Gazi Mühendislik Bilimleri Dergisi
2017, 3(2): 21-27

www.gazipublishing.net/index.php/gmbd

Optimum Design of Compression Spring According to Minimum Volume Using Grey Wolf Optimization Method

İsmail Şahin[a*], Murat Dörterler[a], Harun Gökçe[b]

[a] *Gazi University Tehnology Faculty, ANKARA 06560, TURKEY*
[b] *TÜBİTAK Defense Industries Research and Development Institute, ANKARA 06261, TURKEY*

ARTICLE INFO

Received: 15.04.2017
Accepted: 10.07.2017

Keywords:
Compression spring, grey wolf optimizer, optimimum design of machine elements

Corresponding Authors
e-mail:
isahin@gazi.edu.tr

ABSTRACT

Optimization of machine elements is both an important issue and an intensive study topic in engineering. Design of compression springs according to minimum weight or volume is a sample problem in this area. Various optimization methods such as particle swarm optimization, genetic algorithm are applied to the problem. Grey Wolf optimization (GWO) method, one of the least nature-inspired algorithms, mimics the hunting and leadership hierarchy of grey wolves. The method has attracted attention for a short time due to its successful performance in engineering applications. In this study, GWO was applied to the design of compression springs with minimum volume. The performance of the GWO was compared with the optimization methods used for solving the same problem in previous studies. The results of the study show that the GWO provides very successful results for the design of compression springs with minimum volume.

Bozkurt Optimizasyon Yöntemi Kullanarak Basınç Yaylarının Minimum Hacme Göre Optimum Tasarımı

MAKALE BİLGİSİ

Alınma: 15.04.2017
Kabul: 10.07.2017

Anahtar Kelimeler:
Basınç yayı, bozkurt optimizasyonu, makine elemanlarının optimum tasarımı

Sorumlu Yazar
e-mail:
isahin@gazi.edu.tr

ÖZET

Makine elemanlarının optimizasyonu mühendislikte hem önemli bir problem hemde yoğun bir çalışma alanıdır. Basınç yaylarının minimum hacme veya ağırlığa göre tasarımı bu alandaki örnek problemlerden birisidir. Parçacık sürü optimizasyonu, genetik algoritma gibi çeşitli optimizasyon yöntemleri bu probleme uygulanmıştır. Doğadan esinlenen algoritmaların sonuncularından Bozkurt Optimizasyonu (BO) yöntemi, bozkurtların avlanmaları ve liderlik hiyerarşisinden esinlenmiştir. Bu yöntem, mühendislik uygulamalarındaki başarılı performansıyla kısa sürede dikkatleri çekmiştir. Bu çalışmada BO, basınç yaylarının asgari hacme göre tasarımına uygulanmıştır. BO'nun performansı önceki çalışmalarda aynı problemin çözümü için kullanılan optimizasyon yöntemleriyle karşılaştırılmıştır. Çalışmanın sonuçları BO'nun basınç yaylarının asgari hacme göre tasarımında başarılı sonuçlar verdiğini göstermiştir.

1. Introduction

Engineering Design can be expressed as the activities conducted to create the product that meets the well-defined needs in physical solution space. When these activities are being carried out, there is a process continuously renewed. The designer, during the design, develops a preliminary design of the system based on experiences, perceptions or some basic mathematical analyses. Analyses are made to determine whether or not the preliminary design is acceptable.

Design optimization consists of specific objectives (objective functions), a search field (feasible solutions), a search process (optimization methods). Suitable solutions are a set of all designs that are characterized by all possible values of the design parameters (design variables) [1]. The optimization method searches for the optimum design out of all feasible designs. There are many applications where optimum design methods are useful in system design [2].

Mechanical design involves an optimization process that considers designers to always meet their specific objectives (strength, bending, weight, wear, corrosion, etc.) [1]. Most mechanistic optimal design problems are difficult to be solved with traditional optimization algorithms because they involve specific problem limitations [9]. In recent years, many new generation optimization algorithms have been applied in addition to traditional methods to solve mechanical design optimization problems. Evolutionary algorithms are general population based meta-heuristic optimization algorithms. Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Artificial Bee Colony (ABC) and Ant Colony Optimization (ACO) are among the popular evolutionary algorithms [1]. Evolutionary algorithms are preferred in optimization problems as they have better global search capabilities compared to traditional optimization algorithms [3].

Springs are one of the machine elements that have functions such as flexibility, energy absorption, vibration isolation and mitigating shock. When designing a general spring, trace and cut methods are generally used to determine factors such as load, deflection, number of active turns, and average diameter of spring wire [12]. Springs are structural elements designed to protect and store energy and mechanical work based on the principle of flexible deformation of the material. These are among the components of the heaviest loaded machines and are usually used as follows [11]:

- For energy absorbing and control and reversing devices,
- The inhibitors of static and dynamic forces,
- Elements for the formation of power joints,
- Shock absorbers with anti-vibration protection,
- Control and measuring devices

Springs are optimized according to the minimum volume or minimum weight requirements depending on where they are used. The problem of optimization of the springs with respect to the minimum volume is solved by different optimization methods considering different studies within the scope of nonlinear optimization problems. From these, Sandgren proposed non-linear branch and boundary algorithms based on integer programming [10]. Deb and Goyal used a combined genetic search technique (GeneAS) that combines binary and real-coded genetic algorithms [4]. He and his colleagues have tried to solve the problem with particle swarm optimization [9].

In this study, the optimal design of the compression spring with respect to the minimum volume was carried out by using Grey-Grey Wolf Optimization (GWO) method, which is a rather new meta-heuristic optimization method compared to other evolutionary algorithms. In the study, GWO is introduced and the performance of the GWO is tested in the design of the compression spring with respect to the minimum volume commonly used in mechanical design optimization. The optimization problem used for testing was first described by Sandgren [10]. The problem was then solved by different researchers using different optimization methods [9, 6, 4]. It was investigated whether an improvement was possible by comparing the previous works with the GWO method developed in the study.

2. Grey Wolf Optımızatıon Method

The Grey Grey Wolf Optimization (GWO) algorithm was developed by Mirjalili et al., inspired by the hierarchy and hunting method of the wild grey wolves [7]. In the herd hierarchy of grey wolves, individuals play one of the roles of alpha, beta, delta and omega. Alpha is the leader; the group follows its instructions. Alpha is not the most powerful member of the group but should be the best in terms of managing the group. Beta is the second most dominant individual in the herd. When following the Alpha, he rules the others. The third most dominant

individual in the herd is the Delta. Beta and delta must respect the individual in the positions above and should instruct other lower level individuals. Beta and Delta reinforce the commands of Alpha throughout the herd and give feedback to Alpha. Omega is the lowest grade grey wolf. He is the last resort to benefit from the prey. Omega may appear to be "not an important individual in the herd," but the absence of the omegas can cause various imbalances and issues in the herd.

In addition to the social hierarchy of grey wolves, group hunting is an interesting form of social behaviour of grey wolves [5]. Group hunting consists of the following steps:

- Tracing, tracking and approaching the prey
- Chase, siege and harass until the prey is immobilized
- Attack on the prey

GWO Algorithms

The main parts of the GWO consist of siege, hunting, and attacking the prey. Mirjalili has mathematically modelled this hunting technique and social hierarchy of grey wolves with the algorithm he has developed. The mathematical model consists of social hierarchy, chasing, siege, and attacking the prey.

Social Hierarchy

In order to mathematically model the social hierarchy of the grey wolves, alpha (α) has been considered as the most appropriate solution in the herd [8]. After the Alpha the second best solution is named Beta (β) and the third best solution is named Delta (δ). The remaining solutions are assumed to be Omega (ω). In the GWO algorithm hunting (optimization) is guided by α, β and δ. Delta (ω) wolves follow these three wolves. The hunted (prey) represents the optimum solution in the solution space.

Encircling

The grey wolf hunts his prey in a siege. The siege continues during hunting. In GWO, the behaviour of siege of the wolf is expressed by the following equations:

$$\vec{D} = \left|\vec{C} \,.\, \overrightarrow{X_p}(t) - \vec{X}(t)\right| \quad (1)$$

$$\vec{X}(t+1) = \overrightarrow{X_p}(t) - \vec{A}.\vec{D} \quad (2)$$

In the equation t, expresses the current iteration, $\vec{A}$ and $\vec{C}$ coefficient vectors, and $\overrightarrow{X_p}$ position vector of the prey.

Vectors are calculated in the below equations:

$$\vec{A} = 2\vec{a}.\overrightarrow{r_1} - \vec{a} \quad (3)$$

$$\vec{C} = 2.\overrightarrow{r_2} \quad (4)$$

Here, components of $\vec{a}$ is reduced linearly from *2* to *0* throughout the iteration. The vectors $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ are random vectors in the interval *[0,1]*. The results of the equations *1* and *2*, a two-dimensional position vector and some of the possible neighbours are shown in Figure 1. As can be seen in Figure 1, the grey wolf in the (*X, Y*) position can update the position (*X*, Y**) according to the position of its prey.

By setting the value of $\vec{A}$ and $\vec{C}$ vectors, different locations around the best unit relative to the current position can be reached. For example, (*X* -X, Y**) can be reached by setting the vectors $\vec{A}$ = (1,0) and $\overrightarrow{C}$ = (1,1). The random vectors $\overrightarrow{r_1}$ and $\overrightarrow{r_2}$ allow the wolves to reach any point between the points shown in Figure 1. Thus, a grey wolf can update its position around the predator in any random place with the help of equations [7].

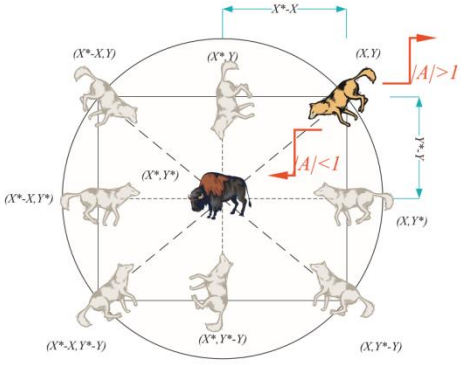

Figure 1: Position vectors with 2D and next possible positions [7]

Hunting

Grey wolves have the ability to recognize and surround the prey. However, the optimum solution to reach the location of the prey that is the hunt is not known in the solution space. For this reason, Alpha, Beta, and Delta, the three individuals with the best value in the herd, are assumed to be the closest individuals to the prey. The other members of the herd update their next positions according to the positions of these three individuals.

To mathematically simulate the hunting behaviour of the grey wolves it is assumed that Alpha (the best candidate solution), Beta and Delta have better knowledge of the potential location of the prey. For this reason, the first three best solutions obtained are recorded. The system forces other search agents to update their location based on the location of the best search agents. In this context, the following equations are used to update the positions of the individuals in the herd:

$$\overrightarrow{D_\alpha} = \left|\overrightarrow{C_1}.\overrightarrow{X_\alpha} - \vec{X}\right| \quad (5)$$

$$\overrightarrow{D_\beta} = \left|\overrightarrow{C_2}.\overrightarrow{X_\beta} - \vec{X}\right| \quad (6)$$

$$\overrightarrow{D_\delta} = \left|\overrightarrow{C_3}.\overrightarrow{X_\delta} - \vec{X}\right| \quad (7)$$

$$\overrightarrow{X_1} = \overrightarrow{X_\alpha} - \overrightarrow{A_1}.(\overrightarrow{D_\alpha}) \quad (8)$$

$$\overrightarrow{X_2} = \overrightarrow{X_\beta} - \overrightarrow{A_2}.(\overrightarrow{D_\beta}) \quad (9)$$

$$\overrightarrow{X_3} = \overrightarrow{X_\delta} - \overrightarrow{A_3}.(\overrightarrow{D_\delta}) \quad (10)$$

$$\vec{X}(t+1) = \frac{\overrightarrow{X_1}+\overrightarrow{X_2}+\overrightarrow{X_3}}{3} \quad (11)$$

Figure 2 shows how an Omega updates its position relative to Alpha, Beta and Delta in the 2D search space. The final location can be observed to be in a random location within a circle defined by the locations of Alpha, Beta, and Delta in the search area. In other words, Alpha, Beta and Delta estimate the position of the prey and other wolves randomly update their position around the prey.

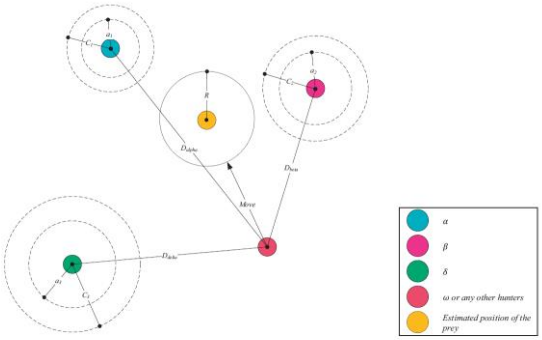

Figure 2: Location update in GWO [7]

Attacking

As mentioned above, when the prey stops, the wolves attack the prey and end the hunting. For the mathematical modelling of approaching the prey in the algorithm, the value of $\vec{a}$ is reduced. $\vec{A}$ is a random value in the range *[-a, a]*, where a is reduced from *2* to *0* during iterations. Whereas the random values of $\vec{A}$ can be *[-1, 1]* the next position of a search agent (grey wolf-search agent) can be any position between the current position and the position of the prey. As shown in Figure 1, when $|A|<1$, it forces wolves to attack their prey.

Exploration (Search for Prey)

Grey Wolves often search in accordance with the locations of Alpha, Beta and Delta. They separate to look for the prey and get closer to the prey. In order to mathematically model the separating state, random values smaller than or greater than 1 are used, which are defined as A to distinguish the search agent from the prey. This emphasizes the state of exploration and allows the GWO algorithm to be searched globally. $|A|>1$ situation shown in Figure 1 forces the grey wolves to leave the prey to find a more suitable prey.

To summarize, the search starts with creation of a random population of grey wolves (candidate solutions) in the GWO algorithm. Along the repeats, Alpha, Beta, and Delta wolves estimate the likely location of the prey. Each candidate solution updates the distance between the prey and the wolf. To emphasize exploration and operation, the parameter a is reduced from 2 to 0, respectively. Candidate solutions tend to leave the prey when $A_j>1$ and they approach the prey when $A_j<1$. Finally, the GWO algorithm is terminated by satisfying a termination criterion.

3. Application: Optimisation Of Compression Spring With Respect To Minimum Volume

In the study, the basic dimensions (d, D, N) of the compression spring with respect to the minimum volume are optimized. The problem was previously solved by Sandgren [10], He et al. [9], Lampinen and Zelinka [6] and Deb and Goyal [4] in different optimization methods. Three design variables have been defined for describing the compression spring problem. As shown in Figure 3, the spring wire diameter $d=X_1$, the average outer diameter $D=X_2$ and the number of active turns $N=X_3$.

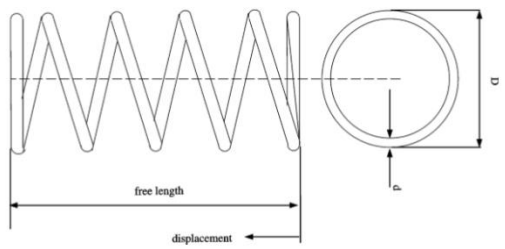

Figure 3: Comperssion Spring

The problem identified in the study is the real-world optimization problem involving discrete, integer and continuous design variables. The challenge is to minimize the volume of the spring compression under static load. There are three different design variants, D: continuous, *N* is an integer, and *d* is a variable with 42 possible values given in Table 1. The objective function of the problem is formulated as shown in equation 12.

$$f(x) = \frac{\pi^2 x_2 x_1^{\,2}(x_3 + 2)}{4} \qquad (12)$$

Other specifications of the problem are as follows: Maximum working load $F_{max} = 1000.0lb$; Maximum free length of spring $l_{max} = 14.0in$; Minimum wire diameter of spring is d_{min}; permissible maximum slip resistance $S = 189000.0psi$; Maximum outer diameter of the spring $D_{max.} = 3.0in$; Preload applied for compression $F_p = 300.0lb$; Maximum permissible deviation under preload $\sigma_{pm} = 6.0in$; Deviation from preload position to maximum load position $\sigma_w = 1.25in$; the shear modulus of the material is $G = 11.5 \times 106psi$.

Variables of the design (x_1, x_2, x_3) are limited in the below figure:

$0.2 \geq x_1 \geq 1, 0.6 \geq x_2 \geq 3, 1 \geq x_3 \geq 70$

4. Conclusion And Discussion

The problem has already been researched by Sandgren. Deb and Goyal apply the problem to genetic adaptive search (GeneAS), an intelligent method based on genetic algorithm. Lampinen and Zelinka have tried to solve the problem by differential evolution (DE) method. He and his colleagues investigated the problem with the particle swarm algorithm.

In this study, the problem was exploited by the codes used by Mirjalili's work [7] for solution with GWO. In the existing codes, the adaptation function according to Equation 12 and necessary adaptations have been made considering the constraints stated in Equation 13 and 20.

Performance studies were conducted with MATLAB 2016b software on a Windows 10 x64 machine with Intel (R) Core (TM) i7 3.3 GHz CPU and 8 GB of RAM.

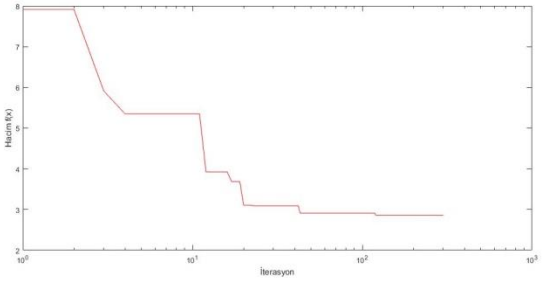

Figure 4: Performance curve of GWO

In this study, the number of populations was determined as 30 and the number of iterations (repeats) as 300. The algorithm was run 100 times and the best result obtained is presented in Table 2 in comparison with previous studies. As seen in Table 2, the optimum value obtained with GWO is almost the same as previous studies. However, as seen in the performance curve in Figure 4, the best value can be obtained at the end of 238 iterations in arithmetic mean with GWO. This means that the suitability function was called up to 7140 times in the average. This value is 15000 in the work of He et al. (2004) and 26000 in the works of Lampinen and Zelinka (1999). The arithmetic mean of best value for 100 runs is the 2.6902 and standard deviation is 0.0572. In terms of the runtime, the average CPU time of the algorithm is measured as 1281 ms.

In the study, GWO was applied for the first time to solve the problem of the design of springs with respect to the minimum volume. According to previous studies, the method produced better results. In optimizing springs with respect to minimum volume, the supremacy of GWO over previous studies proved itself in its speed. GWO can produce solutions about four times faster than previous studies. In very low iterations, it converges to the optimal solution. The speed, consistency and statistical performance of the GWO show that it is a method that can be used in the optimization of machine elements.

Table 1: Possible Spring Wire Diameters

Spring Wire Diameter (*d – in.*)						
0.009	0.0095	0.0104	0.0118	0.0128	0.0132	0.014
0.015	0.0162	0.0173	0.018	0.020	0.023	0.025
0.028	0.032	0.035	0.041	0.047	0.054	0.063
0.072	0.080	0.092	0.105	0.120	0.135	0.148
0.162	0.177	0.192	0.207	0.225	0.244	0.263
0.283	0.307	0.331	0.362	0.394	0.4375	0.500

Table 2: Optimum solutions for compression spring

Design Variables	Sandgren (1990)	Deb and Goyal (1997)	Lampinen and Zelinka (1999)	He et al. (2004)	GWO
$X_1(d)$	0.283	0.283	0.283	0.283	0.283
$X_2(D)$	1.180701	1.226	1.223041	1.223041	1.2230413690
$X_3(d)$	10	9	9	9	9
$g_1(x)$	-54309	-713.510	-1008.811	-1008.811	-1008.773
$g_2(x)$	-8.8187	-8.933	-8.9456	-8.9456	-8.9456
$g_3(x)$	-0.08298	-0.083	-0.083	-0.083	-0.083
$g_4(x)$	-1.8193	-1.491	-1.777	-1.777	-1.776
$g_5(x)$	-1.1723	-1.337	-1.3217	-1.3217	-1.3217
$g_6(x)$	-5.4643	-5.461	-5.4643	-5.4643	-5.4642
$g_7(x)$	0.0000	0.0000	0.0000	0.0000	-0.0000
$g_8(x)$	0.0000	-0.009	0.0000	0.0000	-0.0000
$f(x)$	2.7995	2.665	2.65856	2.65856	2.65855

Publication Ethics

This paper was presented as an oral paper at the International Congress on New Trends in Science, Engineering and Technology 2017 (ICONTRENDS), Barcelona, Spain.

References

[1] Rao, R.V., Savsani, V.J. ve Vakhaira, D.P. (2011).Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. Computer Aided Design, 43, 303-315.

[2] Arora, J.S. (2004). Introduction to Optimum Design, Waltham: Elsevier.

[3] Coello, C. A. (2002). Theoretical and numerical constraint-handling techniques used with evolutionary algorithms: a survey of the state of the art. Computer Methods in Applied Mechanics and Engineering, 191(11–12), 1245–1287.

[4] Deb, K. ve Goyal, M. (1997). Optimizing engineering designs using a combined genetic search. In: Seventh International Conference on Genetic Algorithms, Ed. I. T. Back, 512–528.

[5] Jayakumar, N. Subramanian, S. Ganesan, S. Ve Elanchezhian, E.B. (2016). Grey wolf optimization for combined heat and power dispatch with cogeneration systems. Electrical Power and Energy Systems, 74, 252-264.

[6] Lampinen, J. ve Zelinka, I. (1999) Mixed integer-discrete-continuous optimization by differential evolution. In: Proceedings of the 5th International Conference on Soft Computing, 71–76.

[7] Mirjalili, S., Mirjalili, S.M. ve Lewis, A. (2014). Grey Wolf Optimizer, Advances in Engineering Software, 69, 46-61.

[8] Mirjalili, S., Saremia, S., Mirjalili, S.M ce Coelho, L.S. (2016). Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization, Expert System with Application, 47, 106-119.

[9] S. He., E. Prempain ve Wu, Q. H. (2004). An improved particle swarm optimizer for mechanical design optimization problems. Engineering Optimization, 36 (5), 585-605, DOI: 10.1080/03052150410001704854

[10] Sandgren, E. (1990). Nonlinear integer and discrete programming in mechanical design optimization. Journal of Mechanical Design, 112, 223–229.

[11] Trabelsi, H., Yvars, P. A., Louati, J ve Haddar, M. (2015). Interval computation and constraint propagation for the optimal design of a compression spring for a linear vehicle suspension system. Mechanism and Machine Theory, 67–89.

[12] Yokota, T., Taguchi ve Gen, M. (1997). A solution method for optimal weight design problem of helical spring using genetic algorithms. Computers Ind. Engineering, 33, 71–76

İsmail ŞAHİN

Assoc. Prof. İsmail Şahin, born in 1971, received his BS and MS degrees and his PhD from Gazi University, Ankara, Turkey. He became Assistant Professor at Gazi University in 2009. He has been Associate Professor in Industrial Design Engineering, Gazi University since 2015. His research areas include mechanical design, product design, computer aided design, composites materials and artificial intelligence (artificial neural network, expert systems etc.) and metaheuristic optimisation methods.

Murat DÖRTERLER

Murat Dörterler, Received the BS degree in Department of Electronic and Computer from Gazi University in 2005. He received MSc degree and PhD degree in Electronics Computer from Gazi University 2008 and 2018 respectively. He is currently working as researcher at Faculty of Technology, Department of Computer Engineering at Gazi University. His research interests in Intelligent Transportation Systems, Artificial Intelligence, Cloud computing system, Embedded Systems, Mobile Application, Mobile Forensics, Network Forensics, Virtualization, Internet Technologies, Internet Security, Vehicular Ad Hoc Networks.

Harun GÖKÇE

Dr. Harun Gökçe, born in 1983. He graduated BS and MS degree from BEU. In 2014, he completed PhD thesis on NC Machining Simulation Systems in Gazi University. He worked in OEM automotive industries as a CAD Designer between 2005 to 2011. He has been Mechanical Designer in TÜBİTAK-SAGE since 2011. His research areas include mechanical designer, product design, CAD/CAM/CAE/PDM, optimization methods and artificial intelligence.

PRINT ISSN: 2149-4916 E-ISSN: 2149-9373 © 2017 Gazi Academic Publishing