POWER LOSS OPTIMIZATION OF A WORM GEAR MECHANISM BY USING GENETIC ALGORITHM

[1] Muhammet YAMAN (1) , Hamit SARUHAN (1) , Faruk MENDİ (2)

(1) Technical Education Faculty, Abant Izzet Baysal University, Duzce, Turkey, M.S., Dr.
(2) Technical Education Faculty, Gazi University, Ankara, Turkey, Dr.

ABSTRACT
In this study, a non-conventional algorithm namely genetic algorithm is presented for minimization of power-loss of worm gear mechanism with respect to specified set of constraints. Number of gear tooth, friction coefficient, and helix (thread) angle of worm are used as design variables and lineer pressure, bending strength of tooth,and deformation of worm are set as constraints. The results for minimization of power-loss of worm gear mechanism are presented to provide a comparison with analytical method. The results showed that the genetic algorithm is useful and applicable for optimization of mechanical components design. The genetic algorithm is an efficient search method which is inspired from natural genetics selection process to explore a given search space.

Keywords: *Genetic algorithm, power optimization, worm gear mechanism*

ÖZET
Bu çalışmada, sınır şartları verilen bir sonsuz vida karşılık dişlisi mekanizmasının güç kaybı minimizasyonunu uygun bir şekilde hesaplayan ve geleneksel olmayan bir optimizasyon metodu olan genetik algoritma kullanılmaktadır. Karşılık dişlisinin diş sayısı, sürtünme katsayısı ve vida eğim (helis) açısı tasarım değişkeni; çizgisel basınç kontrolü, eğilme gerilmesi kontrolü ve vida milinde deformasyon kontrolü sınırlama fonksiyonu olarak kullanılmıştır. Genetik algoritma ile elde edilen sonuçlar, analitik çözümle elde edilen sonuçlarla karşılaştırılmıştır. Sonuçlar, genetik algoritmanın makine parçalarının optimizasyonunda kullanışlı ve uygulanabilir olduğunu göstermiştir. Genetik algoritma, doğal genetikteki seçim işlevlerinden esinlenerek, verilen çözüm uzayında optimum çözüm arayan ve etkin çözümler sunan bir araştırma tekniğidir.

Anahtar kelimeler *: Genetik algoritma, güç optimizasyonu, sonsuz vida ve karşılık dişlisi*

1. INTRODUCTION

Up to now, many numerical optimization algorithms have been devoleped and used for design optimization of engineering problems to find optimum design. Solving engineering problems can be complex and a time consuming process when there are large numbers of design variables and constraints. Hence, there is a need for more efficient and reliable algorithms that solve such problems. The improvement of faster computer has given chance for more robust and efficient optimization methods. Genetic algorithm is one of these methods. The genetic algorithm is a search technique based on the idea of natural selection and genetics [1]. In this paper, the results found by genetic algorithm are compared with the analytical method.

2. GENETIC ALGORITHM

Genetic algorithm (GA) maintains a population of encoded solutions, and guide the population towards the optimum solutions [2]. Fitness function provides a measure of performance of an individual how fits. Rather than starting from a single point solution within the search space as in traditional optimization methods, the genetic algorithm starts running

---

[1]Adress, corresponding author, A.I.B.U. Technical Education Faculty, Machine Design and Construction Departmant, Konuralp Yerleskesi, Duzce, 81620 Turkey
E-mail: muhammet_y@ibu.edu.tr Tel:0 380 541 33 44 / 1072

with an initial population which is coding of design variables. GA selects the fittest individuals and eliminates the unfit individuals in this way. The flow chart of genetic algorithm is shown in Figure 1. An initial population is chosen rondomly at the beginning, and fitness of initial population individuals are evaluated. Then an iterative process starts until the termination criteria have been run across. After the evaluation of individual fitness in the population, the genetic operators, selction, crossover and mutation are applied to breed a new genaration. Other genetic operators are applied as needed. The newly created individuals replace the existing generation and reevalution is started for fitness of new individuals. The loop is repeated until acceptable solution is found.

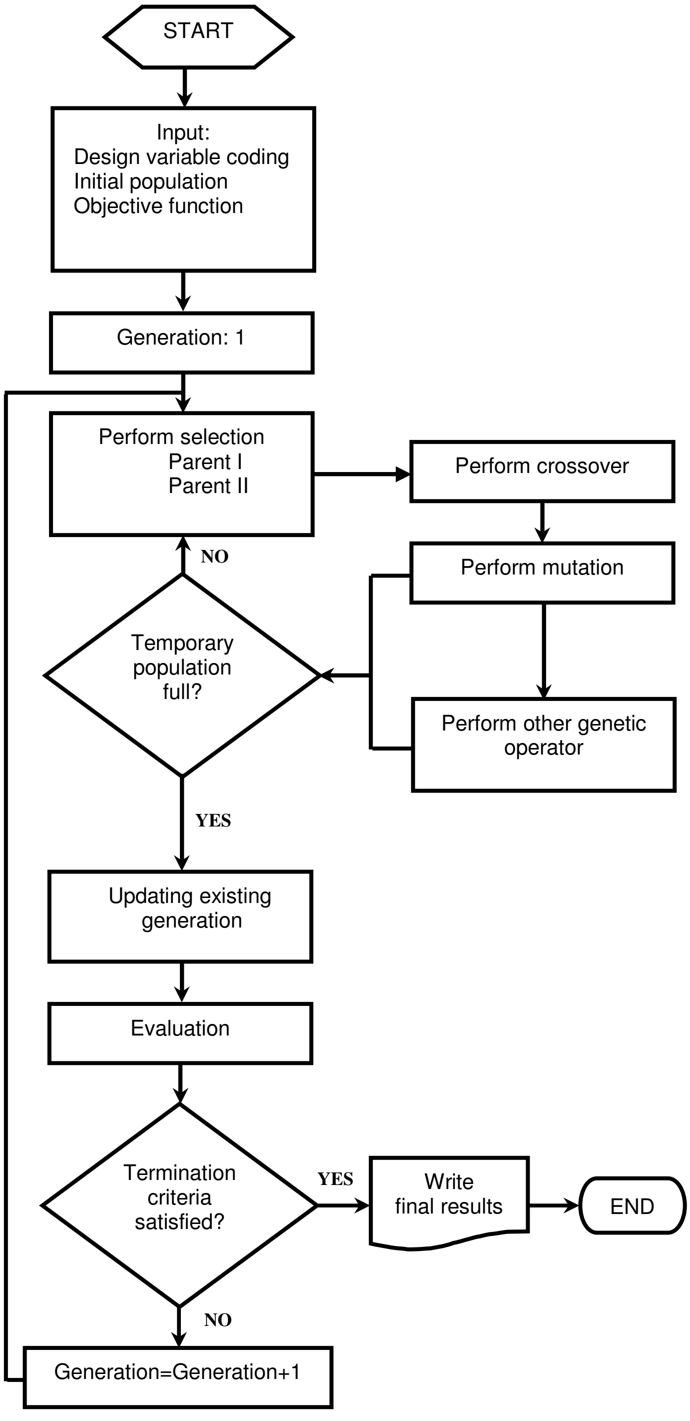

Figure 1. Flow chart for the genetic algorithm

3. PROBLEM STATEMENT

3.1. Problem Setup For Analytical Solution and Genetic Algorithm

The effort in this study is the use of the genetic algorithm method in the optimum design of worm gear for minimum power loss shown in Figure 2. The worm and worm gear are used for power and motion transmission of non-cross shaft and used for propulsion system in load transmission [4]. In the worm gearset, generally propulsion reach from worm shaft to worm gear, rarely may be reverse [5,6]. In this study, propulsion shaft is worm shaft. The materials used for the worm is hardened steel (42CrMo4) and for worm gear is bronze (GzSnBz12) [5]. In the design of gearset, the shafts angle is 90°. Transmission rate is 1/15; and three thread worm. In the worm gear system because of friction, missing power is turned to heat. The objective function is the power loss with respect to constraints such as linear pressure worm gear tooth, bending stress of gear tooth for acceptable deflection, acceptable deflection of worm shaft.

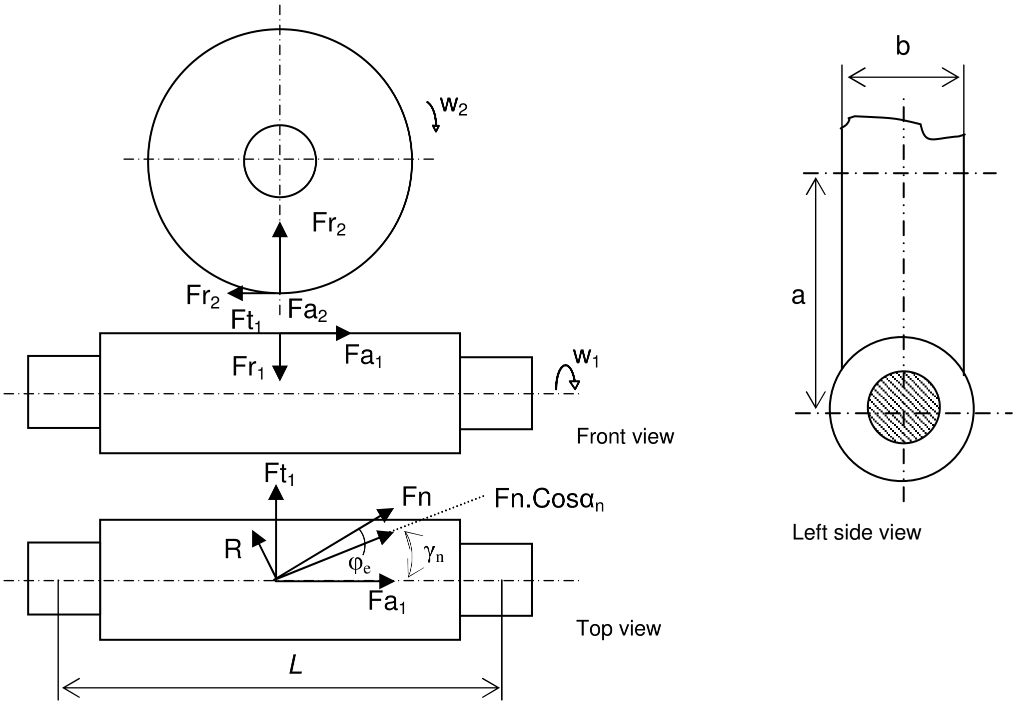

Figure 2. Shematic of a single-enveloping worm gearset

Power loss, ΔP Equation (1), formulated as:

$$\Delta P = P_i - P_o \tag{1}$$

where, P_i, input power, P_o, output power represented with Equation (2) [6].

$$P_o = F_n(\cos\alpha_n \cos\gamma_n - \mu \sin\gamma_n)\frac{m_a z_g}{2}\frac{w_w}{i} \tag{2}$$

where, F_n is normal force represented Equation (3), α_n is pressure angle, γ_n is helix angle, μ is friction coefficient, m_a is module, Z_g is number of gear tooth, w_w is angular velocity of worm, and i is transmission rate.

$$F_n = \frac{F_{t1}}{\cos\alpha_n \sin\gamma_n + \mu\cos\gamma_n} \tag{3}$$

$$F_{t1} = \frac{2P_i}{\left(\frac{2\pi}{60} n\right) do_w} \tag{4}$$

where, F_{t1} is tangential force represented in Equation (4). n is number of revolution tour of worm, do_w is worm diameter.

3.2. Objective Function

It is desired to obtain the lowest power loss of worm gear mechanism subject to linear pressure worm gear, bending stress of gear tooth and deflection of worm shaft under the load. The objective function is formulated as:

$$F_{Obj} = F(z_g, \gamma_n, \mu) = P_i - F_n(\cos\alpha_n \cos\gamma_n - \mu \sin\gamma_n)\frac{m_a z_g}{2}\frac{w_w}{i} \tag{5}$$

3.3. Design Variables and Parameters

The design variables vector consists of gear tooth number, friction coefficient, and helix angle.

gear tooth number $$21 \le z_g \le 80 \tag{6}$$

friction coefficient $$0.03 \le \mu \le 0.05 \tag{7}$$

helix angle $$15^o \le \gamma_n \le 25^o \tag{8}$$

The coefficients used in formulas and input values used for a sample design optimization are given in Table 1. The propulsion mechanism of worm gear system is electrical engine.

Table 1. Coefficients and input values for sample design practice

Definition	Symbol	Unit	Values
Input power	P_o	KW	11
Number of revolution tour of worm	n	rpm	720
Transmission rate	i	-	$15^{+0.4}_{-0.4}$
Center distance of worm gear pair	a	mm	200
Distance between of worm shaft bearings	L	mm	330
Module	m_a	mm	7
number of worm teeth	z_w	-	3
Worm diameter	do_w	mm	71
Bottom of teeth diameter of worm	df_w	mm	55
Pressure angle	α_n	derece	22.5^o
Elastisity module	E	N/mm^2	21.10^4
Inertia	I	mm^4	449000

3.4. Constraints

Constraints are conditions that must be met in the optimum design and include restrictions on the design variables. These constraints define the boundaries of the feasible and infeasible design space domain. The constraints considered for the optimum design of the power loss of worm gear are the following:

$$g_j = (z_g, \gamma_n, \mu) \le 0 \tag{9}$$

$j = 1,.....NCON$ (number of constraints) (10)

$$g_1(x) = \frac{Ft_2}{bo_g m_a z_g} 2.5 - 3.6 \leq 0 \quad (11)$$

$$g_2(x) = \frac{Ft_2}{\pi m_a bo_g} - 30 \leq 0 \quad (12)$$

$$g_3(x) = \frac{df_w}{1000} - \frac{F_{tR1} L^3}{48EI} \leq 0 \quad (13)$$

where, $g_1(x)$ is represents linear pressure worm gear tooth, $g_2(x)$ is bending stress of gear tooth, $g_3(x)$ is acceptable deflection of worm shaft [6]. F_{t2} is axial force of worm gear represented with Equation (14), bo_g is width of worm gear represented with Equation (15) and F_{tR1} is total radial force with Equation (16).

$$F_{t2} = F_n(\cos\alpha_n \cos\gamma_n - \mu \sin\gamma_n) \quad (14)$$

$$bo_g = 0.45(do_1 + 6m_a) \quad (15)$$

$$F_{tR1} = \sqrt{Ft_1^{\ 2} + Fr_1^{\ 2}} \quad (16)$$

where F_{r1} is radial force represented with Equation (17).

$$F_{r1} = F_n \sin\alpha_n \quad (17)$$

The genetic algorithm is unconstrained optimization procedure. In case of any violation of constraint boundary, the fitness of corrosponding solution is penalized, and thus kept with in feasible regions of the design space by increasing the value of the objective function when constraint violations are encountered. The penalty function is represented with Equation (19). The penalty coefficients, r_j, for the j-th constraint have to be judiciously selected. The fitness function, represented with Equation (18), a measure of the performance of an individual, which is used to bias the selection process in favour of the most fit members of the current population.

$$FitnessFunction = F - \lfloor F(z_g, \gamma_n, \mu) + PF \rfloor \quad (18)$$

$$PF = \sum_{j=1}^{NCON} r_j \left[\max\left(0, g_j\right)\right]^2 \quad (19)$$

where F is an arbitrary large enough that is greater than $F(z_g, \gamma_n, \mu) + PF$ to exclude negative fitness function values and PF is the penalty function [7].

3.5. Construction of Design Variables

In this optimization problem, design variables vector, $x = (z_g, \gamma_n, \mu)^T$, represents a solution that minimizes the objective function. The first step for applying the genetic algorithm to assign design problem is encoding of the design variables.

The genetic algorithm requires the design variables of the optimization problem to be coded. Binary coding, as a finite length strings, is generally used although other coding schemes have been used. These strings are represented as chromosomes. Each design design variables has a specified range so that $x_{(i)_{lower}} \leq x_{(i)} \leq x_{(i)_{upper}}$. The continuous design variables can be represented and discretized to a precision of ε. The number of the digits in the binary strings, l, is estimated from the Equation (20) [8].

$$2^l \geq [(x_{(i)_{upper}} - x_{(i)_{lower}}) / \varepsilon] + 1 \tag{20}$$

Where x_{upper} and x_{lower} are the upper and lower bound for design variables, respectively. The design variables are coded into the binary digit $\{0,1\}$. The physical value of the design variables can be computed from the Equation (21).

$$[(x_{(i)_{upper}} - x_{(i)_{lower}}) \big/ (2^l - 1)] d_{(i)} + x_{(i)_{lower}} \tag{21}$$

where $d_{(i)}$ represents the decimal value of string for design variables which is obtained using base-2 form. Design variables are represented in different level of precision. Table-2 gives description of these mapping.

Table 2**.** Coding of binary design variable vectors into binary digits

Design variables vector	**Lower limit**	**Upper limit**	**Precision**	**String length**
z_g	21	80	1	9
μ	0.03	0.05	0.0001	9
γ_n	15	25	0.1°	9

To start the algorithm, an initial population set is randomly assigned. This set of initialized population is potential solution to the problem. The binary string representation for the design variables, (z_g, γ_n, μ), in Table 3 gives an example of a chromosome that represents design variables accordingly. This design string is composed of 27 ones and zeros.

Table 3. The binary string representation of the variables.

Design Variables		
Gear tooth number (z_g)	**Friction coefficient (μ)**	**Helix angle (γ_n)**
1 1 0 1 0 0 0 0 0	0 1 0 0 0 0 1 1 1	0 0 1 0 1 0 1 1 1
Concanated variables head-to-tail 1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 1 0 1 1 1		

In Table 3, the string of 27-bit string length represents one of 2^{27} alternative individual solutions existing in the design space. For running the genetic algorithm, an initial population is needed to be assigned randomly at the beginning. Population size influences the number of search points in each generation. A guideline for an appropriate population size is suggested by Goldberg represented in Equation (22).

$$PopulationSize = 1{,}65.2^{0{,}21.l} \tag{22}$$

For a string length of 27bits, an optimal population size of 85 maybe used.See Table 4. The genetic algorithm then proceed by generating new solutions with bit operations utilizing the genetic algorithm operators such as selection, crossover, and mutation.

Table 4. A set of starting population

Number of Individual	Randomized binary string
1	1 1 0 1 0 0 0 0 0 0 1 0 0 0 0 1 1 1 0 0 1 0 1 0 1 1 1
2	0 1 0 1 1 0 0 1 1 1 0 1 0 0 0 1 1 0 0 0 1 0 0 1 1 1 0
.	. . . .
85	0 0 0 1 1 0 0 0 0 0 1 0 1 0 0 1 0 0 0 1 0 0 1 1 1 1 0

3.6. The Genetic Algorithm Operators

In a genetic algorithm, there are four basic operators for creating the next generation shown Figure 3 [10].

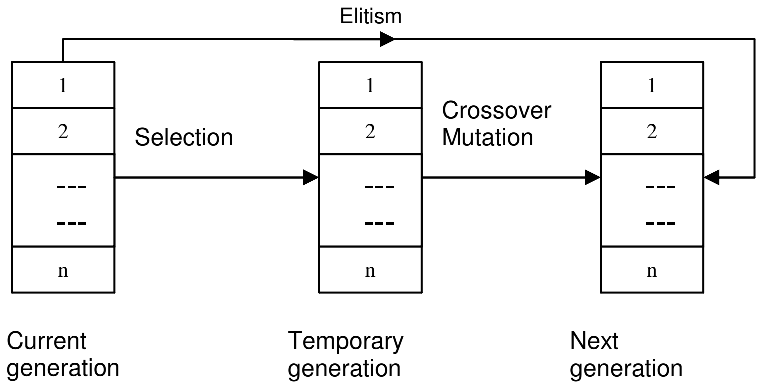

Figure 3. Genetic Algorithm Operators [10]

Each of these operators is explained and demonstrated in following: the selection operator used in this study is a tournament selection. Tournament selection approach works as follows and shown in Figure 4: a pair of individuals from mating pool is rondomly picked and the best fit two individuals from this pair will be chosen as a parent. Each pair of the parent creates two Child as described in the method of single point crossover shown in Table 5. A single point crossover operator is used in this study. Crossover is very important in the success of the genetic algorithm. This operator is primary source of new candidate solutions and provides the search mechanism that efficiently guides the evolution through the solution space towards the optimum. In single point crossover, cuts two chromosomes in one point and splices the two halves to create new ones. In Table 5, the strings, Parent I and Parent II, are selected for crossover and the genetic algorithm decides to mate them. The crossover point has been chosen at position 17. The parent exchanges the sub-strings, which occurs around crossing point that is selected randomly. The newly created strings are Child I and Child II. The Cycle starts again with selection. This iterative process continues until specified criteria are met [12].

Table 5. Single point crossover

Crossover point 17	
Parent I	1 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1 **0 1 1 1 1 0 1 1 0 1 0**
Parent II	0 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 1 0 0 1 1 1 0 0 0 1 0
The new strings	
Child I	1 0 0 0 1 0 0 0 0 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 1 0
Child II	0 1 1 1 1 1 1 1 1 1 1 0 1 0 0 0 **0 1 1 1 1 0 1 1 0 1 0**

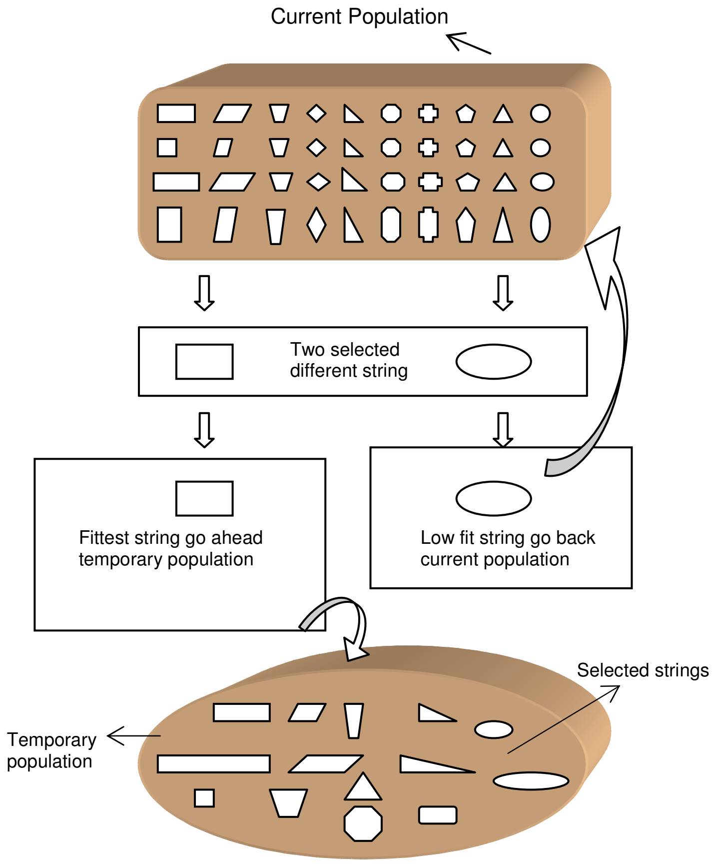

Figure 4. Tournament selection

Preventing the genetic algorithm from premature convergence to a non-optimal solution, which may diversity lost by repeated application of selection and crossover operators, mutation operator is used. Mutation is basically a process of random altering a part of individual to produce a new individual by switching the bit position from a 0 to a 1 or vice versa. Mutation rate is defined as the percentage of the total number of bits in the population. For this study, mutation probabilities of different values were tested for the genetic algorithm performance. Thus, the results show that the mutation probability of 0.005 gives preferable results compared to others. The probability of mutation is that five bit in every thousands will be mutated. A specialized mechanism, *elitism*, is added to the genetic algorithm. Elitism forces the genetic algorithm to retain the best individual in a given generation to proceed unchanged into the following generation [3]. This ensures the genetic algorithm to converge to appropriate solution. In other words, elitism is a safeguard against operation of crossover and mutation operators that may jeopardize the current best solution. The setting parameters of genetic algorithm for this study are chosen as follows:

Choromosome length = 27, population size = 85, number of generation = 100, crossover probability = 0.5, mutation probability = 0.005, and elitism = yes.

4. RESULTS
Figure 5 shows the 3-D plots of design variables values during the working of GA. Design variables have been got different values and take on minimum value of objective function at 69-th generation shown in Figure 5. It has been shown that design variables take on values as: number of gear teeth is 44, friction coefficient is 0.0305, and helix angle is 15.246°. Figure 6 shows the plots of the normalized avarege and best fitness function values in each generation as optimization proceed. The overall results show that the best design converge 69-th generation and refine the design over remaining generations, Figure 6. The results compared with results of analytical method, shown in Table 6.

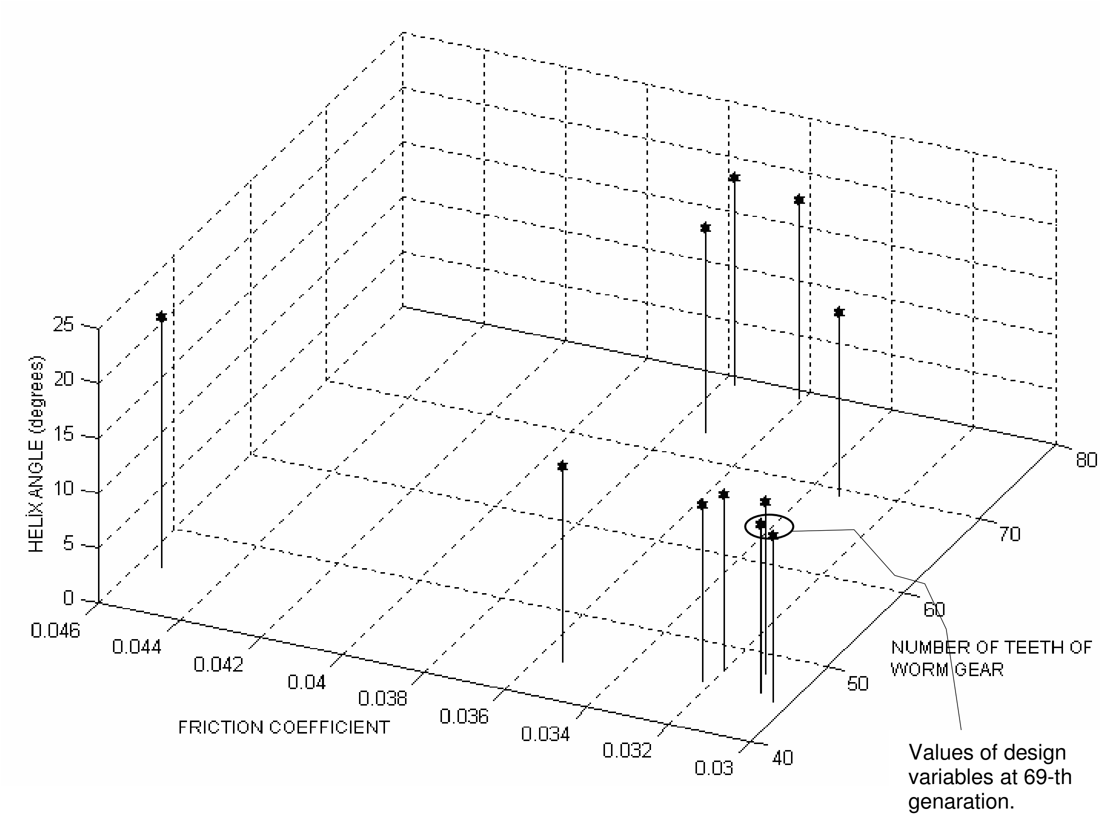

Figure 5. Variation of variables through generations

Table 6. Comparison of the results

Design Variables	Analytical method	Genetic algorithm
Number of teeth of worm gear (z_g)	46	44
Friction coefficiet (μ)	0.0390	0.0305
Helix angle (γ_n)	16.280°	15.246°
Minimum power loss	**1.362 KW**	**0.881 KW**

The starting point of analytical method, z_g =45 , μ =0.0460, γ_n =22.5°. The programme, devoleped in MATLAB 7.0 for analytical method has been run several time for different values of design variables. The results obtained are given in Table 6. As can be seen from the results, the genetic algorithm produced much beter results than analytical method.

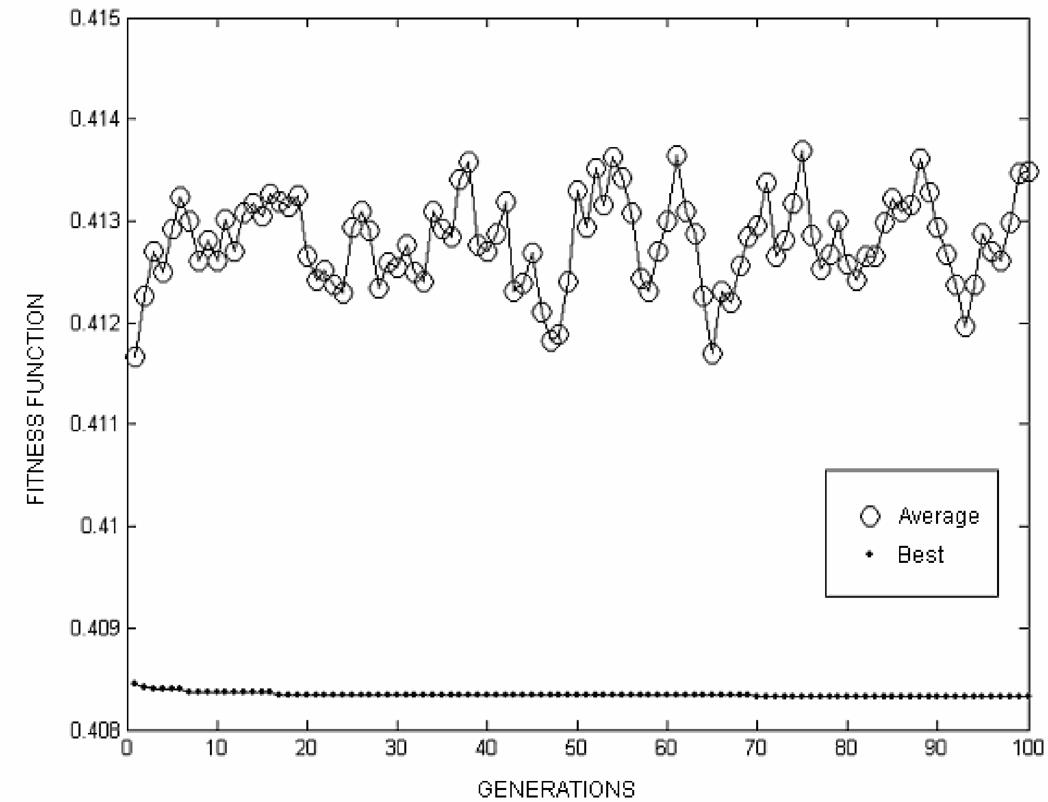

Figure 6. Convergence process of the genetic algorithm for average and best fitness funtion

5. CONCLUSIONS

The aim of this study was to optimize power loss in worm gear system using the genetic algorithm and numerical optimization method. The results showed that obtained the genetic algorithm provide better solution than those obtained from numerical optimization method. It can be concluded that the genetic algorithm that can be successfully and efficiently used for the worm gear system design.

REFERENCES

[1] GOLDBERG, D.E., "*The Design of Innovation: Lessons from Genetic Algorithms, Lessons for the real world*", University of Illinois at Urbana-Champaign, IlliGAL Report: 98004, Urbana, IL,**1998**.

[2] GOLDBERG, D. E., "*Genetic Algorithms in Search, Optimization and Machine Learning*", Addison-Wesley, Reading, **1989**.

[3] MITCHELL, M.,*"An Introduction to Genetic Algorithms"*, The MIT Press, Massachusetts, **1997**

[4] AKKURT, M., "*Makine Elemanları Cilt I-II*", Birsen Yayınevi, İstanbul, **2000**

[5] SHIGLEY,J.E., MISCHKE, C.R., BUDYNAS, R.G., "*Mechanical Engineering Design*", McGraw-Hill, **2004**

[6] NICOLET,G.,TROTTET,E., "*Eléments de Machines*", Spes S.A. Lausanne, p.316-330, **1991**

[7] SARUHAN H., UYGUR I., "*Design Optimization of Mechanical System Using Genetic Algorithms*",SAU Fen Bilimleri Dergisi,*:* 7.Cilt, 2.Sayı ,**2002**

[8] LIN, C.Y. and HAJELA, P., "*Genetic Algorithms in Optimization Problems with Discrete and Integer Design Variables*" Engineering Optimization, 19, 309-327, **1992**

[9] WU, S.J., CHOW, P.T., "*Genetic Algorithms for Nonlinear Mixed Discrete-Integer Optimization Problems via Meta-Genetic Parameter Optimization"*, Engineering Optimization,24,137-159, **1995**

[10] HE,Y, GUO, D., CHU, F. "*Using Genetic Algorithms and Finite Element Methods to Detect Shaft Crack for Rotor-Bearing System*", Deparment of Precision Instruments, Tsinghua University,Beijing 100084

[11] SARUHAN H., " *Genetic Algorithms: An Optimization Technique*",Zonguldak Karaelmas University, Technology*,:* Volume 7, Issue 1, 105-114, **2004**