Computer Engineering and Intelligent Systems
ISSN 2222-1719 (Paper) ISSN 2222-2863 (Online)
Vol.8, No.2, 2017

Optimization of Constrained Function Using Genetic Algorithm

Afaq Alam Khan[1*] Roohie Naaz Mir[2]
1. Department of Information Technology, Central University of Kashmir
2. Department of Computer Science Engineering, National Institute of Technology, Srinagar

Abstract

Optimization is the process of finding the minimum or maximum value that a particular function attains which also means finding the value for the independent variables of a function for *which* function is minimum or maximum. As real world problems are different and can be represented by different types of functions, so are the optimization algorithms. From last few decades, lot of research has been invested in developing different techniques of optimization suitable for different types of functions. These methods are broadly classified in to calculus based and search based methods. After brief description on optimization and classification of different optimization problems, this study focuses on constrained optimization problem and the use of Genetic Algorithm to optimize such problems.

Keywords: Optimization, Genetic Algorithm, Penalty function

1. Introduction

Optimization deals with Maximizing or minimizing a certain goal. Optimization is central to any problem involving decision making, The task of decision making entails choosing between various alternatives. This choice is governed by our desire to make the "best" decision. The measure of goodness of the alternatives is described by an objective function or performance index. Optimization theory and methods deal with selecting the best alternative in the sense of the given objective function (Edwin 2001). Because of the availability and affordability of high speed computers optimization algorithms are becoming increasingly popular (Deb 2014). For example, optimization algorithms are used in aerospace design activities to minimize the overall weight. Manufacturing of mechanical components for the purpose of achieving either a minimum manufacturing cost or a maximum component life. Production engineers are interested in designing optimum schedules of various machining operations to minimize the idle time of machines and overall job completion time. Civil engineers are involved in designing bridges with minimum cost and maximum safety (Deb 2014).

Next section discusses the general procedure for optimal design formulation process. Classification of optimization problems and methods involved is followed by the general description on GA.

2. Optimal Design Procedure

Figure 1 shows an outline of the steps usually involved in an optimal design formulation process. The formulation of an optimization problem begins with identifying the underlying problem variables. The next task is to identify the constraints associated with the optimization problem. The constraints represent some functional relationships among variables and other design parameters satisfying certain physical phenomenon and certain resource limitations. There are usually two types of constraints that emerge from most considerations. Either the constraints are of an inequality type or of an equality type. Equality constraints are usually more difficult to handle and therefore, need to be avoided wherever possible. Fortunately equality constraint can be replaced by two inequality constraints. As an example equality constraint $g(x) = 6$ can be replaced by two inequality constraints $g(x) >= 5$ and $g(x) <=7$. Wherever possible, a small tolerance $\epsilon=.001$ (say) can also be added to the rigid constraints to have more flexible search space, as an example $g(x)=6$ can be translated in to following two constraints

$g(x) -(6+ (\epsilon \times 6)) < 0$ and $-g(x) + (6- (\epsilon \times 6)) <0$

The third task in the formulation procedure is to find the objective function in terms of problem variables and other problem parameters. The common objectives involve minimization or maximization of a certain function. In real world optimization problem, there could be more than one objective that needs to be optimized simultaneously. Even though a multi-objective optimization algorithms exist in literature (Konak 2016)(Chakraborti 2008)(Khalil et al 2012), they are complex and computationally expensive. Thus in most optimal design problem, multi objectives are avoided. Instead the most important objective is chosen as objective function of optimization problem, and the other objectives are included as constraints. As mentioned before the objective function can be of two types. Either the objective function is to be maximized or it has to be minimized. If the algorithm is developed for solving a minimization problem, it can also be used to solve a maximization problem by simply multiplying the objective function by -1 and vice versa. This is called duality principle (Deb 2014). The final task of the formulation procedure is to set the minimum and the maximum bounds on each variables. This information is required to confine the search algorithm within the feasible region. After the above four tasks are completed, the optimization problem can be mathematically written in a special format, as

shown below:
Minimize f(x)

subject to:
$g_j(x)>=0$ j=1,2,.....j;
$h_k(x)=0$ k=1,2,....k;
$x_i^{(L)} <= x_i <= x_i^{(u)}$ i=1,2,.....N.

Where f(x) is the objective function, $g_j(x)$ and $h_k(x)$ are inequality and equality constraints respectively and $x_i^{(L)}$ and $x_i^{(u)}$ are the lower and upper bounds respectively for different variables of the problem.

After the optimization problem is formulated, an optimization algorithm is chosen and an optimal solution of the problem is obtained.

3. Classification of Optimization Algorithms

Optimization Literature contains a large number of algorithms, each suitable to solve a particular type of problem. Optimization Algorithms are classified into a number of groups; single variable v/s multi variable optimization, unconstrained v/s constrained optimization, multi objective optimization, Multi modal optimization, nested optimization, stocastic/ dynamic optimization etc. Accordingly several methods have been proposed depending upon the type of optimization. These methods can be broadly classified as calculus based and search based methods. While Langrange Multiplier and KKT Conditions are calculus based methods, Search based methods include Linear programming, Genetic Algorithm, Simulated Annealing etc.

4. Genetic Algorithm

Genetic Algorithms are search algorithms based on the mechanism of natural selection and natural genetics (Azmathulla et al 2013). The basic objective of natural genetics is the retention of fit genes and discarding of poor ones. In nature, weak and unfit species within their environment are faced with extinction by natural selection. The strong ones have greater opportunity to pass their genes to future generation. In the long run, species carrying the correct combination in their genes become dominant in their population. Sometimes, during the slow process of evolution, random changes may occur in genes. If these changes provide additional advantage in the challenge for survival, new species evolve from the old ones. Unsuccessful changes are eliminated by natural selection (Konak 2016).

In GA terminology, a solution vector X is called an *individual* or a chromosome. Chromosomes are made of discrete units called genes. Each gene controls one or more features of the chromosome. GA operate with a collection of chromosomes, called *population* which is generated randomly. As the search proceeds, GA's Operator called *selection* or sometimes *reproduction* returns the population which includes only fitter chromosomes generation after generation and eventually converges.

GA's two other operators to generate new solutions from existing ones are *crossover* and *mutation.* In crossover, generally two chromosomes called parents are combined together to form new chromosomes, called offspring. The parents are selected among existing chromosomes in the population with preference towards fitness so that offspring is expected to inherit good genes which make the parent fitter(Konak 2016). There are different crossover operators which are selected on the way chromosomes are encoded. Single-point, two-point, Multipoint, uniform, arithmetic, ordered crossover are some of the examples which have been reported in the literature(Datta et al 2012). By iteratively applying the crossover operator, genes of good chromosomes are expected to appear more frequently in the population, eventually leading to convergence to an overall good solution (Konak 2016).

The mutation operator introduces random changes into characteristics of chromosomes. Mutation is generally applied at the gene level. In typical GA implementations, the mutation rate (probability of changing the properties of gene) is very small and depends on the length of the chromosome. therefore the new chromosome produced by mutation will not be very different from the original one. Mutation plays a critical role in GA. Crossover leads the population to converge by making the population alike. Mutation reintroduces genetic diversity back into the population and assists the search escape from local optima (Konak 2016). There are many different forms of mutation for different kinds of representation. Flipping, Interchanging, Reversing Gaussian, Boundary, uniform and non-uniform are some of the mutation operators which have been used by researchers (Datta et al 2012). *Selection* of chromosomes for the next generation is based on the fitness of an individual. There are different *selection* procedures in GA depending on how the fitness values are used. The most commonly used methods of selecting chromosomes to crossover are Roulette wheel selection, Boltzmann selection, Proportional selection, ranking and tournament selection (Datta et al 2012) . upon application of said operators on the population, there is a chance that best chromosomes may be lost when new population is created by crossover and mutation. In such cases researchers have suggested the use of *Elitism*; which recommends the best chromosomes be copied to new population (Chande et al 2009)

The procedure of GA is as follows:

Step 1. Set t=1. Randomly generate N solutions to form the first population, P_t. Evaluate the fitness of solutions in P_t.
Step 2. Crossover: Generate an offspring population Q_t as follows:
2.1. Choose two solutions x and y from P_t based on the fitness values.
2.2. Using a crossover operator, generate offspring and add them to Q_t.
Step 3. Mutation: Mutate each solution x ϵ Q_t with a predefined mutation rate.
Step 4. Fitness assignment: Evaluate and assign a fitness value to each solution x ϵ Q_t based on its objective function value and infeasibility.
Step 5. Selection: Select N solutions from Q_t based on their fitness and copy them to P_{t+1}
Step 6. If the stopping criterion is satisfied, terminate the search and return to the current population, else, set t= t+1 go to step 2. (Konak 2016) (Goldberg 1989)

5. Constrained Optimization in Literature

Most of the real world problems are represented by constrained functions. Genetic Algorithms are most directly suited to unconstrained optimization. Application of Genetic algorithm to constrained optimization problem is often challenging. Michalewicz and Schoenauer (Michalewicz et al 1996) have discussed different constraint handling methods used in GA. They have classified most of the evolutionary constraint handling methods into five categories:
1. Methods based on preserving feasibility of solutions
2. Methods based on penalty functions
3. Methods making distinction between feasible and infeasible solutions
4. Hybrid methods

As reported by (Deb, 2014) and (Michalewicz et al 1996), the most popular approach to handle constraints in GA is the methods based on penalty functions. Penalty functions penalize infeasible solutions by reducing their fitness values in proportion to their degrees of constraint violation (Deb 2014)(Michalewicz et al 1996).

6. Use of Penalty function

Most popular approach in Genetic Algorithm to handle constraints is to use Penalty functions**.** Penalty method transforms constrained problem to unconstrained one. In classical optimization, two types of penalty functions are commonly used: interior and exterior penalty functions. In GAs exterior penalty functions are used more then interior penalty functions.

The general formulation of an **exterior penalty** function is:

$Ø(x) = f(x) + [\sum \mu_k (h_k(x))^2 + \sum \mu_j \text{Max}\{0, g_j(x)\}^{2}]$
(For k= 1,2...p and j= 1,2,... m)

If the inequality holds $g_j(x) <=0$ and $\max[0, g_j(x)]$ will be zero. Therefore the constraint does not effect Ø (x). If the constraint is violated that means $g_j(x) >0$ or $h_k!=0$, a big term will be added to Ø (x) function such that the solution is pushed back towards the feasible region

example

Minimize $f(x)= 4 [1/3 (x_1 + 1)^3 +x_2]$
subject to $g_1(x) = 2-2x_1 <=0$
$g_2(x) = -2x_2<=0$

Sol:
Minimize $Ø(x_1, x_2, \mu) = f(x) + \mu \sum \max\{g_j(x),0\}^2$
$Ø(x_1, x_2, \mu)= 4 [1/3 (x_1 + 1)^3 +x_2] + \mu \max \{g_1(x),0\}^2 + \mu [\max \{g_2(x),0\}^2$
$Ø(x_1, x_2, \mu)= 4 [1/3 (x_1 + 1)^3 +x_2] + \mu (2-2x_1)^2 + \mu (-2x_2)^2$

A survey of constraint handling techniques used with evolutionary algorithms can be found in (Carlos 1999)(Oliver Kramer, 2010) , Under Penalty based methods (Michalewicz and Schoenauer 1996) have reported different approaches which have been used in optimization problems viz, static, Dynamic Adaptive, Annealing, and Death penalty.

Genetic Algorithm's population based approach and ability to make pair wise comparison in tournament selection operator was exploited by (Deb 2000) to devise a penalty function approach that does not require penalty parameter, rather careful comparison among feasible and infeasible solutions are made so as to provide a search direction towards feasible region. Once sufficient feasible solution are found, a niching method (along with controlled mutation operator) is used to maintain diversity among feasible solution. Genetic Algorithm with this constraint handling was tested on nine problems commonly used in literature.

Optimization of constrained function f(x) can also be treated as multi objective optimization problem in which we will have m+1 objectives, where m is the total number of constraints then any multi objective algorithm can be adopted.

A real-time model for a reservoir system meant for irrigation was developed by (Azmatullah 2008) to

optimize the reservoir release over different time periods. Both Genetic Algorithm and Linear Programming were applied and their performance analysed. From the results , the GA model was found to be superior to LP model.
While applying simplex method and Genetic Algorithm to optimize certain linear programming problem, (S. Datta, 2012) found the results returned by GA to be more optimum`as compared to LP.

7. Validation of Results

Genetic Algorithm through the theory of schemata ensures that the set of results returned will be the most optimum. Selection operator after crossover and mutation will automatically discard poor solutions. Short, Low order and above average schemata receive exponentially increasing trials in subsequent generations (Goldberg 1989).

Alternatively an exhaustive search algorithm can be used to record set of optimum results and the results obtained through GA can be compared with them to check the validity of results.

Problems where all the constraints are linear is a suitable candidate for **Linear Programming.** The results thus obtained can be compared with that of GA.

8. Tools for Genetic Algorithm

Global Optimization Toolbox in Matlab (Genetic Algorithm Solver)

http://in.mathworks.com/products/global-optimization/
http://www.mathworks.com/products/global-optimization/index.html

GEATbx: Genetic and Evolutionary Algorithm Toolbox for use with MATLAB

http://www.geatbx.com/docu/index.html

GA: A Package for Genetic Algorithm in R

https://www.jstatsoft.org/article/view/v053i04

An Educational Genetic Algorithms Learning Tool

http://www.ewh.ieee.org/soc/es/May2001/14/Begin.htm

GPDotNet

http://www.codeproject.com/Articles/568620/FunctionplusoptimizationpluswithplusGeneticplusAlg
http://bhrnjica.net/category/gpdotnet/
https://gpdotnet.codeplex.com/

JGAP: Java Genetic Algorithms Package

http://jgap.sourceforge.net/

ECJ: A Java-based Evolutionary Computation Research System

https://cs.gmu.edu/~eclab/projects/ecj/

9. Conclusion

Genetic Algorithms are of major significance to the development of new generation of IT applications. The potential which they offer over existing techniques is enormous. They find application in versatile fields (Chande et al 2009). Where ever the optimization is required, Genetic algorithm finds its use. Among various methods, Penalty function method is most widely used method to deal with constrained optimization using Genetic Algorithm.

References

Deb K. (2014), “Optimization for Engineering Design: Algortithms and Examples”, 2nd Ed. , PHI.

Deb K. (2000), “An efficient Constraint Handling Method for Genetic Algorithms”, Computer Methods in applied mechanics and Engineering, 186.

Ozgur Yeniay (2005), “Penalty Function Methods for Constrained Optimization with Genetic Algorithms”, Mathematical and Computational Applications, 10 (1).

Carlos A. (2001), “An Updated Survey of GA-Based Multiobjective Optimization Techniques ”, ACM.

Pandharipande S. L., Deshmukh A. R., Kalnake R. P. (2014), “Genetic Algorithm for Constrained Optimization with stepwise Approach in Search Interval Selection of Variables”, International Journal of Computer Applications, 87(11).

Oliver Karmer (2010), “A Review of Constraint- Handling Techniques for Evolution Strategies”, Applied Computational Intelligence and Soft Computing.

Datta S., Garai C., Das C. (2012), “Efficient Genetic Algorithm on Linear Programming Problem For Fittest Chromosomes”, Journal of Global Research in Computer Science, 3(6), 2012.

Konak A. (2006), “MultiObjective Optimization using Genetic Algorithm: A tutorial”, Reliability Engineering & Safety System, 2006

Goldberg D.E. (1989) Genetic Algorithms in Search, Optimization and Machine Learning (Addison

Wesley, Reading, MA).

Chande S. V., Sinha M.(2009), "Genetic Algorithm: A versatile Optimization Tool", BVICAM's International of Information Technology, 1(1).

Luca Scrucca (2013), GA: A package for Genetic Algorithms in R, Journal of Statistical Software, 53 (4).

Azamathulla H. Md., Fu-Chun Wu, Aminuddin Ab Ghani (2008), "Comparison between gentic and linear programming approach for real time operation", Journal of Hydro-Environment Research, 2.

Datta S., Chayanika Garai, Chandrani Das(2012) , "Efficient Genetic Algorithm on Linear Programming Problem for fittest Chromosomes", Journal of Global Research in Computer Science., 3(6).

Edwin K. P. Chong, Stanislaw H. Kak (2001) , "An Introduction to optimization", 2nd Ed. WIELY.

Chakraborti N, B. Siva Kumar, V. Satish Babu, S. Moitra, A. Mukhopodyay (2008), "A New Multiobjective Algorithm applied to hot rolling process", Applied Mathematical Modelling, 32.

Rahim Khalil, D. Stockton, P.S. Kang, L. M. Mukhongo (2012), "A Multiobjective optimization Approach using Genetic algorithms for quick response to effects of variability in Flow Manufacturing", International Journal of Advanced Computer Science and applications, 3(9).

Michalewicz, Z. and Schouenauer, M. (1996), "Evolutionary Algorithms for constrained parameter optimization problems", Evolutionary Computation, 4, 1-32.

Carlos A. Coello Coello (1999), "A Comprehensive Survey of Evolutionary-Based Multiobjective Optimization Techniques", Knowledge and Information Systems, 269-308.

http://www.genetic-programming.com

http://geneticalgorithms.ai-depot.com/

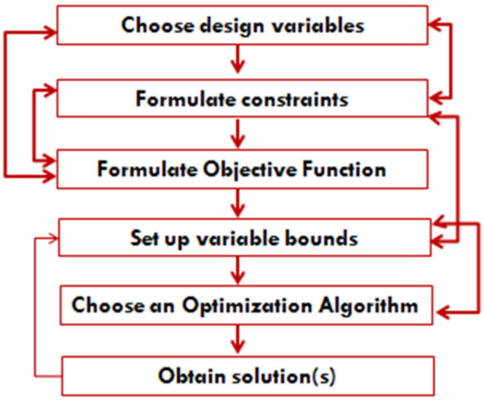

Figure 1. A flowchart of the optimal design procedure (Deb 2014)