Khan, A.I., Leite, J.P.B., Topological Design of Truss Structures Using Simulated Annealing, *Proceedings of CIVIL-COMP93, The Fifth International Conference on Civil and Structural Engineering Computing and Artificial Intelligence, Neural Networks and Combinatorial Optimization in Civil and Structural Engineering*, Civil-Comp Press, 1993, pp. 151-165.

[8] Camp, C.V., Bichon, B., Stovall, S.P., Design of Steel Frames Using Ant Colony Optimization, *Journal of Structural Engineering*, Vol. 131, 2005, pp. 369-379.

[9] Christodoulou, S., Optimal Truss Design Using Ant Colony Optimization, *5th GRACM International Congress on Computational Mechanics*, Limassol, Cyprus.

[10] Shigley, J.E., Mischke, C.R., Budynas, R.G., *Mechanical Engineering Design*, McGraw-Hill Co. Ltd., 2004.

[11] Topping, B.H.V., Leite, J.P.B., Parallel Simulated Annealing for Structural Optimization, *Computers and Structures*, Vol. 73, 1999, pp. 545-564.

[12] Dorigo, M., and Gambardella, L. M., Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem, *IEEE Transactions on Evolutionary Computation*, Vol.1, No.1, 1997, pp. 53-66.