Advances in Computational Sciences and Technology
ISSN 0973-6107 Volume 10, Number 8 (2017) pp. 2201-2211
© Research India Publications
http://www.ripublication.com

Genetic Algorithm: An Efficient Tool for Global Optimization

[1]Sushruta Mishra
Dept. of Computer Science and Engineering, CVRCE, India.

[2]Soumya Sahoo
Dept. of Computer Science and Engineering, CVRCE, India.

[3]Mamata Das
Dept. of Computer Science and Engineering, CVRCE, India.

Abstract

The genetic algorithm (GA) is a search heuristic that is routinely used to generate useful solutions for optimization and search problems. It generates solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. Genetic algorithms are one of the best ways to solve a problem for which little is known. They are very general algorithms and so efficient in any search spaceThus they can be implemented as a global optimization tool in analyzing massive data sets.

Keywords: Search Space, Mutation, CrossOver, Global optimization

1. INTRODUCTION

Genetic Algorithms (GA) are direct, parallel and stochastic method for global search and optimization that imitates the evolution of the living beings which was described by Charles Darwin. GA is the part of the group of Evolutionary Algorithms (EA). The evolutionary algorithms use the three main principles of the natural evolution: reproduction, natural selection and diversity of the species that maintained by the differences of each generation with the previous. Genetic Algorithms works with a set of individuals, representing possible solutions of the task. The selection principle is applied by using a criterion, giving an evaluation for the individual with respect to the desired solution. The best-suited individuals create the next generation. The large variety of problems in the engineering sphere, as well as in other fields, requires the usage of algorithms from different type, with different characteristics and settings.

1.1 Main ingredients of GA

1.1.1 Chromosomes

During the division process of the human cells the chromatin (contained in the nucleus and built from DNA (deoxyribonucleic acid), proteins and RNA (ribonucleic acid)) become shorter and thicker and forms spiral strings – chromosomes. Chromosomes are the genes which carry the inherited information from parent to children. Every gene codes particular protein and is independent factor of the genetic information, which determines the appearance of different peculiarities.

For the genetic algorithms, the chromosomes represent set of genes, which code the independent variables. Every chromosome represents a solution of the given problem. Individual and vector of variables will be used as other words for chromosomes. In other hand, the genes could be Boolean, integers, floating point or string variables, as well as any combination of the above. A set of different chromosomes (individuals) forms a generation. By means of evolutionary operators, like selection, recombination and mutation an offspring population is created.

1.1.2 Selection

In the nature, the selection of individuals is performed by survival of the fittest. The more one individual is adapted to the environment - the bigger are its chances to survive and create an offspring and thus transfer its genes to the next population.

In EA the selection of the best individuals is based on an evaluation of fitness function or fitness functions. Examples for such fitness function are the sum of the square error between the wanted system response and the real one; the distance of the poles of the closed-loop system to the desired poles, etc. If the optimization problem is a minimization one, than individuals with small value of the fitness function will have bigger chances for recombination and respectively for generating offspring.

2. RECOMBINATION

The first step in the reproduction process is the recombination (crossover). In it the genes of the parents are used to form an entirely new chromosome. The typical recombination for the GA is an operation requiring two parents, but schemes with more parent's area also possible. Two of the most widely used algorithms are Conventional (Scattered) Crossover and Blending (Intermediate) Crossover.

2.1 Conventional (Scattered) Crossover

In this recombination type, the parents exchange the corresponding genes to form a child. The crossover can be single or multipoint as shown in Fig. 2.1. For the recombination a bit Mask is used. The equations describing the process are:

C1=Mask1&p1+Mask2&p2

C2=Mask2&p1+Mask1&p2

P1, P2 – parent's chromosomes

C1, C2 – children's chromosomes

Mask1, Mask2: bit masks

Mask2 = NOT (Mask1)

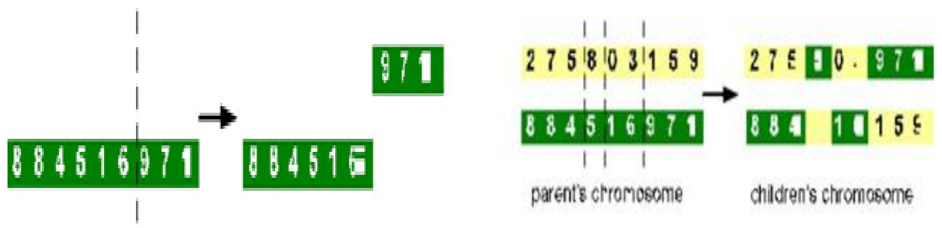
& : bit operation "AND".

Mask1 = [1 1 1 0 1 1 0 0 0]

Mask2 = NOT (Mask1) = [0 0 0 1 0 0 1 1 1]

P1= [2 7 5 8 0 3 1 5 9]

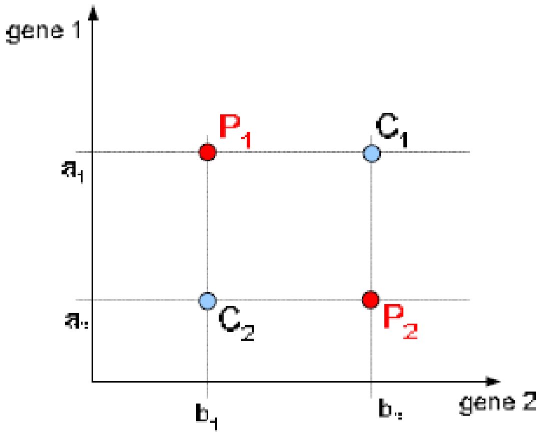
P2 = [8 8 4 5 1 6 9 7 1]



a) Single point crossover b) Multipoint crossover

(Fig. 2.1 – Crossover with bit mask)

Geometric representation of this type of crossover of a chromosome with two genes is shown at Fig. 2.2. This crossover type (with bit mask) could be used with all gene types listed above



(Fig. 2.2)

Genes in n population n(parent genes)

P1=[a1,b1],P2=[a2,b2]

Mask=[1,0]

Genes in population n+1: (child genes)

C1=[a2,b1],C2=[a1,b2]

2.1 Blending (Intermediate) crossover

The mathematic description of this crossover is:

$C_1 = \gamma.P_1 + (1- \gamma)$

$C_2 = (1- \gamma) .P_1 + \gamma$

$C_2 = (1+2.\alpha).r - \alpha$

P1, P2 – chromosomes of the parents

C1, C2 – chromosomes of the children (offspring individuals)

r – random number between 0 and 1

The graphical representation is shown on Fig. 2.3 and Fig. 2.4.

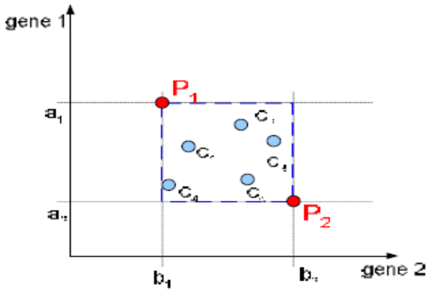


Fig. 2.3 Graphical representation of blending crossover α=0

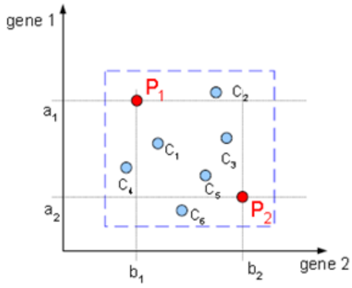


Fig. 2.4 Graphical representation of blending crossover α>0

The coefficient α allows the user to change the area in which the value of the resultant (offspring) gene can appear. When $_{\gamma}=0$ it is guaranteed that the value of the resultant gene is between the values of the corresponding genes of the parents. When the value of α is above 0, neighbor areas could be explored Figure 2.5.

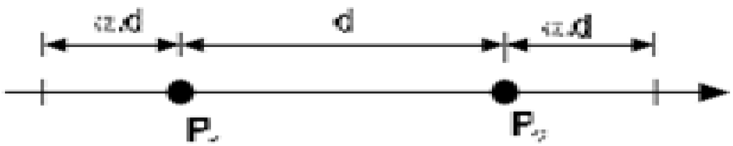


Fig. 2.5 Change of the search diapason at change of α

2.2 Mutation

The newly created by means of selection and crossover population can be further applied to mutation. Mutation means, that some elements of the DNA are changed. Those changes are caused mainly by mistakes during the copy process of the parent's genes.

In the terms of GA, mutation means random change of the value of a gene in the population Fig.2.6 a). The chromosome, which gene will be changed and the gene itself are chosen by random as well Fig. 2.6 b).

a) mutation in a chromosome

b) mutation places in the population

Fig.2.6. Mutation in the genetic algorithm

3. SCHEMES OF THE EVOLUTIONARY ALGORITHMS

The EA holds a population of individuals (chromosomes), which evolve by means of selection and other operators like crossover and mutation. Every individual in the population gets an evaluation of its adaptation (fitness) to the environment. In the terms of optimization this means, that the function that is maximized or minimized is evaluated for every individual. The selection chooses the best gene combinations (individuals), which through crossover and mutation should drive to better solutions in the next population. One of the most often used schemes of GA is shown on Fig.3.

1. Generate initial population: In most of the algorithms the first generation is randomly generated, by selecting the genes of the chromosomes among the allowed alphabet for the gene. Because of the easier computational procedure it is accepted that all populations have the same number (N) of individuals.

2. Calculation of the values of the function that we want to minimize of maximizes.

3. Check for termination of the algorithm: As in the most optimization algorithms, it is possible to stop the genetic optimization by:
   - Value of the function: The value of the function of the best individual is within defined range around a set value. It is not recommended to use this criterion alone, because of the stochastic element in the search the procedure, the optimization might not finish within sensible time.
   - Maximal number of iterations: This is the most widely used stopping criteria. It guarantees that the algorithms will give some results within some time, whenever it has reach extremum or not;
   - Stall generation: If within initially set number of iterations (generations) there is no improvement of the value of the fitness function of the best individual the algorithms stops.

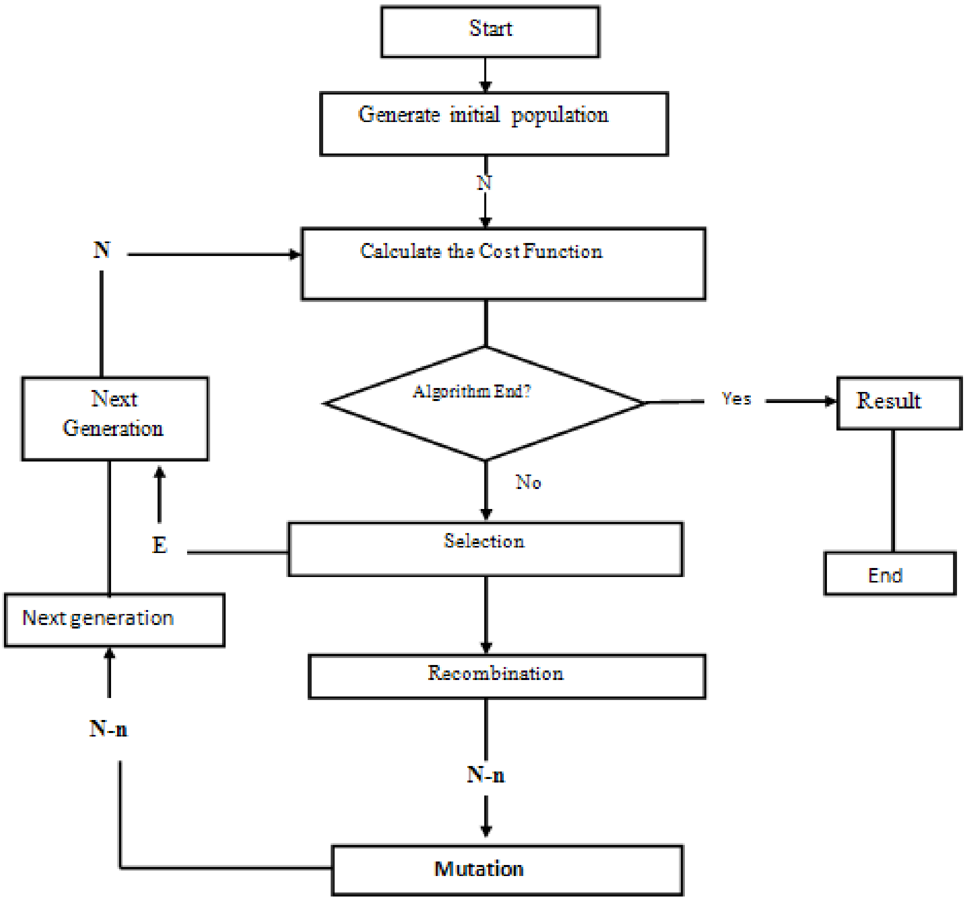


Fig. 3

Selection – between all individuals in the current population are chose those, who will continue and by means of crossover and mutation will produce offspring population. At this stage elitism could be used – the best n individuals are directly transferred to the next generation. The elitism guarantees, that the value of the optimization function cannot get worst (once the extremum is reached it would be kept). Crossover in which the individuals chosen by selection recombine with each other and new individuals will be created. The aim is to get offspring individuals that inherit the best possible combination of the characteristics (genes) of their parents.

Mutation which means of random change of some of the genes, it is guaranteed that even if none of the individuals contain the necessary gene value for the extremum, it is still possible to reach the extremum. New generation – the elite individuals chosen from the selection are combined with those who passed the crossover and mutation, and form the next generation.

4. GLOBAL OPTIMIZATION BY GENETIC ALGORITHM

Optimization is the science of finding decisions that satisfy given constraints, and meet a specific goal at its optimal value. In engineering, constraints may arise from physical limitations and technical specifications. In business, constraints are often related to resources, including manpower, equipment, costs, and time.

The objective of global optimization is to find the "best possible" solution in nonlinear decision models that frequently have a number of sub-optimal (local) solutions. In the absence of global optimization tools, engineers and researchers are often forced to settle for feasible solutions, often neglecting the optimum values. In practical terms, this implies inferior designs and operations, and related expenses in terms of reliability, time, money, and other resources.

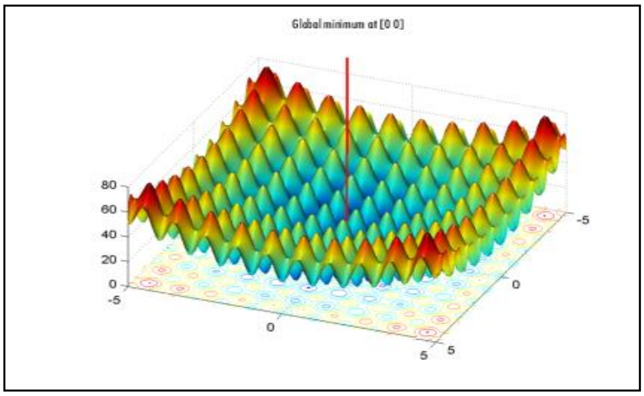
The classical optimization techniques have difficulties in dealing with global optimization problems. One of the main reasons of their failure is that they can easily be entrapped in local minima. Moreover, these techniques cannot generate or even use the global information needed to find the global minimum for a function with multiple local minima.

The genetic algorithm solves optimization problems by mimicking the principles of biological evolution, repeatedly modifying a population of individual points using rules modeled on gene combinations in biological reproduction. Due to its random nature, the genetic algorithm improves the chances of finding a global solution. Thus they prove to be very efficient and stable in searching for global optimum solutions. It helps to solve unconstrained, bound-constrained, and general optimization problems, and it does not require the functions to be differentiable or continuous.

We next discuss an example that shows how to find the global minimum of Rastrigin's function using genetic algorithm. For two independent variables, Rastrigin’s function is defined as:

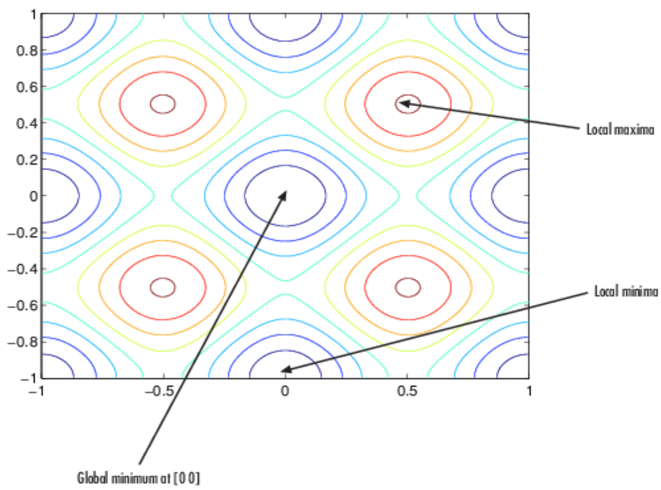
$$Ras(x) = 20 + x_1^2 + x_2^2 - 10(\cos 2\pi x_1 + \cos 2\pi x_2).$$

Global Optimization Toolbox software contains the rastriginsfcn.m file, which computes the values of Rastrigin's function. The following figure shows a plot of Rastrigin's function.

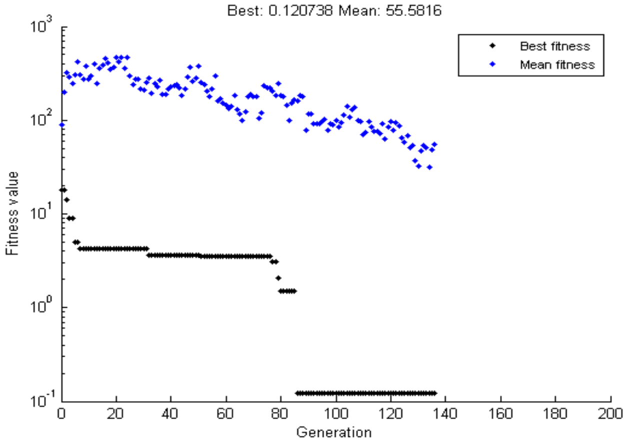


As the plot shows, Rastrigin's function has many local minima—the "valleys" in the plot. However, the function has just one global minimum, which occurs at the point [0 0] in the x-y plane, as indicated by the vertical line in the plot, where the value of the function is 0. At any local minimum other than [0 0], the value of Rastrigin's function is greater than 0. The farther the local minimum is from the origin, the larger the value of the function is at that point.

Rastrigin's function is often used to test the genetic algorithm, because its many local minima make it difficult for standard, gradient-based methods to find the global minimum.The following contour plot of Rastrigin's function shows the alternating maxima and minima.



The final value of the fitness function (Rastrigin's function) when the algorithm terminated: Objective function value 0.05531602101322264. The value obtained is very close to the actual minimum value of Rastrigin's function, which is 0.



5. CONCLUSION

The problem of finding the global optimum in a space with many local optima is a classic problem for all systems that can adapt and learn. GA provides a comprehensive search methodology for optimization. GA is applicable to both continuous and discrete optimization problems. In global optimization scenarios, GAs often manifests their strengths: efficient, parallelizable search; the ability to evolve solutions with multiple objective criteria; and a characterizable and controllable process of innovation.

6. FUTURE TRENDS

Some limitations of GAs are that in certain situations, they are overkill compared to more straightforward optimization methods such as hill-climbing, feed forward artificial neural networks using back propagation, and even simulated annealing and deterministic global search. To make genetic algorithm more effective and efficient it can be incorporated with other techniques within its framework to produce a hybrid genetic algorithm that can reap best from its combination. Hybrid genetic algorithms have received significant interest in recent years and are being increasingly used to solve real-world problems quickly, reliably and accurately without the need for any forms of human intervention. More research needs to be concentrated on the development of hybrid design alternatives for its efficiency enhancement.

REFERENCES

[1] Genetic algorithms for optimization – application in the controller synthesis task – Popov A., diploma thesys, department Systems and Control, faculty Automatics, Technical University Sofia, 2003

[2] An Empirical Study of Evolutionary Techniques for Multiobjective Optimization in Engineering Design – Coello C., Department of Computer Science, Tulane University

[3] The Role of Mutation and Recombination in Evolutionary Algorithms – Spears William, dissertation for Doctor of Philosophy at George Mason University

[4] A survey of Multiobjective Optimization in Engineering Design - Johan Andersson, Department of Mechanical Engineering, Linköping University, Sweden

[5] Comparison of Two Multiobjective Optimization Techniques with and within Gentec Algorithms – Azar S, Reynolds B., Narayanan S, Department of Mechanical Engineering, University of Maryland

[6] PDE: A Pareto–Frontier Differential Evolution Approach for Multi-objective OptimizationProblems, Hussein A. Abbass, Sarker R., Newton C., School of Computer Science, University of New South Wales, University College, Canberra, Australia

[7] An Analysis of Multiobjective Optimization within Genetic Algorithms - Bentley P.,Wakefield J., Division of Computing and Control Systems Engineering, The University of Huddersfield The University of Huddersfield

[8] Wing Design Using Evolutionary Algorithms - Akira Oyama, Department of Aeronautics and Space Engineering of Tohoku University

[9] Non-linear Goal Programming Using Multiobjective Genetic Algorithms - Kalyanmoy Deb.,Kanpur Genetic Algorithms laboratory (KanGAL)Department of Mechanical Engineering, Indian Institute of Technology, Kanpur, India.

[10] Genetic Algorithms Applied to Real Time Multiobjective Optimization Problems - Z. Bingul, A. Sekmen, S. Palaniappan, S. Sabatto, Tennessee State University, Nashville, USA

[11] A genetic algorithm with adaptable parameters - D. Quagliarella, A. Vicini, C.I.R.A., Centro Italiano Ricerche Aerospaziali, Via Maiorise, Capua, Italy

[12] Practical Optimization, Gill Ph., Murry W., Wright M., Academic Press, 1981