Optimization Techniques
Genetic Algorithm

Satishkumar L. Varma

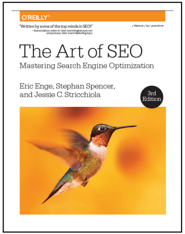
Pillai College of Engineering (PCE), New Panvel

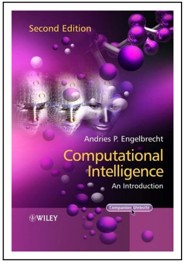
2022-01-14

Table of Contents

References

1. David E. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning,” Addison-Wesley, Inc.
2. Eric Enge, Stephan Spencer, and Jessie C. Stricchiola, “The Art of SEO: Mastering Search Engine Optimization,” 3ed, Oreilly, 2015.
3. Mykel Kochenderfer and Tim Wheeler, “Algorithms for Optimization,” MIT Press, 2019.
4. Andries Engelbrecht, “Computational Intelligence: An Introduction,” 2ed, Wiely, 2007.
5. Author(s):Edwin K. P. Chong, Stanislaw H. Żak, “An Introduction to Optimization, 3ed,” John Wiley, 2008.
6. S. Rajasekaran, G. A. Vijayalakshmi, “Neural Networks, Fuzzy Logic and Genetic Algorithms: Synthesis & Applications,” PHI, 2004.





Learning Objectives

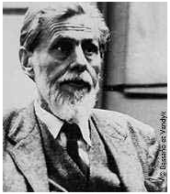
During this presentation, an instructor/mentor aims to

1. List Types of Optimization Problem
2. Explain how GA works by solving a very simple optimization problem
3. Estimate the optimal values and objective function
4. Demonstrate optimization process using R

History

SC(Zadeh, 1981) = [McCulloch(NN, 1943) + Zadeh(FL, 1965) + Rechenberg(EC, 1960)]

EC(Rechenberg, 60) = Fogel(EP, 62)+Rechenberg(ES, 65)+Holland(GA, 70)+Koza(GP, 92)






Classification

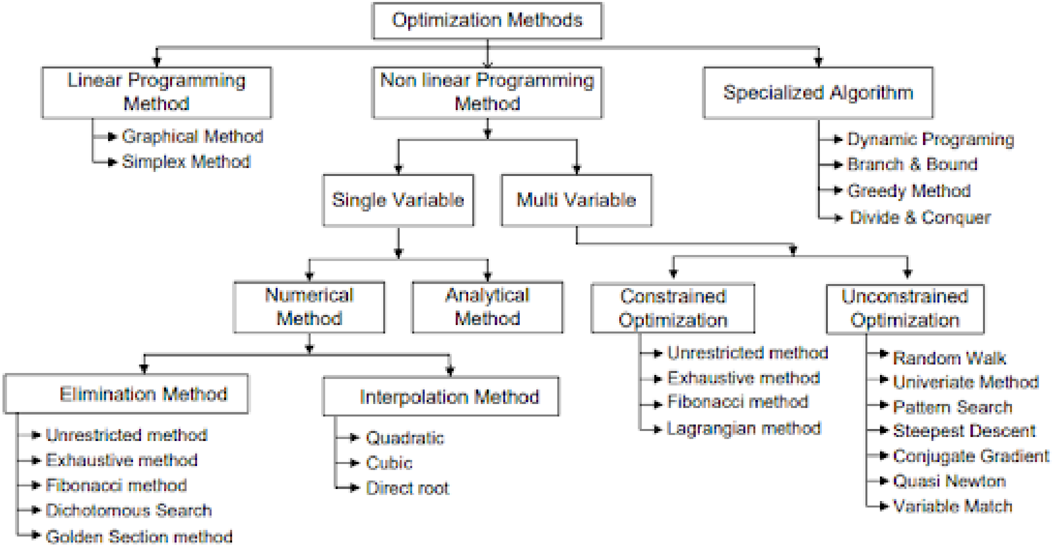
Lotfi A. Zadeh, "What is Soft Computing", Soft Computing. Springer-Verlag, Germany/USA 1997 He defined SC into one multidisciplinary system as the fusion of

- Neural Network: for learning and adaptation
- Fuzzy Logic: for knowledge representation via fuzzy If – Then rules.
- Genetic Computing: for evolutionary computation

Learning

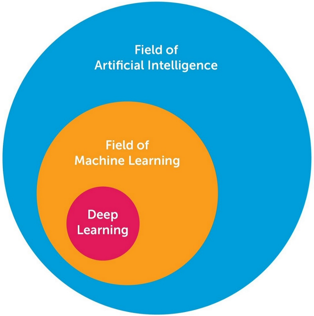
Soft Computing

- Fuzzy Computing
  1. Fuzzy Login
  2. Fuzzy Set
  3. Fuzzy Theory
- Neural Network
  1. Artifical NN
  2. Backpropogation NN
  3. Feed Forward NN
- Evolutionary Computing
  1. Genetic Algorithm
  2. Evolutionary Algorithm
  3. Different Evolutions
- Machine Learning
  1. Semi Supervised
  2. Unsupervised
  3. Supervised
- Swarm Computing
  1. Anti Colony Optim
  2. Particle Swarm Optim
  3. Firefly Algorithm
- Additional Techniques
  1. Chaos Theory
  2. Belief Theory
  3. Probabilistic Reasoning
  4. Wisdom System


Optimization Methods
Linear Programming Method
Graphical Method
Simplex Method
Non linear Programming Method
Single Variable
Multi Variable
Numerical Method
Analytical Method
Elimination Method
Unrestricted method
Exhaustive method
Fibonacci method
Dichotomous Search
Golden Section method
Interpolation Method
Quadratic
Cubic
Direct root
Constrained Optimization
Unrestricted method
Exhaustive method
Fibonacci method
Lagrangian method
Unconstrained Optimization
Random Walk
Univeriate Method
Pattern Search
Steepest Descent
Conjugate Gradient
Quasi Newton
Variable Match
Specialized Algorithm
Dynamic Programing
Branch & Bound
Greedy Method
Divide & Conquer

Genetic Algorithm

- 1965: Introduced by John Holland, Professor, Michigan University, USA
- 1975: Publication of first article on GA
- GA is based on two fundamental principles of biological processes:
- 1865: Genetics: Gregor Johan Mendel
- 1875: Evolution: Charles Darwin



Biological Background

Genetics:

- Living bodies contain cells and each cell carries some unit of heredity called gene
- Each gene contains chromosome in the nucleus
- Different species carry fixed number of chromosomes (e.g Human 46, Frogs 26, Mosquito 6)

Genetic code:

- Spiral helix of protein substance is called DNA
- DNA code is unique for a specie and used as biometric trait
- DNA inherits some characteristics from one generation to next generation

Reproduction

Organism's cell (Cell division), Reproductive $cell = gameteX + gameteY$

- Output is haploid if reproductive cell has half the number of chromosomes
- O/p is diploid if each chromosome from both haploids are combined to have full no.

Crossing over

- Method of combining information from two different organism's body cells
- Random crossover points makes infinite diversities in information possible

Mutation

- To make the process forcefully dynamic when variations in popu. going to stable

Evolution (Natural Selection)

Information propagation

- Heredity: An offspring has many of its characteristics of its parents (i.e. information passes from parent to its offspring)

Population diversity

- Diversity: Variation in characteristics in the next generation

Survival for existence

- Selection: Only a small percentage of the offspring produced survive to adulthood

Survival of the best

- Ranking: Offspring survived depends on their inherited characteristics

Genetic Algorithm

- GAs are the heuristic search and optimization techniques that mimic the process of natural evolution.
- Principle Of Natural Selection: Select The Best, Discard The Rest
- GA is a population-based probabilistic search and optimization techniques
- GA works based on the mechanisms of natural genetics and natural evaluation
- GA is an iterative process.
- GA is a searching technique.
- Working cycle with / without convergence.
- Solution is not necessarily guaranteed. Usually, terminated with a local optima.

Concept of optimization problem

- Definition: Optimum value that is either minimum or maximum value
- Example: $y = F(x)$
- Let, $2x - 6y = 11$ or $y = (2x - 11) / 6$
- Can we determine an optimum value for y.
- It is simple to understand but not related to optimization problem in reality.

Traditional optimization and search technique

Example of Optimization Problems

- Travelling Salesman Problem
- Knapsack Problem
- Graph Coloring Problem
- Job Machine Assignment Problem
- Coin Change Problem
- Binary search tree construction problem

Optimization requirements

- Input parameters
- Objective function(s)
- Fitness evaluation / mathematical formula or algorithm
- Constraint(s)
- Encoding and Decoding

Learning Activity 1: Operators in genetic algorithm

Learning Activity 1: Operators in genetic algorithm
Exercise 1: Individual activity

- Online polls 1

Learning activity 1: Solution

Before Crossover

Parent 1 :	0	1	1	0	0	0	1	0
Parent 2 :	1	0	1	0	1	1	0	0

Crossover Point k_1 Crossover Point k_2

Select two crossover points randomly

After Crossver

Offspring 1:	0	1	1	0	1	0	1	0
Offspring 2:	1	0	1	0	0	1	0	0

Limitations of the traditional optimization approach

- Computationally expensive
- For a discontinuous objective function, methods may fail
- Method may not be suitable for parallel computing
- Discrete (integer) variables are difficult to handle
- Methods may not necessarily adaptive

Soft Computing techniques

- Address the above mentioned limitations of solving optimization problem with traditional approaches

Types of Optimization Problem

- Unconstrained optimization problem: Problem is without any functional constraint
- Constrained optimization problem: Optimization problem with at one or more functional constraint(s)
- Integer Programming problem: If all the design variables take some integer values
- Real-valued problem: If all the design variables are bound to take real values
- Mixed-integer programming problem: Some of the design variables are integers and the rest of the variables take real values
- Linear optimization problem: Both objective functions as well as all constraints are found to be some linear functions of design variables
- Non-linear optimization problem: If either the objective function or any one of the functional constraints are non-linear function of design variables

Constraints in GA

Unconstrained opt. problem: Problem is without any functional constraint

- Example: Minimize $y = f(x_1, x_2) = (x_1 - 5)^2 + (x_2 - 3)^3$ where $x_1, x_2 \geq 0$

Constrained optimization problem

- Optimization problem with at one or more functional constraint(s)
- Example: Maximize $y = f(x_1, x_2, ..., x_n)$
- Subject to some constants ci where $i = 1, 2, ..., k$ and $k > 0$ with design parameters $x_1, x_2, ..., x_n$

Integer Programming problem

- If all the design variables take some integer values
- Example: Minimize $y = f(x_1, x_2) = 2x_1 + x_2$
- Subject to $x_1 + x_2 \leq 3$ and $5x_1 + 2x_2 \leq 9$

Constraints in GA

Linear optimization problem: Both objective functions as well as all constraints are found to be some linear functions of design variables

- Example: Maximize $y = f(x_1, x_2) = 2x_1 + x_2$
- Subject to $x_1 + x_2 \leq 3$ and $5x_1 + 2x_2 \leq 10 and x_1, x_2 \geq 0$

Non-linear optimization problem: If either the objective function or any one of the functional constraints are non-linear function of design variables

- Example: Maximize $y = f(x_1, x_2) = x_1^3 2 + 5x_1^2$ Subject to $x1^4 + 3x_2^2 \leq 629$ and $2x_1^3 + 4x_2^3 \leq 133$ and $x_1, x_2 \geq 0$

A simple optimization problem

- Let us estimate the optimal values of a and b using GA which satisfy $2a^2 + b = 57$
- Any opt. problem starts with an objective function,
- so rewrite above equation as: $f(a, b) = 2a^2 + b - 57$

Optimization Process

- Optimization process using 4 steps in R for one iteration (or generation).
- Step — 1: Initialize population
- Step — 2: Selection (roulette wheel method)
- Step — 3: Crossover

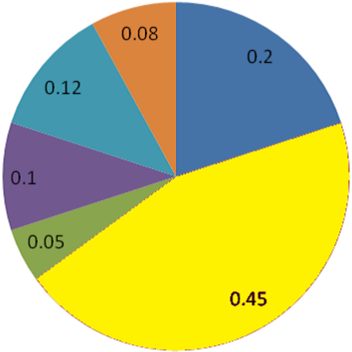
Optimization Process

Roulette wheel method

- Expression is used to calculate fitness probability (FP) of a single chromosome: $FP = Fi/Sum(Fi)$; Fi = fitness value of ith chromosome
- Roulette wheel is a pie plot where the value of each pie is expressed in terms of fitness probability.
- Note that fitness value and fitness probability are two different terms.
- In this optimization problem, chromosome which produces low fitness value has high fitness probability.
- Fitness probability of each chromosome is computed from fitness values.
- Chromosome having high fitness probability will have higher chance of getting selected.

Roulette wheel method

- FP shown for 6 diff. chromosomes as computed using $FP = Fi/Sum(Fi)$
- The sum of all fitness probabilities must be always equal to 1
- To select the fittest chromosomes, 6 random prob. generated bet. 0 and 1



Mutation

Step 4:

- Mutation is the process of altering the value of gene
- i.e to replace the value 1 with 0 and vice-versa
- How many genes need to be altered and at what positions.
- The mutation parameter decides how many genes to be mutated.
- If mutation parameter is 0.1 (usually kept low values).
- Then 0.1 times the total genes are allowed to mutate.

Mutation

Step 4:

- In this opt. problem, total number of genes is 48 (6 x 8).
- So, 4.8 i.e 5 randomly selected genes allowed for mutation.

Binary representation of chromosomes of offspring

1	1	0	1	1	1	$\boxed{0}$	1
0	$\boxed{1}$	0	1	1	1	1	0
0	1	1	0	1	1	$\boxed{1}$	1
0	1	1	$\boxed{1}$	0	0	0	1
0	1	1	0	0	1	0	1
1	0	0	0	$\boxed{0}$	0	1	1

After mutation

1	1	0	1	1	1	$\boxed{1}$	1
0	$\boxed{0}$	0	1	1	1	1	0
0	1	1	0	1	1	$\boxed{0}$	1
0	1	1	$\boxed{0}$	0	0	0	1
0	1	1	0	0	1	0	1
1	0	0	0	$\boxed{1}$	0	1	1

Summary

- List Types of Optimization Problem
- Explain how GA works by solving a very simple optimization problem
- Estimate the optimal values and objective function
- Demonstrate optimization process using R

Thank You.