DIFFERENTIAL EVOLUTION OPTIMIZATION TECHNIQUE TO DESIGN GEAR TRAIN SYSTEM

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF TECHNOLOGY
IN
MECHANICAL ENGINEERING

DEPARTMENT OF MECHANICAL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY
ROURKELA 769008

DIFFERENTIAL EVOLUTION OPTIMIZATION TECHNIQUE TO DESIGN GEAR TRAIN SYSYTEM

A thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor Of Technology

In

Mechanical Engineering

By

Rishabh Malhotra

(110ME0304)

Under the guidance of

Prof. S.S. Mahapatra

Department of Mechanical Engineering National Institute of Technology Rourkela 769008

CERTIFICATE

This is to certify that the work in this thesis entitled “Differential Evolution Optimization Technique To Design Gear Train System” by ***Rishabh Malhotra***, has been carried out under my supervision in partial fulfillment of the requirements for the degree of *Bachelor of Technology in Mechanical Engineering* during session 2013-2014 in the *Department of Mechanical Engineering, National Institute of Technology, Rourkela*.

To the best of my knowledge, this work has not been submitted to any other University/Institute for the award of any degree or diploma.

DATE- 7/5/2014

PLACE- Rourkela

Prof. Siba Sankar Mahapatra

(Supervisor) Professor

Department of Mechanical Engineering

National Institute of Technology Rourkela

ACKNOWLEDGEMENT

I am extremely fortunate to be involved in an exciting and challenging research project like "Differential Evolution Optimization Technique To Design Gear Train System". It has enriched my life, giving me an opportunity to work in a new environment of MATLAB. This project increased my thinking and understanding capability as I started the project from scratch. I would like to express my greatest gratitude and respect to my Supervisor P***rof. Siba Sankar Mahapatra***, for his excellent guidance, valuable suggestions and endless support. He has not only been a wonderful supervisor but also a genuine person. I consider myself extremely lucky to be able to work under the guidance of such a dynamic personality. Actually he is one of such genuine person for whom my words will not be enough to express.

I am also thankful to ***Dr. K. P. Maity*****, H.O.D** of Department of Mechanical Engineering, National Institute of Technology, Rourkela for his constant support and encouragement.

Last, but not the least I extend my sincere thanks to all faculty members of Mechanical Engineering Department for making my project a successful one, for their valuable advice in every stage and also giving me absolute working environment where I unlashed my potential. I would like to thank all whose direct and indirect support helped me completing my thesis in time. I want to convey my heartiest gratitude to my parents for their unfathomable encouragement.

Rishabh Malhotra

110ME0304

Bachelor Of Technology, Mechanical Engineering

Contents

ABSTRACT

Determination of volume or center-to-center distance of a gear is an important issue in design of power transmission systems. The aim of this research work was to automate the design of gear drives by minimizing volume and center-to-center distance of gear trains. Differential Evolution Optimization technique was applied to a parallel axis gear train problems. Dynamic penalty to the objective function was also introduced for handling the constraints.

Differential Evolution Optimization is metaheuristics search algorithm. It optimizes a problem by trying to improve a candidate solution iteratively. For this it takes a given measure of quality i.e. fitness fuction. DE does not require the problem to be differential as is usually required by classical optimization methods. It can be used on problems that are not even continuous.

DE assumes a population of candidate solution and creates a new candidate solution by combining existing ones in accordance to its formula. It then compares the existing solution with the new candidate solution and keeps whichever has the best score. This process is repeated several times until the stopping criterion is met.

Keywords: Constrained design problems, Differential Evolution Optimization, fitness function.

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

- k_d → Dynamic Velocity Factor
- K_{fe} → Form Factor
- K_c → Stress Concentration Factor
- F_t → Tangential Force
- ε → Overlap Ratio
- b → Face Width
- m → Module
- σ_k → Ultimate Tensile Strength
- K_m → Material Factor
- $K\alpha$ → Flange Transverse Coefficient
- K_ε → Tooth Overlap Factor
- $K\beta$ → Tooth Slope Factor
- Z_1 → No. of Teeth on pinion
- Z_2 → No. of Teeth on Gear
- P_{all} → Allowable Surface Pressure= 0.25 H_B
- H_B → Brinell Hardness
- K_0 → Working Factor

INTRODUCTION

Problems Associated with Conventional Design

Many problems in today's world rely on the trial-and-cut method which in return takes a considerable time to obtain the optimal solution. Gear is a machine element which has widespread application in industries. It transmits power with great accuracy. While designing a gear we usually use the trial –and-cut method to determine various factors such as rotation frequency, bending strength, input power, and torsional strength. However, these methods do not include the method of optimizing gear weight and center-to-center distance. Nevertheless, solving engineering problems involve a large number of conflicting objectives. In conventional methods, gear drive design requires a large number of calculations based on recommendations of gear standards, trial and error methods, etc. This is a very time consuming process and may often end with inadequate design outcomes.

Some Features of Differential Evolution Optimization Algorithm

- For minimizing non-linear and non-differentiable continuous space function.
- Requires few control variables, is robust, easy to use, and adapts itself well to parallel computation.
- System's parameters are usually represented as vectors.
- Approach to an optimization problem begins by developing an objective function that can describe the problem's objectives while incorporating any constraints.
- Once a variation is made, a decision has to be made regarding whether or not to accept the newly derived parameters.
- The new parameter vector is considered to replace the previous one if and only if it reduces the value of the objective function.
- The convergence of the greedy function is very fast, so it runs the risk of becoming trapped in a local minimum. The parallel search techniques like GA and Evolutionary techniques have some built-in safeguards to forestall misconvergence.
- Several vectors are run simultaneously. Parameter configurations that are superior help other vectors escape the local minimum.

Expectations of User from a Practical Optimization Technique

- The true global minimum should be found out, regardless of initial system parameter values.
- Convergence rate should be fast.
- Control variables should be minimum so that it is easy to use.

Problem Formulation in Differential Evolution

A problem with the following real-valued properties is considered

g_m ; m = 0,1,2,…,P-1 -----------(1)

which constitutes the objectives of the system to be optimized.

Additionally, the real valued constraints

g_m ; m = P, P+1, P+2,…, P+C-1 ------(2)

which describe the properties of the system that should not be optimized but neither should be violated.

g_m (m = 0,1,2,…,P-1) → represents properties to be optimized.

g_m (m = P, P+1, P+2,…, P+C-1) → constraint properties.

The properties of the system are dependent on the real-valued parameters:

X_j ; j=0,1,2,…,D-1 --------------------(3)

For most technical systems

$X_j \in [X_{jl}, X_{ih}]$ ---------------------(4)

Where X_{jl} lower bound on X_j & X_{jh} is the upper bound. Usually bounds on the X_j will be incorporated into the collection g_m , m>=P, of constraints.

In optimization the D-dimensional parameter vector is varied.

$X = (X_0, X_1, \ldots, X_{D-1})^T$ --------------------------(5)

Until the properties g_m are optimized and constraints g_m , m>=P are met. An optimization task can always be formulated as the minimization problem.

$\text{Min } f_m(X)$ ------------------------------(6)

Where $f_m(X)$ represents the function used to calculate the property g_m and its optimization or constraint preservation is represented by the minimization of $f_m(X)$.

All the functions $f_m(X)$ can be combined together to form a single objective function H(X), which is computed via weighted sum,

$$H(X) = \sum_{m=0}^{P+C-1} w_m * f_m(X) \quad \text{-------------------(7)}$$

or via $H(X) = \max(w_m * f_m(X))$ -------------(8)

with $w_m>0$ ----------------------------(9)

The weights are used to give the importance associated with different objectives & constraints as well as to normalize different physical units. So the optimization task can be restated as

$\min H(X)$ -----------------(10)

The min-max relation (8) & (10) guarantees that all local minima, also including the possibility of multiple global minima, can at least be found theoretically. This is true for objective function (7) only if the region of realizability of X is convex which may not be true for most technical problems.

The Method of Differential Evolution

Differential Evolution is a parallel direct search method which makes use of NP parameter vectors

$X_{i,G}$, i=0,1,2,…,NP-1 -------------(11)

as a population of Generation G. NP is made not to change during the minimization process.

The initial population is selected randomly if nothing is known about the system. We will assume probability distribution to be uniform for all random decision unless otherwise stated. If a preliminary solution is available, the normally distributed random deviations is added to the nominal solution, $X_{nom,0}$ to generate the initial solution. The crucial idea behind DE is a method to generate trial parameter vectors. A weighted difference vector between two population members is added to a third member to generate a new parameter vector. If the resulting test vector yields a lower objective function than an existing population member, the test vector will replace the existing vector with which it was compared in the following generation. The test vector may or may not be part of the generation process. In addition the best parameter vector $X_{best,G}$ is found out for every generation in order to keep track of the progress taking place during the minimization process.

SCHEME DE

For each vector $X_{i,G}$, i=0,1,2,…,NP-1, the trial vector generated is

$V= X_{1,G} + F.(X_{r2,G} - X_{r3,G})$ ----------------(12)

With r1, r2, r3 Є [0,NP-1], integer and are mutually independent, and F>0 ---------(13)

The integers r1, r2 and r3 are randomly chosen from the interval [0,NP-1] and are not same as the running index i. F is a real factor and is constant. It controls the amplification of differential vector.

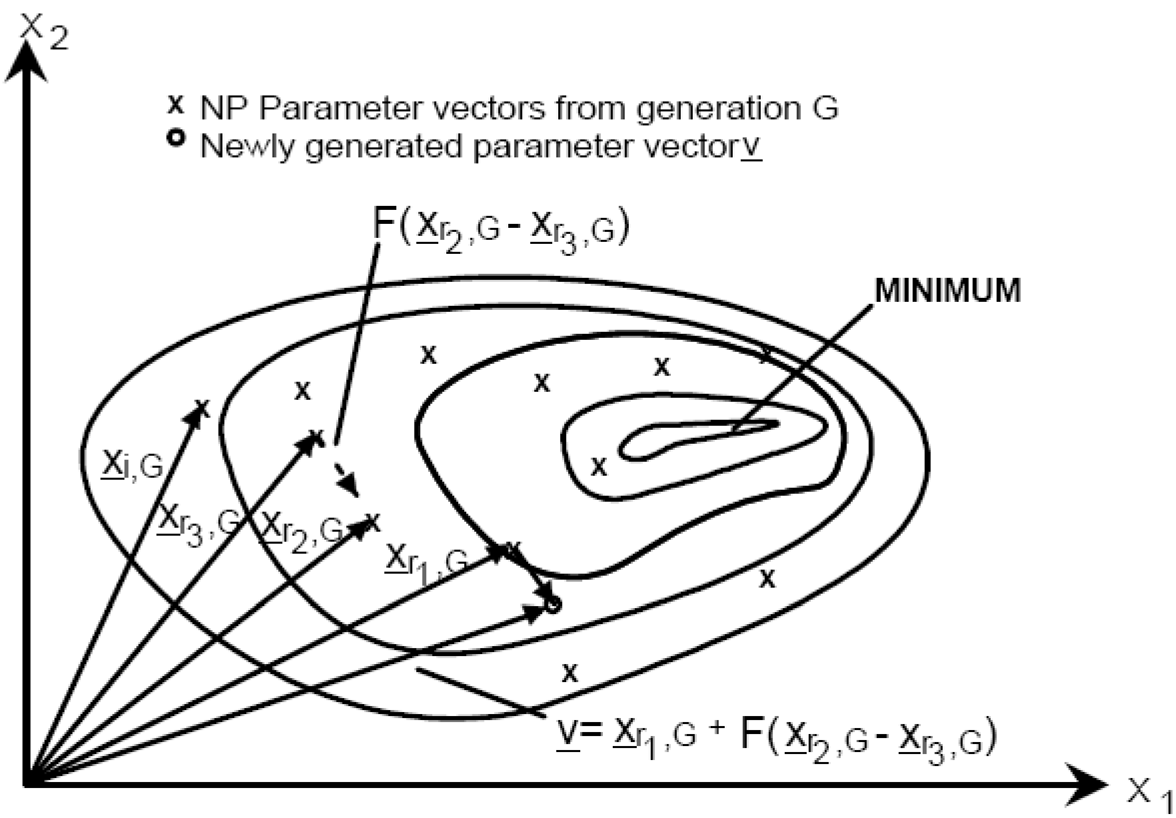


Fig. 1 Two-dimensional example of an objective function showing contour lines and the process for generating V in Scheme DE.

To increase the diversity of the parameter vector, the vector

$U = (U_0 , U_1 , \ldots, U_{D-1})^T$ -------------------(14)

With $U_j =$ V_j for $j=\langle n\rangle_D, \langle n+1\rangle_D,\ldots,\langle n+L-1\rangle$

-----------------(15)

$(X_{i,G})_j$ for all other $j \in [0,D-1]$

Where $\langle \ \rangle_D$ denotes the modulo function with modulus D.

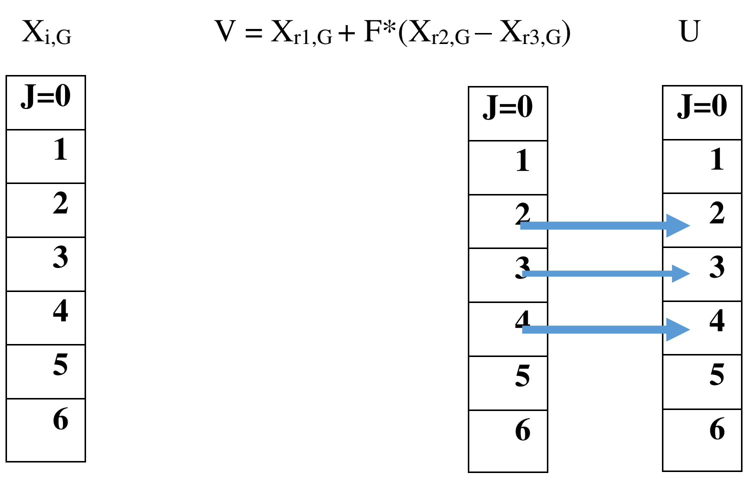
The above equations result in certain sequence of the vector elements of U to be identical to that of V, the other elements of U take up the original values of $X_{i,G}$. The idea is illustrated in the figure given in the next page for D=7, n=2 and L=3. The starting index n is chosen randomly from the interval [0, D-1]. The integer L denotes the number of parameters to be exchanged. It is selected from the interval [1, D]. The algorithm that determines the working of L according to the following lines of code rand(), which generates a random number from [0,1].

```
L=0;

do{

        L=L+1;

   }
While {rand () <CR AND (L<D)}
```



Parameter vector joining
The parameters $\mathbf{x_{j,\ j=0,1,2,\ldots,D-1}}$

Fig. 2 Mutation in DE

In order to check if the new vector U becomes the member of generation G+1, it is compared to $X_{i,G}$. If the vector has a smaller function value than $X_{i,G}$, $X_{i,G+1}$ is set to U, otherwise the earlier value is retained.

The Algorithm

- The required DE parameters are input. All the vector populations are initialized randomly in the limits specified for the decision variables.
- Each member of the population is evaluated. The population that give nondominated solutions in the current population are identified and stored in nondominated elitist archive (NEA).
- Mutation and crossover operations are performed on all the members of the population, i.e. for each parent P_i
    a) The distinct vectors are selected randomly from the current population other than the parent vector.
    b) The new mutation vector is calculated.
    c) The mutated vector is modified by binary crossover with the parent using crossover probability CR.
    d) The variables are restricted to their boundaries, if any variable is outside the lower or upper bound.
- Each member of the population is evaluated. Dominance with the parents is checked for. If the candidate dominates the parent, it replaces the parent. If the parent dominates the candidate, it replaces the candidate. Otherwise the candidate is added to a temporary population (tempPop).
- The latest solution vectors are added to the tempPop. The nondominated crowding and sorting assignment operators are used to select the individuals to the next generation. The nondominated solutions are stored in the nondominated elite archive (NEA). If the size of NEA exceeds the desired number of pareto optimal set, then desired number of least crowded members is selected with the help of crowding assignment operator. The tempPop is emptied.
- The generation counter is incremented, G to G+1 and termination criterion is checked for. If the termination criterion is not met, then return to set 3. Otherwise the nondominated solution set from NEA is output.

DE and GA: A Comparison

- Both DE and GA make use of same evolutionary operations like mutation and crossover for moving towards the optimal solution.

- In GA's, mutation occurs due to small perturbations to the genes of an individual while in DE mutation occurs due to arithmetic combination of individuals. In DE, mutation is the major role playing operation while in GA crossover plays the major role.

Literature survey

The main focus of design of gear drives has been on single stage gear drives. Several applications have been developed by the researchers using different design and calculation methods. A gearbox was designed to produce the desired output speed by using GA [1] The objective function stated the number of teeth and number of shafts. The constraints used were maximum transmission ratio, number of teeth of gear and maximum number of shafts. For automating preliminary design of multi stage gear an algorithm was proposed [2]. The algorithm which consisted of four steps was run iteratively so as to obtain a desirable solution. The steps in the algorithm were conducted manually, by random search and generate and test methods. A simulated annealing algorithm for minimizing geometrical volume of a gearbox by means of integrating configurationally and dimensionally design process was used. An optimal weight design problem using GA was studied for a gear pair system [3]. A generalized optimal design formulation to gear trains was presented [4]. The tradeoff between minimum volume and surface fatigue life using multiobjective optimization was studied. A CAD approach to gears was proposed [5] to optimize single stage gear pair. GA was used for minimizing volume of gear by reducing center distance of gear pairs and other parameters such as transmitting power, reduction ratio. An epert system involving a GA module was developed in a study [6].

In recent times, many algorithms have been introduced for multiobjective optimization. Most of these exist in the field of Evolutionary Algorithms (EAs) – also known as Multiobjective Optimization EAs (MOEAs). Among these are NSGA (Non-dominated Sorting Genetic Algorithm-II) by Deb et al. [7] and SPEA2 (Strength Pareto Evolutionary Algorithm 2) by Zitzler et al. [8] which are the most popular. MOEA work by taking strong points of EAs and apply them to Multiobjective Optimization Problem (MOPs). An important EA used for multiobjective optimization is Differential Evolution (DE). DE is found to be very simple but a very powerful evolutionary algorithm by Price & Storn [9], it has been successful in solving single-objective optimization problems [10]. Hence, it has been tried by several researchers to extend this to handle MOPs.

Madavan [11] achieved very good results by using Pareto Differential Evolution Approach (PDEA1). PDEA is applied to DE to create new individuals. It combines both populations and carries out the calculation for nondominated rank (with Pareto-based ranking assignment) and diversity rank (with the crowding distance metric) for all the individuals. Two variants of PDEA were found to be investigated. The first used a method to compare each child with its parent. The child was found to replace the parent if had higher or same same nondominated rank and a higher Diversity rank. Otherwise the algorithm discarded the child. The variant didn't produce very likely results. Although the diversity was found to be good, but the convergence was slow. The other variant simply used nondominated rank and diversity rank to find the best individual. It was found to produce highly efficient and produced favourable results for several MOPs to which it was applied.

Xue [12] introduced Multiobjective Differential Evolution (MODE). The algorithm uses crowding distance metric and Pareto-based ranking assignment, but in an approach that is different from PDEA (Pareto Differential Evolution Approach). Fitness is calculated using Pareto-based ranking and it is then reduced according to the individuals crowding distance value. This fitness value is used to select best individuals for the upcoming population. It produced better results than SPEA (Strength Pareto Evolutionary Algorithm) in five benchmark problems.

METHODOLOGY ADOPTED

1. Problem Definition

In optimizing the weight and center-to-center distance the design variables used were number of teeth, module and face width. In this study, larger values of modules were not taken as they do not contribute to the objective function. In design of gear pairs, the material of gear and pinion was taken to be the same. Process of design was conducted based on the pinion. Thus tooth of gear was defined depending on tooth of pinion and gear ratio. The interval for number of tooth of pinion was chosen to be $17 \le z \le 24$. Face width was determined at the beginning with due consideration to design width factor which lies between 20 and 40 based on recommendation (Bozaci, llknur, Colak, 2001).

Width= design width factor *module

2. Variables

The vector of design variables

X(i) = {module(m), number of teeth(z), face width(b)}

So, X1=module
X2=number of teeth
X3=face width

3. Input Parameters

The input parameters are material properties that are entered by the user to the system. These are used for calculation of the objective function value and various constraints.

The various input parameters are:

- Transferred Power
- Input Speed
- Gear Ratio
- Brinel Hardness Number
- Ultimate Tensile Strength
- Working Factor

- Overlap Ratio
- Stress Concentration Factor
- Material Factor
- Flank Transverse Coefficient
- Tooth Overlap Factor
- Helical Angle
- Tooth Slope Factor
- Type of Gear
- Weightage
- Crossover Probability

4. Forming Objective Function

An objective function is defined as the quantity to be minimized or maximized by analyzing a search space under the imposed constraints (Saruhan, Rouch & Roso, 2004). In this research work, minimization of material weight and center-to-center distance are the objectives. So objective function is defined by

$$F_{obj} = w_1 * \text{weight} + w_2 * \text{center-to-center distance}$$

This can be represented as

F_cost_tol=(w1*7.7005*x1^2*x2*x3^2*density*(1+gear_ratio^2)/1000000000)+w2*0.5 *x1*x3*(1+gear_ratio)/1000

A number of constraints were included in the objective function to be minimized. The constraints provide suitable design choices and act as sub functions to restrict the objective function so that suitable contents are inculcated. Types of failures that are mostly seen are surface fatigue and tooth failure in a transmission gear (Akinci, Yilmaz & Canakci, 2005). Thus the main constraints in gear design were contact stress and bending strength. The others were related to standards of gear sizing. These are defined as

g(j) = Bending Strength Constraint, Contact Stress Constraint, Module Constraint, Pinion Teeth Constraint, Face Width Constraint.

5. Forming Penalty Function

A penalty function was used to check that the constraints are not violated. It was incorporated with the objective function. Every time the constraint was violated, a high value is added to the objective function, which increases its value and is against the minimization task. The value added measures how much the left-side of the constraint equation varies with respect to the right-side.

So, to incorporate the effects of penalty function the following changes to the objective function were made:

```
L=stress_concentration_factor*transferred_power*1000000*cos(20)*2/(input_speed*2*3
.14*overlap_ratio*0.55*ultimate_tensile_strength)

G=cos(helical_angle)*(material_factor*flank_transverse_coefficient*tooth_overlap_ratio
*tooth_slope_factor)^2*transferred_power*1000000*cos(20)*2*60/(2*3.14*input_speed
*gear_ratio*(allowable_surface_pressure)^2)


if(x1^2*x2*x3^2/dynamic_velocity_factor>=G)
F_cost_tol=(w1*7.7005*x1^2*x2*x3^2*density*(1+gear_ratio^2)/1000000000)+w2*0.5
*x1*x3*(1+gear_ratio)/1000;
else
F_cost_tol=(w1*7.7005*x1^2*x2*x3^2*density*(1+gear_ratio^2)/1000000000)+w2*0.5
*x1*x3*(1+gear_ratio)/1000+(G-x1^2*x2*x3^2/dynamic_velocity_factor);
end

if(x1^2*x2*x3^2/(dynamic_velocity_factor*(0.48356*x3-2.86368))<L)
F_cost_tol=F_cost_tol+(L-x1^2*x2*x3^2/(dynamic_velocity_factor*(0.48356*x3-
2.86368)));
end


if(x2<20*x1)
F_cost_tol=F_cost_tol+(20*x1-x2);
elseif(x2>40*x1)
F_cost_tol=F_cost_tol+(x2-40*x1);
else
F_cost_tol=F_cost_tol;
end
```

6. Forming Constraints

The formulas of contact stress, bending strength and face width were used for forming constraints for design of gears.

$k_d * K_{fe} * K_c * F_t - \mathcal{E} * b * m * (0.55 * \sigma k) <= 0$
$k_d * (K_m * K\alpha * K_\varepsilon * K\beta)^2 * F_t * (Z_1 + Z_2)/Z_1 - b * m * Z_2 * (P_{alw})^2 <= 0$
$20 * m - b <= 0$
$b - 40 * m <= 0$
$17 - Z <= 0$
$Z - 24 <= 0$

7. Steps

a. Selection of material
b. Inputting the following material properties: Transfer power, Input speed, Gear ratio, Brinell hardness number, Ultimate tensile strength, Working factor, Overlap ratio, Stress concentration factor, Material factor, flank transverse coefficient, Tooth overlap factor, Helical angle, Tooth slope factor, Type of gear
c. Allot the weightage to the objectives
d. Run the algorithm
e. Results are displayed
f. Carry out the same steps for different types of gears & different weightages.
g. Find out the weightage which gives the best result for a particular type of gear.

8. Possibilities

- **Three types of weightages considered for examples:** (0.25,0.75), (0.5, 0.5), (0.75, 0.25). However there exist infinite possibilities.
- **Types of Gears considered for examples:** 6. However the program provides 8 types.
- **Type of DE:** 5 variants are available.

So total number of different cases = 3 * 6 * 5

EXPERIMENTAL DATA

For the examples considered the following are the assumptions:

Assumptions

- A helical gear pair
- Pressure Angle = 20º
- Full Depth System
- Material = Any material
- Available Gear types
    a. Ordinary Cut Gears
    b. Carefully Cut Gears
    c. Carefully Cut & Ground Metallic Gears
    d. Hardened Steel, ground and lapped in precision
    e. Non-metallic Gears
    f. Gears whose tooth are finished by hobbing or shapping
    g. Gears with high precision shaped or ground teeth & if dynamic load is developed
    h. Gears with high precision shaped or ground & there is applicable dynamic load
- Range of module = 1-5 mm
- Range of face width = 20-30 mm
- Range of number of teeth = 17-24
- Crossover probability = 0.5

Input Parameters

For the examples (Cementite Steel) considered the following are the inputs:

1. Transferred Power (KW)= 7.5
2. Material: Cementite Steel
3. Input Speed (rpm) =1800
4. Gear Ratio, i = 6
5. Brinell Hardness Number (N/mm^2) = 1460
6. Ultimate Tensile Strength, σ_k (N/mm^2) =1100
7. Working Factor, K_o = 1.25

8. Overlap Ratio, ε = 1.6
9. Stress Concentration Factor, K_c = 1.5
10. Material Factor, K_m (N/mm^2) = 271.11
11. Flank Transverse Coefficient, $K\alpha$ = 1.76
12. Tooth Overlap Factor, K_ε =0.79
13. Helical Angle, ß = 18^o

RESULTS

We have 8 selection criterion for gears, namely

- **Ordinary Cut Gears**
  Dynamic velocity factor= 3/(3+V)

- **Carefully Cut Gears**
  Dynamic velocity factor=4.5/(4.5+V)

- **Carefully Cut & Ground Metallic Gears**
  Dynamic velocity factor=6/(6+V)

- **Hardened Steel, ground and lapped in precision**
  Dynamic velocity factor=5.6/(5.6+sqrt(V))

- **Non-metallic Gears**
  Dynamic velocity factor=0.75/(1+V) +0.25

- **Gears whose tooth are finished by hobbing or shapping**
  Dynamic velocity factor=50/(50+sqrt(200*V))

- **Gears with high precision shaped or ground teeth & if dynamic load is developed**
  Dynamic velocity factor=(78/(78+sqrt(200*V)))^1/2

- **Gears with high precision shaped or ground & there is applicable dynamic load**
  Dynamic velocity factor=1

For each selection criterion we have a different value for dynamic velocity factor. This in turn will affect the constraint conditions. With the variation of constraint conditions, we obtain different penalty for different cases, which in turn affects the function values and the optimal value of variables. Here in this research work 6 cases (except 4^{th} and 5^{th}) are considered. The results for the cases is tabulated below:

Ordinary Cut Gears

Type of Algorithm	Weightage	Iterations	Function Value	Module (mm)	Face Width (mm)	No. of teeth	Weight	Center to Center distance
Standard DE	(0.25,0.75)	50	4.316890	1.09369	22.4233	17	17.068	65.073
	(0.50,0.50)	40	8.635823	1.08321	23.0399	17	17.206	64.45
	(0.75.0.25)	50	12.776666	1.09651	22.2318	17	17.025	65.24
Parent Centric	(0.25,0.75)	50	4.306695	1.0913	22.4685	17	17.028	64.932
	(0.50,0.50)	40	8.5327675	1.08957	22.4981	17	17	64.829
	(0.75,0.25)	60	12.790328	1.09447	22.3384	17	17.03	65.12
DE with Jitter	(0.25,0.75)	230	4.282545	1.09537	22.1743	17	16.932	65.174
	(0.50,0.50)	50	8.556773	1.09455	22.3567	17	17.046	65.125
	(0.75,0.25)	200	12.690054	1.09654	22.0793	17	16.897	65.244
DE with per vector dither	(0.25,0.75)	150	4.299084	1.10018	22.0655	17	16.996	65.46
	(0.50,0.50)	220	8.504125	1.09557	22.1773	17	16.942	65.186
	(0.75,0.25)	620	12.776798	1.09818	22.1642	17	17.013	65.341
DE with per generation dither	(0.25,0.75)	200	4.278728	1.09641	22.1119	17	17.092	65.236
	(0.50,0.50)	60	8.546620	1.09423	22.3432	17	17.028	65.106
	(0.75,0.25)	650	12.739897	1.09593	22.191	17	16.964	65.207

Table 1: Results of DE for ordinary cut gears

Carefully Cut Gears

Type of Algorithm	Weightage	Iterations	Function Value	Module (mm)	Face Width (mm)	No. of teeth	Weight	Center to Center distance
Standard DE	(0.25,0.75)	560	5.853739	1.19749	25.4191	17	23.2	71.25
	(0.50,0.50)	1730	11.681697	1.19174	25.766	17	23.3	70.908
	(0.75.0.25)	700	17.327191	1.20423	25.0032	17	23.077	71.651
Parent Centric	(0.25,0.75)	1630	5.881739	1.19253	25.7558	17	23.312	70.955
	(0.50,0.50)	1550	11.746289	1.18702	26.1156	17	23.42	70.627
	(0.75,0.25)	120	17.459389	1.20507	25.1587	17	23.253	71.701
DE with Jitter	(0.25,0.75)	40	6.132619	1.1401	29.3666	17	24.292	67.835
	(0.50,0.50)	1360	11.875636	1.17199	27.0868	17	23.68	69.733
	(0.75,0.25)	690	18.158173	1.14737	28.8661	17	24.18	68.268
DE with per vector dither	(0.25,0.75)	40	5.855116	1.19626	25.4776	17	23.204	71.177
	(0.50,0.50)	1370	11.657716	1.19544	25.5537	17	23.246	71.128
	(0.75,0.25)	1710	17.378355	1.120037	25.2387	17	20.152	66.642
DE with per generation dither	(0.25,0.75)	680	5.974844	1.17421	26.9937	17	23.968	69.865
	(0.50,0.50)	1730	11.527748	1.20997	26.6641	17	24.846	71.993
	(0.75,0.25)	1880	17.516991	1.19016	25.8785	17	23.33	70.814

Table 2: Results of DE for carefully cut gears

Carefully Cut and Ground Metallic gears

Type of Algorithm	Weightage	Iterations	Function Value	Module (mm)	Face Width (mm)	No. of teeth	Weight	Center to Center distance
Standard DE	(0.25,0.75)	1250	7.328080	1.27229	28.2291	17	29.084	75.701
	(0.50,0.50)	1730	14.468203	1.28248	27.5669	17	28.858	76.307
	(0.75.0.25)	1400	21.492458	1.29281	26.9128	17	28.629	76.922
Parent Centric	(0.25,0.75)	570	7.313105	1.2743	28.0818	17	28.676	75.82
	(0.50,0.50)	1390	14.505219	1.27997	27.7464	17	28.932	76.158
	(0.75,0.25)	230	21.655710	1.28998	27.2366	17	28.846	76.75
DE with Jitter	(0.25,0.75)	1370	7.406166	1.25151	29.6406	17	29.548	76.464
	(0.50,0.50)	110	14.871500	1.24648	30	17	29.666	74.165
	(0.75,0.25)	560	21.381890	1.29865	26.534	17	28.388	77.269
DE with per vector dither	(0.25,0.75)	1540	7.313766	1.2767	27.9784	17	29.024	75.96
	(0.50,0.50)	1420	14.357933	1.29192	26.9574	17	28.368	76.869
	(0.75,0.25)	610	21.702304	1.2808	27.6881	17	28.909	76.207
DE with per generation dither	(0.25,0.75)	1640	7.439237	1.25237	29.583	17	29.532	74.516
	(0.50,0.50)	1080	14.440172	1.28438	27.4321	17	28.802	76.42
	(0.75,0.25)	70	21.499640	1.29367	26.886	17	28.638	76.973

Table 3: Results of DE for carefully cut and ground metallic gears

Gears whose teeth are finished by hobbing or shaping

Type of Algorithm	Weightage	Iterations	Function Value	Module (mm)	Face Width (mm)	No. of teeth	Weight	Center to Center distance
Standard DE	(0.25,0.75)	330	159.568144	4.54267	30	17	394.036	270.288
	(0.50,0.50)	350	258.012615	4.54267	30	17	362.118	362.118
	(0.75.0.25)	420	356.457087	4.54267	30	17	394.026	270.288
Parent Centric	(0.25,0.75)	320	159.568144	4.54267	30	17	394.036	270.288
	(0.50,0.50)	290	258.012615	4.54267	30	17	362.118	362.118
	(0.75,0.25)	230	356.457087	4.54267	30	17	394.026	270.288
DE with Jitter	(0.25,0.75)	210	159.568144	4.54267	30	17	394.036	270.288
	(0.50,0.50)	270	258.012615	4.54267	30	17	362.118	362.118
	(0.75,0.25)	340	356.457087	4.54267	30	17	394.026	270.288
DE with per vector dither	(0.25,0.75)	290	159.568144	4.54267	30	17	394.036	270.288
	(0.50,0.50)	480	258.012615	4.54267	30	17	362.118	362.118
	(0.75,0.25)	260	356.457087	4.54267	30	17	394.026	270.288
DE with per generation dither	(0.25,0.75)	180	159.568144	4.54267	30	17	394.036	270.288
	(0.50,0.50)	310	258.012615	4.54267	30	17	362.118	362.118
	(0.75,0.25)	470	356.457087	4.54267	30	17	394.026	270.288

Table 4: Results of DE for gears whose tooth are finished by hobbing or shaping

Gears with high precision shaped or ground teeth & if dynamic load is developed

Type of Algorithm	Weightage	Iterations	Function Value	Module (mm)	Face Width (mm)	No. of teeth	Weight	Center to Center distance
Standard DE	(0.25,0.75)	320	130.732468	4.07124	30	17	290.856	242.238
	(0.50,0.50)	330	209.79783	4.07124	30	17	316.496	242.238
	(0.75.0.25)	360	288.863498	4.07124	30	17	316.496	242.236
Parent Centric	(0.25,0.75)	370	130.732468	4.07124	30	17	290.856	242.238
	(0.50,0.50)	240	209.79783	4.07124	30	17	316.496	242.238
	(0.75,0.25)	310	288.863498	4.07124	30	17	316.496	242.236
DE with Jitter	(0.25,0.75)	170	130.732468	4.07124	30	17	290.856	242.238
	(0.50,0.50)	460	209.79783	4.07124	30	17	316.496	242.238
	(0.75,0.25)	280	288.863498	4.07124	30	17	316.496	242.236
DE with per vector dither	(0.25,0.75)	230	130.732468	4.07124	30	17	290.856	242.238
	(0.50,0.50)	250	209.79783	4.07124	30	17	316.496	242.238
	(0.75,0.25)	340	288.863498	4.07124	30	17	316.496	242.236
DE with per generation dither	(0.25,0.75)	430	130.732468	4.07124	30	17	290.856	242.238
	(0.50,0.50)	390	209.79783	4.07124	30	17	316.496	242.238
	(0.75,0.25)	330	288.863498	4.07124	30	17	316.496	242.236

Table 5: Results of DE for gears with high precision, shaped or ground teeth & if dynamic load is developed

Gears with high precision shaped or ground teeth & there is no dynamic load

Type of Algorithm	Weightage	Iterations	Function Value	Module (mm)	Face Width (mm)	No. of teeth	Weight	Center to Center distance
Standard DE	(0.25,0.75)	130	791362.595	5	30	23	873.8	402.5
	(0.50,0.50)	70	791580.951	5	30	23	873.802	402.5
	(0.75.0.25)	90	791799.307	5	30	23	873.01	402.5
Parent Centric	(0.25,0.75)	150	791362.595	5	30	23	873.8	402.5
	(0.50,0.50)	80	791580.951	5	30	23	873.802	402.5
	(0.75,0.25)	130	791799.307	5	30	23	873.01	402.5
DE with Jitter	(0.25,0.75)	120	791362.595	5	30	23	873.8	402.5
	(0.50,0.50)	100	791580.951	5	30	23	873.802	402.5
	(0.75,0.25)	90	791799.307	5	30	23	873.01	402.5
DE with per vector dither	(0.25,0.75)	130	791362.595	5	30	23	873.8	402.5
	(0.50,0.50)	100	791580.951	5	30	23	873.802	402.5
	(0.75,0.25)	190	791799.307	5	30	23	873.01	402.5
DE with per generation dither	(0.25,0.75)	110	791362.595	5	30	23	873.8	402.5
	(0.50,0.50)	130	791580.951	5	30	23	873.802	402.5
	(0.75,0.25)	170	791799.307	5	30	23	873.01	402.5

Table 6: Results of DE for gears with high precision, shaped or ground teeth & there is no dynamic load

Below are a few out of many graphical results that were obtained during the research work:

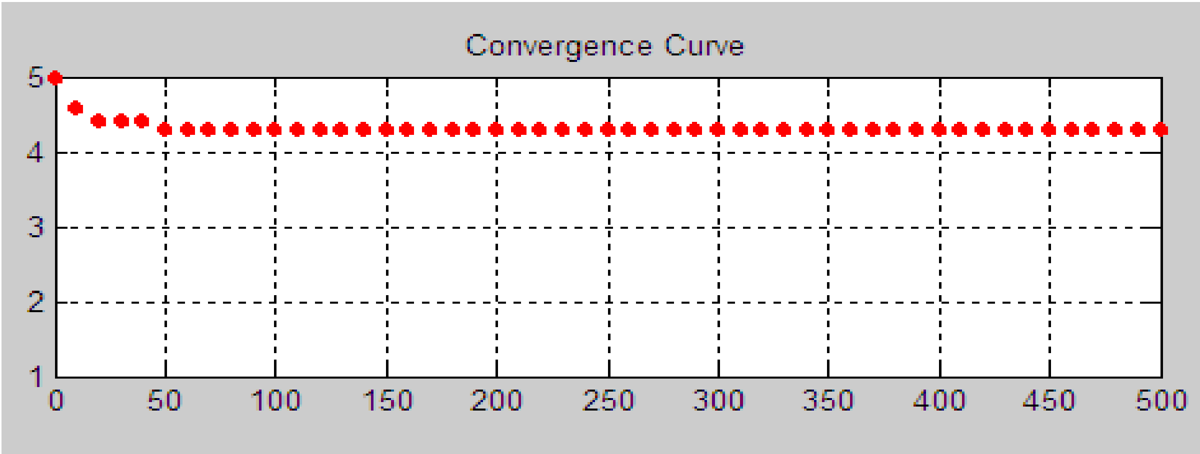


Fig. 3 Plot of function value vs iterations for ordinary cut gears, DE with jitter for weightage (0.25,0.75)

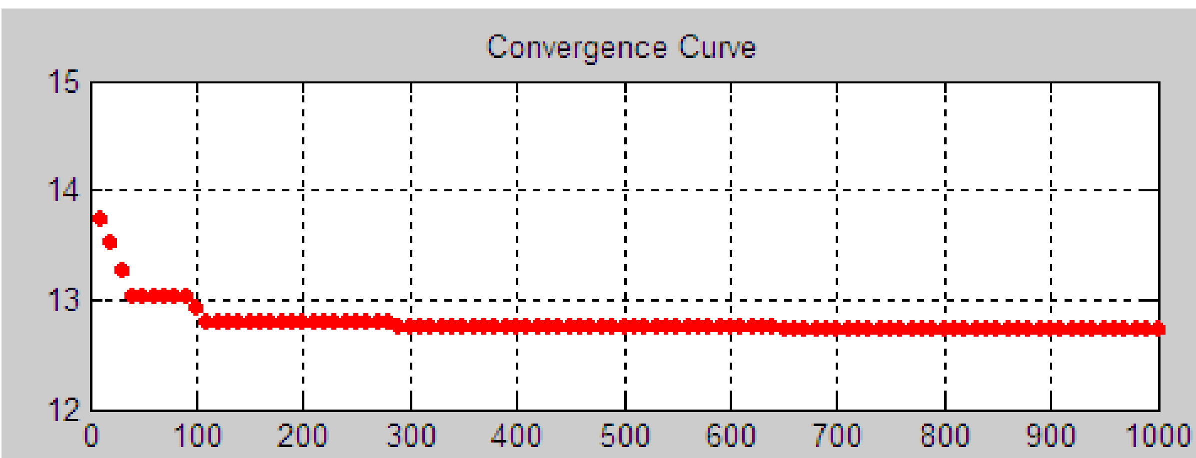


Fig. 4 Plot of function value vs iterations for ordinary cut gears, DE with per vector dither for weightage (0.75,0.25)

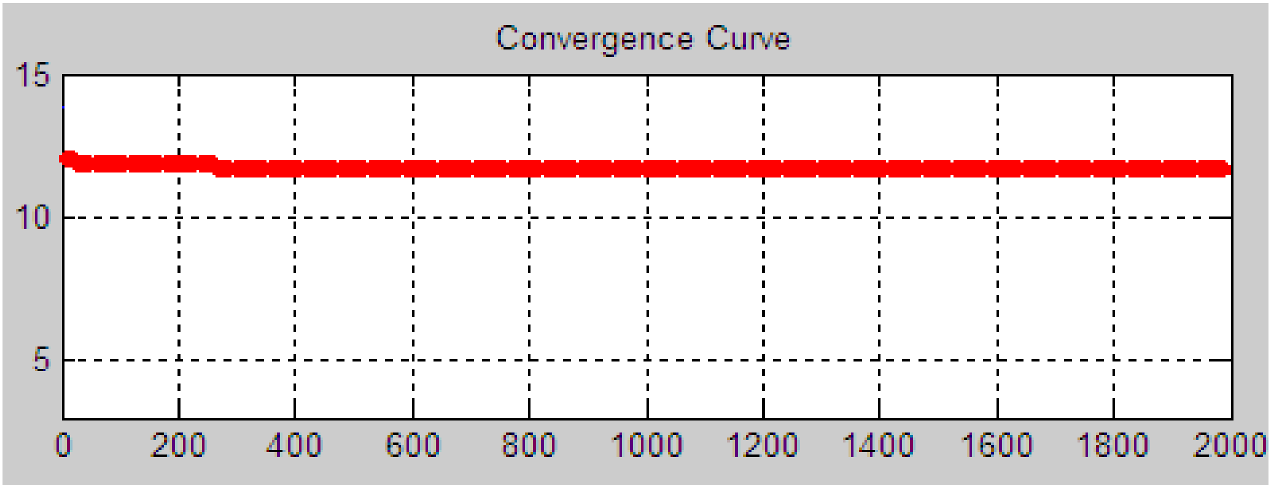


Fig. 5 Plot of function value vs iterations for carefully cut gears, Standard DE for weightage (0.5,0.5)

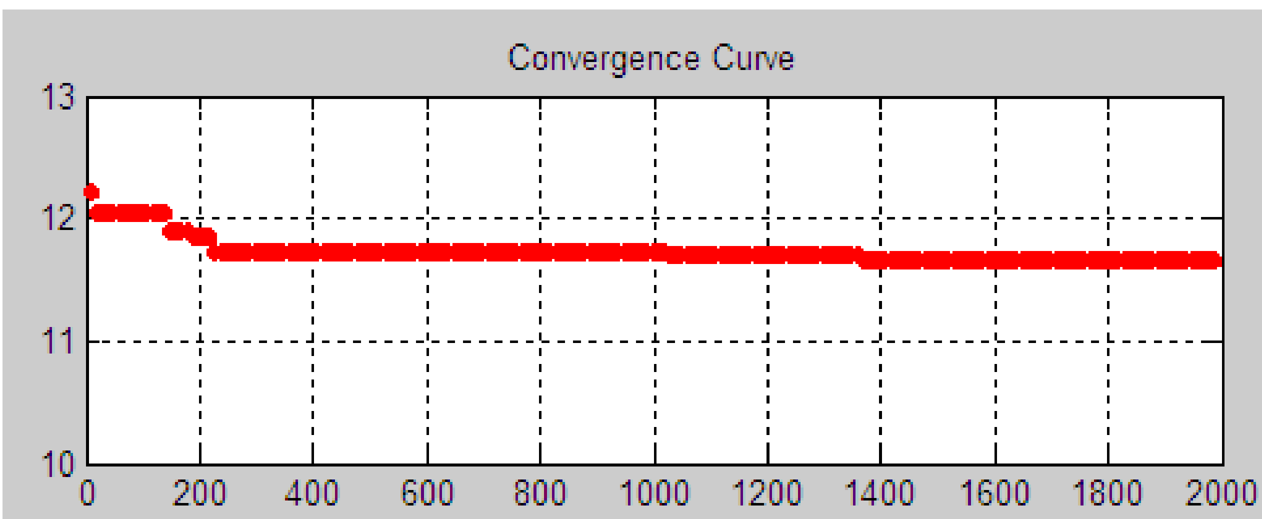


Fig. 6 Plot of function value vs iterations for carefully cut gears, DE with per vector dither for weightage (0.5,0.5)

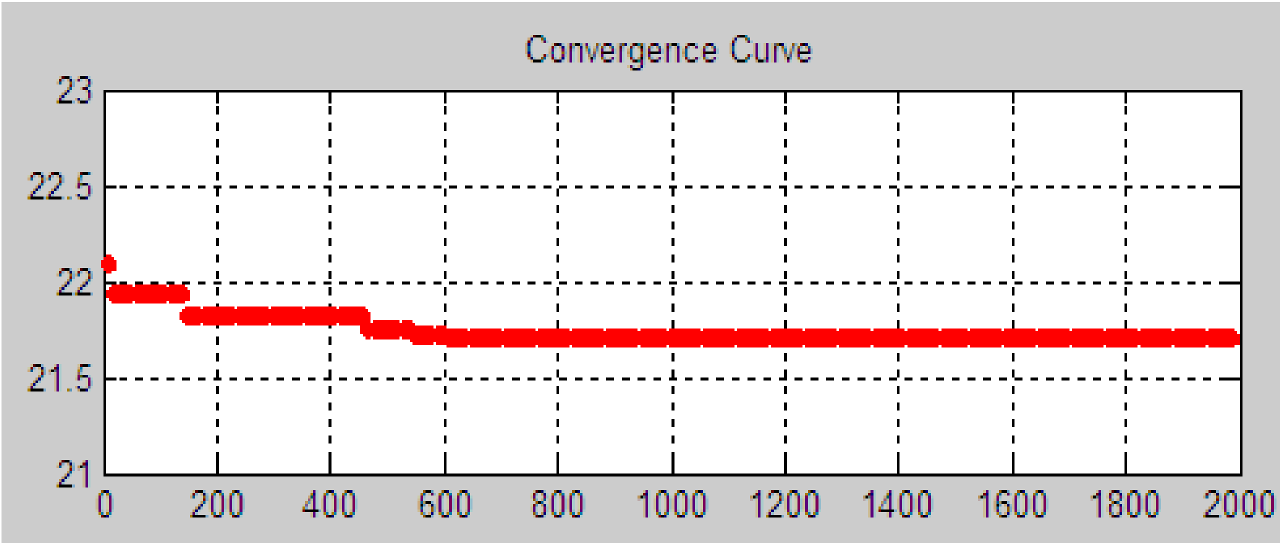


Fig. 7 Plot of function value vs iterations for carefully cut & ground metallic gears, DE with per vector dither for weightage (0.75,0.25)

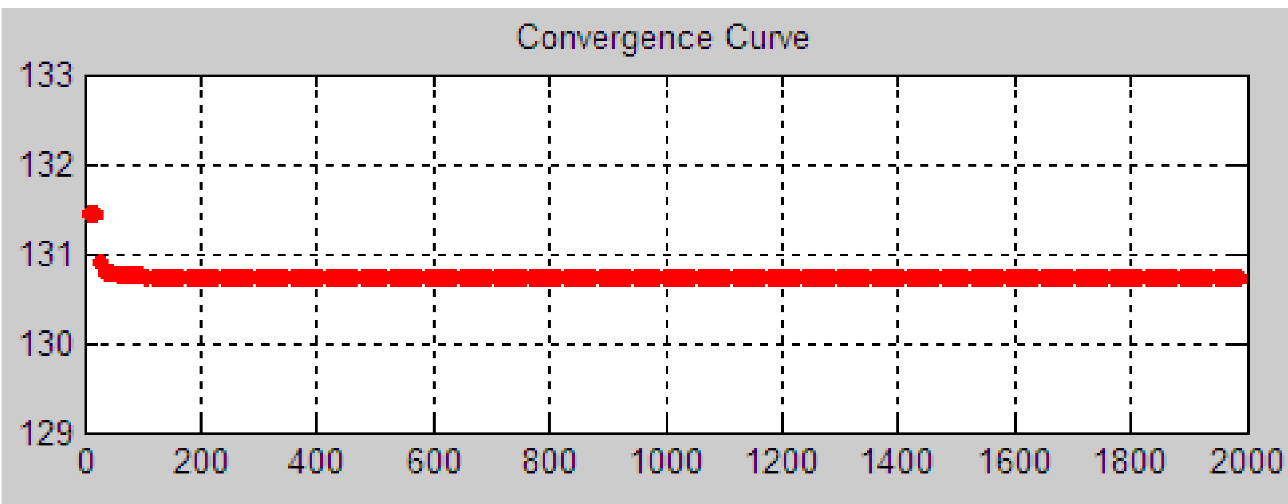


Fig. 8 Plot of function value vs iterations for gears with high precision shaped or ground teeth and if dynamic load is developed , Parent Centric DE for weightage (0.5,0.5)

DISCUSSIONS

- In the first case i.e. of ordinary cut gears it was observed that the values of objective function didn't change much when different variants of DE were used, keeping the weightage constant. The optimal values of the variables namely module, face width and no. of teeth were not affected significantly by the change of weightage and the change of variant of DE. The minimum value of objective function for the weightage (0.25, 0.75) was obtained for DE with per generation dither. The minimum value of objective function for the weightage (0.5, 0.5) was obtained for DE with per vector dither. The minimum value of objective function for the weightage (0.75, 0.25) was obtained for DE with Jitter
.
- Again it was observed, in the case of carefully cut gears, the values of the objective function didn't change much when different variants of DE were used, keeping the weightage constant. Small variation for variables in few cases were seen by the change of weightage and a change of variant of DE. The minimum value of objective function for the weightage (0.25, 0.75) was obtained for standard DE. The minimum value of objective function for the weightage (0.5, 0.5) was obtained for DE with per generation dither. The minimum value of objective function for the weightage (0.75, 0.25) was obtained for standard DE.

- Again it was observed, in the case of carefully cut & ground metallic gears, the values of the objective function didn't change much when different variants of DE were used, keeping the weightage constant. Some significant changes were observed for the variables by the change of variants and a weightage. The minimum value of objective function for the weightage (0.25, 0.75) was obtained for parent centric DE. The minimum value of objective function for the weightage (0.5, 0.5) was obtained for DE with per vector dither. The minimum value of objective function for the weightage (0.75, 0.25) was obtained for DE with jitter.

- No change in the values of the objective function was observed, in the case of gears whose teeth are finished by hobbing or shaping when different variants of DE were used, keeping the weightage constant. No change was observed for the variables by the change of variants and weightage. The minimum value of objective function for the weightage (0.25, 0.75) was found to be 159.568144. The minimum value of objective function for the weightage (0.5, 0.5) was found to be 258.012615. The minimum value of objective function for the weightage (0.75, 0.25) was found to be 356.457087.

- No change in the values of the objective function was observed, in the case of gears with high precision, shaped and ground teeth where dynamic load is developed when different variants of DE were used, keeping the weightage constant. No change was observed for the variables by the change of variants and weightage. The minimum value of objective function for the weightage (0.25, 0.75) was found to be 130.732468. The minimum value of objective function for the weightage (0.5, 0.5) was found to be 209.79783. The minimum value of objective function for the weightage (0.75, 0.25) was found to be 288.863498.

- No change in the values of the objective function was observed, in the case of gears with high precision, shaped and ground teeth where no dynamic load is developed when different variants of DE were used, keeping the weightage constant. No change was observed for the variables by the change of variants and weightage. The minimum value of objective function for the weightage (0.25, 0.75) was found to be 791362.595. The minimum value of objective function for the weightage (0.5, 0.5) was found to be 791580.951. The minimum value of objective function for the weightage (0.75, 0.25) was found to be 791799.307.

CONCLUSION

- As the weightage for weight is decreased and center-to-center distance is increased, the function value is found to increase. So it is recommended that weightage for weight should be less.

- From the number of iterations it can be confirmed that Differential evolution initially explores and then exploits. That's the reason for number of iterations to be high for a few cases.

- By varying the ranges for the variables module, face width and number of tooth, a large variety of gears can be brought into the consideration. However, these are decided on the basis of availability.

- In the above research work, the module range was specified by upper limits and lower limits. However, it can also be specified by discrete values.

- In order to obtain integer value of the number of teeth, floor() function was used. However, it is also possible to ceiling function or both the functions by applying breaks at decimal values. For e.g. if no. of teeth is greater than decimal 0.5 then use ceiling function, else use floor function.

- Crossover Probability was used as 0.5. However, higher values can also be used which would reduce the number of crossovers. But this would negatively affect the explorative behavior of DE.

- The above developed model can be used for future works involving design or selection of gears.

REFERENCES

1. Pham, D.T., & Yang, Y. (1993). Optimization of multi-model discrete functions using genetic algorithms.

2. Chong T.H., Bae, I., Park, G. J. (2002). A new and generalized methodology to design multi stage gear drives by integrating the dimensional and configuration design process. Mechanism and Machine Theory, 37, 295-310.

3. Yokota, T., Taguchi, T., & Gen, M. (1998). A solution method for optimal weight design problem of the gear using genetic algorithms. Computers & Industrial Engineering, 35(3-4), 523-526.

4. Thompson, D.F., Gupta, S., & Shukla, A. (2000). Tradeoff analysis in minimum volume design of multi-stage spur gear reduction units. Mechanisms and Machine Theory, 35, 609-627.

5. Marcelin, J.L. (2001). Genetic optimization of gears. International journal of Advanced manufacturing Technology, 179, 910-915.

6. Abersek, B., & Popov, V. (2004). Intelligent tutoring system for training in design and manufacturing. Advances in Engineering Software, 35, 461-471.

7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA–II. IEEE Transactions on Evolutionary Computation, 6, 182–197.

8. Zitzler, E., Laumanns, M., Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland.

9. Price, K.V., Storn, R. (1997). Differential evolution – a simple evolution strategy for fast optimization. Dr. Dobb's Journal, **22,** 18–24

10. Lampinen, J.: (A bibliography of differential evolution algorithm)

11. Madavan, N.K. (2002). Multiobjective optimization using a pareto differential evolution approach. In: Congress on Evolutionary Computation (CEC'2002). Volume 2, Piscataway, New Jersey, IEEE Service Center, 1145–1150

12. Xue, F., Sanderson, A.C., Graves, R.J. (2003). Pareto-based multi-objective differential evolution. In: Proceedings of the 2003 Congress on Evolutionary Computation (CEC'2003). Volume 2, Canberra, Australia, IEEE Press, 862–869