International Journal of Computer Applications (0975 – 8887)
Volume 119 – No.10, June 2015

A Paper on Multiple Objective Functions of Genetic Algorithm

Anju Bala
Department of Computer Science & Applications
Maharshi Dayanand University, Rohtak

Rajender Singh Chhillar, PhD
Department of Computer Science & Applications
Maharshi Dayanand University, Rohtak

ABSTRACT

Creating or preparing Multi-objective formulations are a realistic models for many complex engineering, AI, mathematical and optimization problems etc. Customized genetic algorithms have been expressed as effective to determine best solutions to these problems. In many real-life problems, there are many conflicts to each other towards objective, and mainly by taking single objective to optimizing a particular solution can give unacceptable result with respective to other objective. An inevitable features of Genetic Algorithm are to generate set of solutions for multi objective problem with satisfying objective at acceptance level without dominating to any other solution. Genetic Algorithm is used in maximization as well as minimization of function. This paper tried to show an overview and tutorial is presented how Genetic Algorithm is best to solve the maximization of function for given function. This feature make Genetic Algorithm very unique from traditional genetic algorithms. Roulette-Wheel selection method is adopted to calculate fitness function and other functions.

Keywords

Genetic Algorithm, optimization and its techniques, Multi-objective functions, conclusion.

1. INTRODUCTION

The objective of this paper to present an overview of multiple-objective optimization methods using genetic algorithms (GA). Genetic Algorithm can be used for multiple-objective problems in several variables. Here we want to minimize two objectives, each having one decision variable as well as calculate maximum value of given function. Multiple objective problems generally are very conflicting by nature, but GA prevents the simultaneous optimization of each objective.

In many areas, or even most, real engineering problem i.e. in software engineering actually it have many objective i.e. minimize cost & time, maximize performance, maximize reliability, maintaining integrity etc. These are so challenging but realistic problems. In present era, GA becomes a popular to solve such type of optimization problem. GA is meta-heuristic searching technique that is particularly well-suited for this kind of problems. Traditional searching techniques are not suitable for such class of prob:lem i.e hill climbing, tabu search, memetic algorithm etc, are fully or partially local search techniques, stuck at local maxima. GA is global searching technique ,customized to accommodate the best optimize solution for a optimization problem by using specialized fitness functions, introducing operators to promote solution diversity.

2. CONCEPT OF GENETIC ALGORITHM

Genetic algorithm fall under the category of heuristic search techniques. Gas is based on the natural phenomenon i.e concept of GA is “Natural Selection” and “Genetic inheritance”. This idea proposed by father of GA is **John Holland, university of Michigan (1970) in U.S.A**. It is based on the evolutionary principle that inspired by Darwin,^{s} theory “Survival Of Fittest”. GA exploits the random search to solve the optimization problem i.e. works on randomly generated population. Select best individual using fitness function i.e. close to objective by suppressing the weak, like nature stronger individual suppress the weaker ones. Genetic algorithms have most challenging operators to mimicking the natural process: Selection, Crossover, Mutation and recombination, used to solution of a problem.

3. WHY GA ADOPT

- It gives better result than AI, it is more robust.
- Unlike older technique, by slightly changes in inputs. GA do not break easily, until no reasonable noise.
- It gives the significant results, while performing search in large search space or search state or multi model than other searching techniques like DFS, BFS, linear programming etc.

“Genetic Algorithm is good at taking large, potentially huge search spaces and navigating them, looking for optimal combinations of things, the solutions one might not, otherwise find in the lifetime.”

(Salvatore Mangano Computer Design, May 1995)

Meaning of optimization

Optimization (make the best use of something) is a process to find the best or optimum solution of given problem. The optimization problem mainly rely or focused on three factors i.e.

1) **An objective function** is used to maximize or minimize to a given function.

Example

# In **SDLC,** we want to cover every aspect of testing objective with minimum time and cost.
# In **production**, we want to maximize profit with minimum cost.

2) **A set of variables** used in objective function to solve the objective problem.

   # In SDLC, set of test cases or time used in software testing.

   # In production, raw material or time & cost to prepare a product.

3) **A set of constraints** is used to set the criteria on variables to take value but exclude others.

   # In SDLC, give negative output for wrong input, & positive output for right input.

   # In production, within time limit show non-negative value.

   So, optimization problem may be defined as , finding the value of given variables to maximize or minimize the objective function while satisfying the given constraints.

Optimizations techniques:

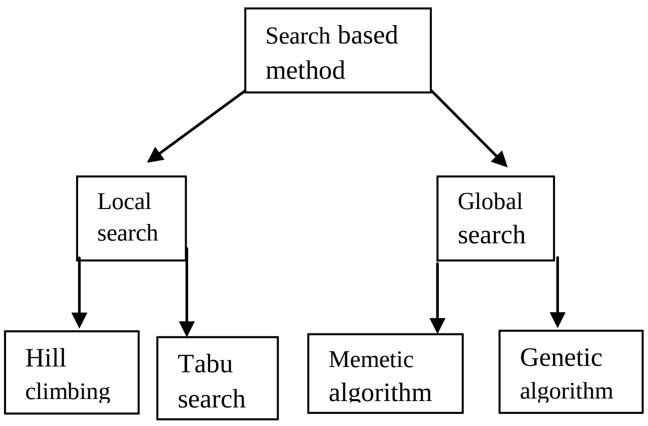

Fig.1. optimization techniques

Main operators used by GA to generate new offsprings from older ones: *Crossover* and *Mutation*.

The crossover operator is the most important operator of GA. In crossover, select two chromosomes from given population to recombination to generate new offspring. These two chromosomes selected on the basis of its fitness function. The intent behind crossover to get fittest individual or make the fittest parents. Crossover operator use iterative process, expected to to appear best chromosome frequently in the population to overall good solution. Crossover probability can be high or any point(single point, two point or uniform etc.)

Mutation operator takes place after crossover to maintain the diversity of population. Mutation probability should be less i.e. approximate 0.01, is a best choice. In fact, mutation applied at gene level randomly. Mutated chromosomes will not be very different from original population. if mutation rate set high, It will take totally other side of solution. So, mutation play very important role in GA, prevent the population from stagnating at local minima. Mutation can be bit-flip, uniform, non-uniform etc..

Recombination involves selection of chromosomes for the next generation. And again calculate the fitness function until a stopping criteria is met , or fittest individual is met.

There are different selection strategies in GA, depending on problems or fitness value i.e. Roulette-wheel selection, tournament selection, boltzman selection etc. In this paper, Roulette-Wheel Selection is used to find the maximum value of given function.

4. MULTI-OBJECTIVE GENETIC ALGORITHMS

GA is the best optimization technique to solve the multi-objective optimization problems. GA can find set of solution The challenging part of GA to simultaneously search different regions of a solution space makes it acceptable to search a varied set of outcomes for

non-trivial problems with non-convex, discontinuous, and multi-modal solutions spaces. . In GAs operator, crossover is most valuable feature especially where building blocks (i.e schemas) exchange is necessary [Arabas94].crossover is problem dependent, i.e. its implementation and representation is closely related to problem domain and provide new non-dominated solutions in unexplored parts of the Pareto front. In present time, GA has been most popular heuristic approach to multi-objective or multi-model solution without requiring user prioritize.

Jones et al. [25] reported that 90% of the approaches to multi-objective optimization aimed to approximate the true Pareto front for the underlying problem.

The procedure of a generic GA is given as follows:

Step 1.Initiate with the generation of population randomly, *P*1. Evaluate

the fitness of each chromosome in *P*1.

Step 2. **Selection**: choose the parent chromosomes from the population, *p1*.

Step 3. **Crossover**: randomly crossover the pair according to crossover probability, *pc(probability can be at high rate).*

Step 4. **Mutation:** Mutate two offspring ,maintaining mutation rate, *pm*(mutation rate should be less, 0.01).

Step 5. **Fitness function:** Evaluate fitness of each chromosomes rely on objective function.

Step 6. Select the new population ,replacing current population, stopping criteria not met, go to step 2.

Problem solving using Genetic Algorithm

Problem: ***maximizing a Function***

i.e. $f(x)=x^2+3x=5$

where x can vary between 0 to 31.

To solve the problem steps are given below:

Step1: first, encode the variable ”x” into a finite length string.

Here, five bit(binary integer) unsigned integer, range between 0(00000) to 31(11111).

Here, two generation of GA is performed to maximize the objective function with encoding, selection, crossover and mutation. The algorithm starts with randomly generated population. Initial generated population size is 4. But it can be differ from requirement to requirement.

Step2: decoding the x value for each string for the initial population(i.e. 4 string in given population).

Step 3: evaluate the fitness function by putting the value of x.

Step 4: calculate the probability of selection,

$$Pi= F(x)_i / \Sigma F(x)_i$$

Where f(x) is the fitness value

Σ F(x) is summation of all fitness value.

Step 5: calculate percentage probability.

Step 6: In next step calculate expected count, as

Expected count = $f(x)_i /[Avg\ f(x)]_i$

Where

$$(Avg\ f(x))_i = [\Sigma f(x)_i /n]$$

Same procedure follow to compute expected count for entire population. It helps in selection for further mating pool.

Step 7: now calculate actual count for individual for participating in crossover using Roulette-Wheel Selection strategy.

Roulette-Wheel Selection strategy performed like:

Roulette –wheel covers 100% and according to step 4 (probability or probability % of individuals) fits into roulette wheel, wheel spun and numbers of occurrence of population is noted to actual count. Actual count are tabulated, more te actual count, more chance of selection for crossover.

Step 8: crossover operator is performed to produce new individual, single point crossover is performed to produce new offspring.

Step 10: After crossover, decode the new string and calculate the fitness function.crossover probability assumes 1.0.

Step 11: In last step, mutation flipping operator is performed by bit by bit with 0.01 mutation rate. Now a new population is ready to be tested, decoding the strings and calculate the fitness function.

Follow these steps to find maximum value of objective function until no stopping criteria is met.

1. Selection

Sr. no.	Initial pop	X-value	Fitness value $F(x)=x^2+3x+5$	Prob-ability (i)	%Prob-ability	Expe-cted count	real count
1	01101	12	185	0.1362	13.62	0.5449	1
2	11001	25	705	0.5191	51.91	2.0765	2
3	00101	5	45	0.0331	03.31	0.1325	0
4	10011	19	423	0.3114	31.14	1.2459	1

Sum	1358	0.9998	99.98	3.9998	4	\
Average	339.5	0.2499	24.99	0.9998	1	
Maximum	705	0.5191	51.91	2.0765	2	

2. Crossover

String no.	Crossover point	Crossover point	Child after crossover	X value	Fitness value x2+3x+5
1	01100	3	01101	13	213
2	11001	3	11000	24	653
3	11001	2	11011	27	815
4	10011	2	10001	17	345

Sum	2026
Average	506.5
Maximum	815

3. Mutation

Serial no.	Child after crossover	Flipping chromosomes	Child after flipping	X value	Fitness value f(x):x2+3x+5
1	01101	10000	11101	29	933
2	11000	00000	11000	24	653
3	11011	00000	11011	27	815
4	10001	00100	10101	21	509

Sum	2910
Average	727.5
Maximum	933

After 1st generation

Average 339. $\longrightarrow$ 727.5

Maximum 705 $\longrightarrow$ 933

2nd Generation

1. Selection

Sr No..	New pop	X value	Fitness value	Prob-ability i	% prob-ability	Expected count	Real count
1	11101	29	933	0.3206	32.06	1.2824	2
2	11000	24	653	0.2243	22.43	0.8975	1
3	11011	27	815	0.2800	28.00	1.1202	1
4	10101	21	509	0.1749	17.49	0.6996	0

Sum	2910	0.9998	99.98	3.997	4
Average	727.5	0.2499	24.99	0.9999	1
Maximum	933	0.3206	32.06	1.2824	2

2. Crossover

String no.	Mating pool	Crossover point	After crossover	X value	Fitness value f(x):x2+3x+5
1	11101	4	11101	29	933
2	11101	4	11101	29	933
3	11000	3	11011	27	815
4	11011	3	11000	24	653

Sum	3334
Average	833.5
Maximum	933

3. Mutation

Serial no.	Child after crossover	Flipping chromosomes	After flipping	X value	Fitness value f(x):x2+3x+5
1	11101	00011	11111	31	1059
2	11101	00000	11101	29	933
3	11011	00000	11011	27	815
4	11000	00010	11010	26	759

Sum	3566
Average	891.5
Maximum	1059

After 2nd Generation:

Average 727.5 $\longrightarrow$ 891.5

Maximum 933 $\longrightarrow$ 1059

Generation can be more till no stopping criteria is met. But from above tables, it can be analysed that GA is the best optimization technique to provide high performance to achieve better performance. In these tables, it can be observed how maximal and average values is going to be improved. The population average fitness value improved from 339.5 to 727.5. in one generation. And in second generation it improved from 727.5 to 891.5 and so on. Therefore, random process proves, it can make the best solution.

Coding the fitness function in MATLAB (m-file):

Create m-file named quadratic.m.

```
%function to maximize a quadratic equation

function  z=quadratic(x)

 z = (x*x+3*x+5);
```

Creation of gatool:

Type gatool on command prompt, the gatool box will be appear. In tool, type @qudratic for fitness function and mention numbers of variables defined in function. Select **best fitness** with specifying plot interval and specify other parameter regarding function.

We can also minimize the objective function. Here, minimization is shown with the help of plotting a graph in MATLAB.

Simple Multi objective Optimization Problem

Genetic Algorithm multi objective formulation (gamultiobj) can be used to sort out the problem in several variables. Here consider a example to minimize two objectives, each having one decision variable.

min F(x) = [objective1(x); objective2(x)]

 x

where, objective1(x) = (x+2)^2 - 20,

and objective2(x) = (x-2)^2 + 30

```
% Plot two objective functions on the same axis

x = -20:0.10:20;

f1 = (x+2).^2 - 20;

f2 = (x-2).^2 + 30;

plot (x, f1);

hold  on;

plot (x, f2, 'r');

grid on;

title ('Plot of objectives "(x+2)^2 - 20" and "(x-2)^2 + 30"');
```

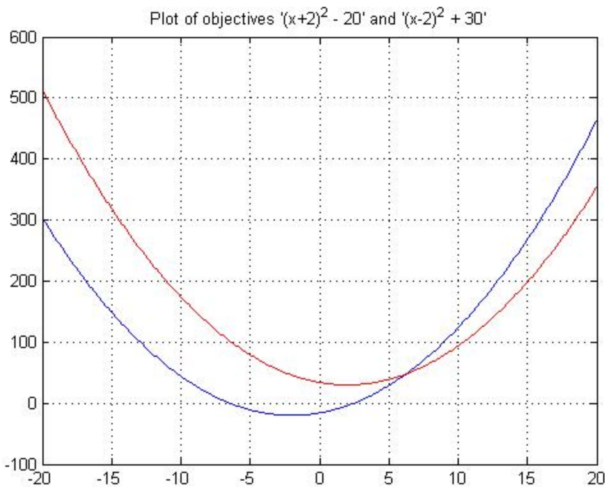

Figure shows their minima at x = -2 and x = +2 respectively. However, in a multi objective problem, x = -2, x = 2, and solution lie in the range -2 <= x <= 2 is equally optimal position. Here in multi objective problem have no single solution The main focus of the multi objective genetic algorithm, is to find a set of solutions in given range, maintaining good search space. The set of solutions is also known as a Pareto front. All solutions on the Pareto front are optimal.

Coding the Fitness Function in MATLAB

We create a MATLAB-file named simple_multiobjective.m:

```
function y = simple_multiobjective(x)

y(1) = (x+2)^2 - 20;

y(2) = (x-2)^2 + 30;
```

In MATLAB, file save with the same name of function name with extension .m. In Genetic Algorithm process, assumes the fitness function will take one input x, where x is a row vector with as many elements corresponding to the number of variables in the problem. The fitness function calculate the value of each objective function and returns these values in a single vector output y.

5. CONCLUSION

This paper extended the previous work that is also related to maximization or minimization of objective function, which is calculated by other optimization techniques. Here, Genetic Algorithm is adopt to calculate objective function. There are so many reasons or features of GA to adopt. With the help of Genetic Algorithm, generate test cases in software testing is very easily and calculate maximization or minimization of function. Genetic algorithm is used for selection of those string that has highest fitness value, with the intent of find maximum value for objective function. In MATLAB gatool provide so many remarkable features to solve problem pertaining to any area.

6. REFERENCES

[1] Bo Zhang, Chen Wang, "Automatic Generation of Test Data for Path Testing by Adaptive Genetic Simulated Annealing Algorithm", IEEE, 2011, pp. 38-42.

[2] M. A. Ahmed, I. Hermadi, "Genetic Algorithm based multiple paths test data generator", Computer and operations Research (2007).

[3] Parveen Ranjan Srivastava, Tai-hoon Kim, "Application of Genetic Algorithm in Software Testing", International Journal of Software engineering and its Application, Vol. 3, No.4, October 2009, pp. 87-95.

[4] SnageetaSabharwal, Ritu Sibal, Chanyanika Sharma, "Prioritization of test cases scenarios derived from activity diagram using genetic algorithm", ICCCT, IEEE, 2010, pp. 481-485.

[5] Sangeeta Sabharwal et al., "Applying Genetic algorithm for Prioritization of test cases Scenario derived from UML diagrams", International journal of computer science, Vol.8, Issue 3, No.2, May 2011.

[6] Sultan H. Alijahdali et al, "The Limitation of Genetic Algorithm in Software Testing", pp. 1-8.

[7] V. Mary Sumalatha, G.S.V.P. Raju, "Object Oriented Test Cases Generation Techniques using Genetic Algorithms", International Journal of Computer Applications Vol. 61- No.20, January 2013, pp 20-26.

[8] Xanthakis S, Ellis C, Skourlas C, Le Gall A, "Application of Genetic algorithms to Software Testing". In 5th International Conference on Software Engineering and its Applications pp. 625-636.

[9] Holland, J.H., Adaptation in Natural and Artificial Systems, University of Michigan Press, Ann Arbor, 1975.

[10] Jones, D.F., Mirrazavi, S.K., and Tamiz, M., Multi objective meta-heuristics: an overview of the current state-of-the-art, *European Journal of Operational Research* 137(1) (2002) 1-9.