International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-04, Issue-03, June 2018

An Investigation of Genetic Algorithm for Different Population Size with an Engineering Design Problem

Arvind Gothwal, Assistant Professor, SPIT, Mehsana, Gujarat India, aur_mech@yahoo.in

Abstract- **Genetic algorithm is very helpful to fine optimum solution where many variables are in problem such that noise pollution, industrial wastage, traveling and traffic problems etc. The objective of this research is to evaluation of genetic algorithm for different numbers of populations. Genetic algorithm is use to find approximate solution for given problem and population size is very important to create search space for evaluation or find optimum solution. The effect of population size is in investigation for engineering design problem and try to find the effect of this in result collected. As we know population size generates different numbers which use in objective function and algorithm evaluate them for optimum result. In this investigation objective function changes are observed at different population size..**

Keywords — Genetic Algorithm, Population Size, Engineering Optimization, Helical Compression Spring

I. INTRODUCTION

Genetic algorithm is based on evaluation and population size is the important factor. Many researcher worked on different operators such that selection, crossover and mutation. Darrell et al [1] presented an abstract model of genetic algorithms to outperform single population models on linearly separable problems and suggested that Island Models may be at an advantage when increasing the population size does not help to solve the problem. Eiben et al [2] showed the population resizing mechanisms exhibit significant differences in speed, measured by the number of fitness evaluations to a solution and the best EAs with adaptive population resizing outperform the traditional genetic algorithm by a large margin. Siamak et al [3] tested different mutation and crossover methods on a population with a size of 100 individuals. Pedro et al [4] suggested evaluating the initial population depending of the problem solving and can choose gene-level diversity, chromosome-level diversity, population-level diversity, or a combination of those. Olympia et al [5] investigated of the influence of the population size on the genetic algorithm performance for a model parameter identification problem. Dermot et al [6] presented the choice of coding system affects convergence and gives guidance on choosing a coding system and the relationship between population size, critical schema length, problem constraints and convergence. Yong et al [7] presented the adaptive elitist-population search method, a new technique for evolving parallel elitist individuals for multimodal function optimization. The technique is based on the concept of adaptively adjusting the population size according to the individuals' dissimilarity using direction dependent elitist genetic operators. Janne et al [8] confirmed the common belief that decreasing population size increases optimization speed to a certain point, after which premature convergence slows the optimization speed down. The optimization reliability in turn usually increases monotonically with increasing population size. There were no any direct relation of population size and objective function so this work is done on observation of objective function with population size.

Here in this research an engineering problem taken and investigate results at different number of populations.

II. GENETIC ALGORITHM

Genetic algorithm is an optimization technique based on evolution. It is very useful for design optimization in engineering. Genetic algorithm is a tool which can optimize any equation with constraint or without constraints. Algorithm starts with encoding. Here a set of solution represented by chromosomes that is population. Population size is constant in all iteration. This chromosome generates new chromosomes with help different operators that are crossover and mutation. After using operators new population comes and better chromosomes or set of solution will be select for next iteration. After number of iteration algorithm converges to best solution and it may be optimal or sub optimal.

Encoding is a key to generate a set of solutions for evolution. Mostly binary codes use for this operation. Population size depends on difficulty of problem, some researcher take ten times of variables.[9] The number of bits depends on accuracy required.

If a function $[f(x_1, x_2, x_3, \dots, x_N)]$, then one chromosome represents a function cost.

101011|111011|100111|……………………………………………… …………|101101

Fig.1: Representing chromosome in binary coding.

In Figure 1, one chromosome has $(6 \times N)$ bits and 6 bits represents one variable and function depends on N

 DOI : 10.18231/2454-9150.2018.0295

variable. Accuracy of function depends on number of bits. Number of bits and number of chromosomes in population varies according to difficulty of problem. [10]

Decoding is use to convert chromosomes, binary to decimal form in a required range.
Formula used for decoding given by [10], is

$$x_i = x_i^L + \frac{x_i^U - x_i^L}{2^n - 1} \qquad (1)$$

Where n is the no. of bits used per variable,
$x_i^L \rightarrow$ Lower limit of i^{th} variable,
$x_i^U \rightarrow$ Upper limit of i^{th} variable,
$x_i \rightarrow$ Decoded value of i^{th} variable

Selection is a process to find better chromosomes from population and how many times it should be selected. Chances of selection is depends on fitness value of that chromosome. There some chromosome eliminates but population size is constant in all iteration and there are many methods for selection.

Crossover is an operator use for crossing two chromosomes and produces two new chromosomes. The new chromosomes may be better than old chromosomes. Different types of crossover can be used. These are single point, multi point and uniform crossover. Crossover rate should be high and it is probability that crossover will performed it also depends on problem.

101011|111011|100111|…… ……….|101101 (P_1)	**101**101|**110**111|**100**101|…… ……….|110010 (C_1)
001101|100111|110101|…… ……….|110010 (P_2)	001**011**|101**011**|110**111**|…… ……….|**101101** (C_2)

Fig. 2: Showing new chromosomes produced for evaluation.

Where C_1, C_2 are children produced by parents P_1, P_2. [10]

Mutation is an operator which adds new information at random way. Randomly selects new chromosomes and applies mutation. It removes local optima and it help to generates global minima. In binary coding it converts 0 to 1 and 1 to 0. Mutation gives new chromosomes for evolution in random way. It may be single point or multi-point and its probability should be low.

III. PROBLEM FORMULATION

Objective function

$$W = \rho \times N_t \times (\pi \times D) \times (\frac{\pi}{4} \times d^2) \qquad (2)$$

Where N_t, d *and* D *are* total number of coil, diameter of spring wire and diameter of coil respectively. Weight directly depends on these variables but these variables also must be satisfied in constraint equations for this we establish these constraints.

CONSTRAINTS

Condition of Coil Not Touch

In helical spring, when force applies on spring then spring compress but coils must not touch under maximum load condition so

Term	Plain	Plain and Ground	Squared or Closed	Squared and Ground
End coils,	0	1	2	2
Total coils,	N_a	$N_a + 1$	N_{a+2}	$N_a + 2$
Free length,	$pN_a + d$	$p(N_a + 1)$	$pN_a + 3d$	$pN_a + 2d$
Solid length,	$d(N_t + 1)$	dN_t	$d(N_t + 1)$	dN_t

Table 3 represents different lengths of springs according to end condition and number of end coils. [11]

$$free\ length - solid\ length > maximum\ diflection$$

$$l_f - l_s > \delta_{max} \qquad (3)$$

$$p = 2D\tan\left(\frac{\beta}{2}\right) \text{ and } \delta_{max} = \frac{8F_{max}D^3n}{Gd^4}$$

Where $\delta_{max}$is maximum deflection, 'p' is pitch and β is helix angle (β should be less than15^o)[11]

3.1.2Condition of Critical Frequency- When helical spring uses at where high rapid reciprocating motion than there will be chances of resonance. For safety from resonance problem, fundamental critical frequency of helical spring must be at least 15 to 20 times greater than frequency of applied force [11]

$$f_c > 20f_{force} \qquad (4)$$

Condition	f_c(critical frequency)
one end against a flat plate and the other end free	$\frac{1}{4}\sqrt{\frac{S}{m}}$
one end at flat plate and other is free	$\frac{1}{2}\sqrt{\frac{S}{m}}$

Table 4: Representing critical frequency of helical compressive spring according to mounting condition.[7]

$$S = \frac{d^4G}{8D^3n} \text{ and } m = \rho \times \frac{\pi}{4} \times d^2 \times \pi Dn$$

Where 'S' is stiffness of spring and 'm' is mass of spring

3.1.3Condition of Buckling- Buckling is also a problem, it depends on ratio of free length of spring to coil diameter of spring and if we increase the length of spring and uses as a compressive spring then there will be high chances of buckling so it must be maintain a ratio for prevent buckling.[11]

 DOI : 10.18231/2454-9150.2018.0295

ISSN : 2454-9150 Vol-04, Issue-03, June 2018

$$l_f < \frac{\pi D}{\alpha} \times \sqrt{\frac{2(E-G)}{2G+E}} \qquad (6)$$

Where $'\alpha', 'E', 'G'$ are End-condition constant, Young's modulus and Modulus of Rigidity respectively.

End Condition	α
Spring supported between flat parallel surfaces (fixed ends)	0.5
One end supported by flat surface perpendicular to spring axis (fixed),other end pivoted (hinged)	0.707
Both ends pivoted (hinged)	1
One end clamped; other end free	2

Table 5: Representing value of End condition constants according to end conditions.[11]

3.1.4 Condition of Fatigue Loading- At some applications spring used for millions number of cycles such that automotive engine, cam and follower etc. so spring subjected to variable stress and there are large chances of fatigue failure and we must check for fatigue and static stress, if F_{max} and F_{min} are respectively maximum and minimum force applied on spring then

$$F_a = \frac{F_{max}-F_{min}}{2}, \qquad F_m = \frac{F_{max}+F_{min}}{2}$$

$$\tau_a = K_B \times \frac{8F_a D}{\pi d^3}, \qquad \tau_m = K_s \times \frac{8F_m D}{\pi d^3}$$

$$K = \frac{4C-2}{4C-4} + \frac{0.615d}{D} \text{ and } K_s = 1 + \frac{1}{2C}$$

Where F_m is mean force, F_a is force amplitude, K_s is shear stress correction factor and K is Whal correction factor. τ_a and τ_m are stress produced for F_a and F_m respectively. Factor of safety for helical spring should be greater than 1.5[11]

$$fos = \frac{0.5 \times S'_{se} \times S_{sy}}{\tau_a S_{sy} + 0.5\tau_m S'_{se} - 0.5\tau_a S'_{se}} > fos_{desired} \qquad (7)$$

Where S'_{se}-torsional shear stress, S_{sy} -torsional yield strength and S_{ut}-ultimate tensile strength

Material	S_{sy}
Music wire and cold-drawn carbon steel	$0.45S_{ut}$
Hardened and tempered carbon and low-alloy steel	$0.50S_{ut}$
Austenitic stainless steels	$0.35S_{ut}$
Nonferrous alloys	$0.35S_{ut}$

Table 6: Representing torsional yield strength according to type of material.[11]

Spring Index

Spring index of helical spring should be in a range so we can say $C_{min} < C < C_{max}$[8]

$$C_{min} - \frac{D}{d} < 0 \qquad (8)$$

$$\frac{D}{d} - C_{max} < 0 \qquad (9)$$

Number of populations	No. of bits in one chromosome	Type of coding	Type of cross over	Cross over probability	Probability of mutation	Type of selection	Number of Iterations
60	10	Binary	Single point cross over	0.8	0.06	Rolette Wheel	200

Table 7 Representing Data selected for Genetic Algorithm

$E(Mpa$	$G(MPa$	$\rho(Kg/mm^3$	α	β	$S_{sy}(MP$	$S'_{se}(MP$	$S_{ut}(MP$	$f(Hz$
207×10^3	81370	$7.8 \times 10^{-}$	0.707	12^o	$0.45S_{ut}$	$0.22S_{ut}$	1440	25

Table 8

$d(mm)$	$D(mm)$	n	$F_{max}(N$	$F_{min}(N)$	C_{max}	C_{min}	fos
2 to 15	20 to 35	3 to 10	350	100	10	6	1.5

Table 9

Table 8 and Table 9 representing input parameters for design of spring.

Number of populations	Number of Runs					
	1st Run	2nd Run	3rd Run	4th Run	5th Run	**Avg. of 5**
10	66.7	71.8	68.2	69.8	59.5	**67.2**
20	78.1	96.3	76.1	60.0	71.6	**76.4**
30	65.2	68.2	71.	59.7	60.6	**65.0**
40	74.6	77.9	92.9	63.9	71.5	**76.2**
50	59.9	63.7	75.6	60.3	61.3	**64.2**
60	60.1	64.8	96.0	59.1	59.4	**67.9**
70	58.5	62.6	61.9	59.9	60.1	**60.6**
80	77.7	58.4	101.0	65.5	89.6	**78.5**
90	95.7	59.9	64.1	70.9	77.7	**73.7**
100	68.2	61.6	59.0	60.1	58.6	**61.5**
110	70.8	59.4	70.9	64.2	74.6	**68.0**
120	95.4	59.3	58.4	62.5	61.9	**67.5**
130	63.9	64.5	90.2	59.4	60.7	**67.8**
140	59.3	58.6	58.8	58.7	63.9	**59.9**
150	84.7	59.5	59.6	58.8	58.9	**64.3**
160	60.8	59.1	58.8	58.5	64.9	**60.4**
170	61.0	101.8	70.7	58.5	62.9	**70.8**
180	61.4	63.3	70.8	58.8	59.1	**62.7**

Table 10 Representing Average Results w.r.t.. Numbers of Population

DOI : 10.18231/2454-9150.2018.0295

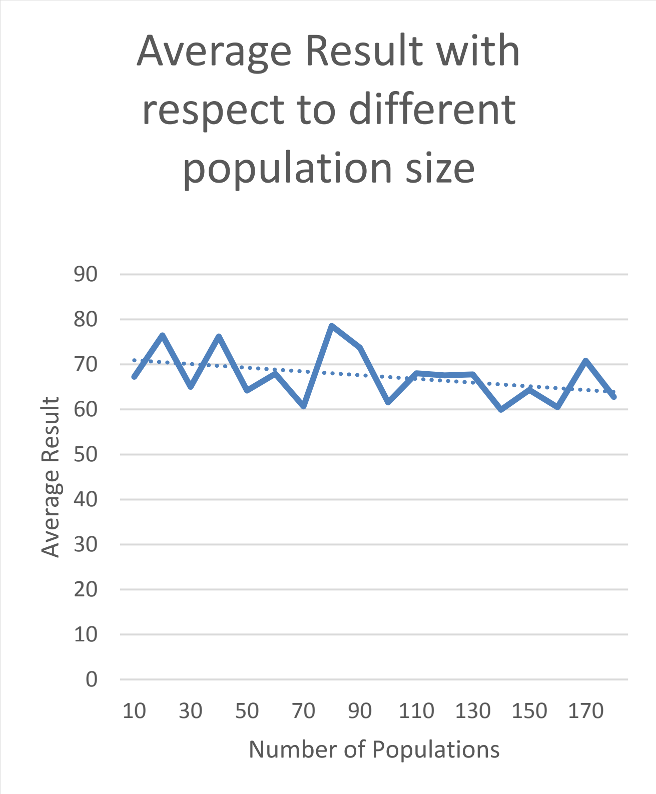

Fig.3 Graphical Representation of Average Results

IV. Results and Discussion

There, results are calculated at different Number of population and it is found that population size has very important role to create space. GA is based on evaluation and here more chances to find better solution with large population. Population size change from 10 to 180 and here we can see as population size increases better results come. The optimization problem is a minimization problem and results are minimized as population increased. Here five run done on same population size and then average result is taken to easily understand the effect of population size. Here large number of population size is suggested to get better results but when it goes high then algorithm time increase so it should be 10 to 20 times of number of variables. The algorithm results are find out for 150 numbers of generations to investigate effect of population size when it increases result be approximately same for all population size. This result shows that population size is very important tool to create variables values for evaluation. This is very help full to handle many engineering design, management of materials, creating time table of railway and buses, noise pollution, hydro and thermal plants, and various problems where we need to save material, money, time, energy etc.

References

[1] Darrell Whitley, Soraya Rana, Robert B. Heckendorn, “The Island Model Genetic Algorithm: On Separability, Population Size and Convergence”, Department of Computer Science Colorado State University November 4, 1998

[2] Eiben A.E., E. Marchiori V.A. Valk, “Evolutionary Algorithms with on-the-y Population Size Adjustment”,Department of Arti_cial Intelligence, Vrije Universiteit Amsterdam

[3] Siamak Sarmady, “An Investigation on Genetic Algorithm Parameters” School of Computer Sciences, Universiti Sains Malaysia, Penang, Malaysia

[4] Pedro A. Diaz-Gomez and Dean F. Hougen, “Initial Population for Genetic Algorithms: A Metric Approach” School of Computer Science University of Oklahoma Norman, Oklahoma, USA

[5] Olympia Roeva, Stefka Fidanova, Marcin Paprzycki, “Influence of the Population Size on the Genetic Algorithm Performance in Case of Cultivation Process Modelling” , Proceedings of the 2013 Federated Conference on Computer Science and Information Systems pp. 371–376

[6] Dermot W. O'Dwyer1 & Eugene J. O’Brien, “Genetic Algorithm Encoding Probabilities & Population Size”, Transactions on Information and Communications Technologies vol 20, © 1998 WIT Press,ISSN 1743-3517

[7] Yong Lianga, Kwong-Sak Leungb, “Genetic Algorithm with adaptive elitist-population strategies for multimodal function optimization”Applied Soft Computing 11 (2011) 2017–2034 ScienceDirect

[8] Janne Koljonen and Jarmo T. Alander, “Effects of population size and relative elitism on optimization speed and reliability of genetic algorithms”, Department of Electrical Engineering and Automation, University of Vaasa

[9] Kalylanmoy Deb., 1995, "Optimization for Engineering Design Algorithms and Examples", Prentice-Hall of India Private Limited, New Delhi.

[10] David E. Goldberg “Genetic Algorithms”, Pearson Education India, 2006.

[11] Richard Budynas, Keith Nisbett “Mechanical Engineering Design”, Tata Mcgraw Hill, Education Private Limited Eighth edition.

[12] Arthur Munzenmaier Wahl, “Mechanical springs” McGraw-Hill, 1963.

 DOI : 10.18231/2454-9150.2018.0295