Comparative Analysis of Threshold Acceptance Algorithm, Simulated Annealing Algorithm and Genetic Algorithm for Function Optimization

Dr. Tejas P Patalia[1] and Dr. G.R. Kulkarni[2]

1

Received: 16 December 2011 Accepted: 3 January 2012 Published: 15 January 2012

Abstract

The goal of this study of threshold acceptance algorithm (TA), simulated annealing algorithm (SA) and genetic algorithm (GA) is to determine strength of Genetic Algorithm over other algorithm. It gives a clear idea of how genetic algorithm works. It gives the idea of various sub methods used in genetic algorithm to improve the results and outcome. Basically genetic algorithm and all traditional heuristic methods are used for optimization. Optimization problems are class NP complete problems. Genetic algorithm can be viewed as an optimization technique which exploits random search within a defined search space to solve a problem by some intelligence ideas of nature. In this work we have done Comparative analysis of Threshold Acceptance Algorithm, Simulated Annealing Algorithm and Genetic Algorithm by considering different test functions and its constraints to minimize the test functions.

Index terms— Heuristic methods, Genetic Algorithm, Chromosomes, Mutation, threshold acceptance algorithm, simulated annealing algorithm, function optimization.

1 INTRODUCTION

ptimization is the process of finding absolutely best values of the variables so that value of an objective function becomes optimal. Optimization problems are a class of NP-Complete problems. This work contains overview of threshold acceptance algorithm, simulated annealing algorithm and brief introduction to Genetic algorithm. Genetic algorithm is probabilistic, heuristic, robust search algorithm premised on the evolutionary ideas of natural selection and genetic. Main idea behind the design of genetic algorithm is to achieve robustness and adaptiveness in real world complex problems. Genetic algorithm can be viewed as an Optimization technique, which exploits random search within a defined search space to solve a problem, by some intelligence ideas of nature.

2 II. THRESHOLD ACCEPTANCE ALGORITHM

Threshold Accepting (TA) is a local search method and was first described by Dueck and Scheuer and Moscato and Fontanari.

A classical local search starts with a random feasible solution and then explores its neighbourhood in the solution space by moving (usually randomly) from its current position, accepting a new solution if and only if it improves the objective function. TA overcomes the problem of stopping in local minima by also allowing uphill-moves that is TA also accepts new solutions which lead to higher objective function values.

To implement TA, three points need to be specified:

1. The objective function f: This function is generally given by the problem at hand.

3 The neighborhood definition (the function N):

Given a candidate solution x c , one needs to define how to move from this solution to an alternative, but 'close' solution x n . 3. The thresholds: Given a neigbourhood definition, one needs to determine the magnitude of the deterioration in the objective function that the algorithm should still accept for a new solution.

The pseudo-code of TA can be given as follows:

1: Initialize ?Rounds and ?Steps 2: Compute threshold sequence Tr 3: Randomly generate current solution x c ? X 4: for r = 1: ?Rounds do 5: for i = 1: ?Steps do 6: Generate xn ? N (xc) and compute ? = f (x n) -f (x c) 7: if ? < Tr then x c = x n 8: end for 9: end for 10:x sol = x c

Here, f is the objective function to be minimized. x c denotes the current solution, x n is the 'new' (or neighbor) solution, and X is the set of feasible solutions.

TA starts with a (random) feasible solution. Given a threshold sequence T of length ?Rounds, one can see that TA always accepts a solution that improves the objective function f, but deteriorations are only accepted if they are not worse than a particular threshold, Tr. Over time, the threshold decreases to zero, thus TA turns into a classical local search.

4 III. SIMULATED ANNEALING ALGORITHM

Simulated Annealing (SA) was introduced by Kirkpatrick. Like other trajectory methods, it evolves a gradually, the algorithm follows some path ('trajectory') through the search space.

SA starts with a random solution x c and creates a new solution x n by adding a small perturbation to x c . If the new solution is better (? < 0), it is accepted. In case it is worse, though, SA applies a stochastic acceptance criterion, thus there is still a chance that the new solution is accepted, albeit only with a certain probability. This probability is a decreasing function of both the order of magnitude of the deterioration and the time the algorithm has already run. The latter feature is controlled by the temperature parameter T which is reduced over time; hence impairments in the objective function become less likely to be accepted and eventually SA turns into classical local search. Here, the algorithm stops after a predefined number of iterations Rmax; of course, alternative stopping criteria are possible.

The pseudo-code of SA can be given as follows:

1: Generate initial solution x c , initialize Rmax and T 2: for r = 1 to Rmax do 3: while stopping criteria not met do 4: Compute x n ? N (x c) (neighbour to current solution) 5: Compute ?= f (x n) -f (x c) and generate u (uniform random variable) 6: if (? < 0) or (e -?/T > u) then x c = x n 7: end while 8: Reduce T 9: end for IV.

5 GENETIC ALGORITHM

What is Genetic Algorithm? Genetic algorithms are probabilistic, robust and heuristic search algorithms premised on the evolutionary ideas of natural selection and genetic.

Darwin's Principle of Natural Selection IF there are organisms that reproduce, and IF offspring's inherit traits from their progenitors, and IF there is variability of traits, and IF the environment cannot support all members of a growing population, THEN those members of the population with less adaptive traits will die out, and THEN those members with more-adaptive traits will thrive.

6 Concept

The basic concept of genetic algorithms is designed to simulate the processes in natural system necessary in for evolution, specifically for those that follow the principle of survival of the fittest. They

7 Permutation Encoding

Useful in ordering problems such as the Traveling Salesman problem (TSP). In TSP every chromosome is a string of numbers, each of which represents a city to be visited.

8 Value Encoding

Used in problems where complicated values, such as real numbers, are used and where binary encoding would not suffice. Good for some problems, but often necessary to develop some specific crossover and mutation techniques for these chromosomes.

9 Tree Encoding

Tree encoding is used mainly for evolving programs or expressions. In the tree encoding every chromosome is a tree of some objects, such as functions or commands in programming language. Tree encoding is useful for evolving programs or any other structures that can be encoded in trees. Programming language LISP is often used for this purpose, since programs in LISP are represented directly in the form of tree and can be easily parsed as a tree, so the crossover and mutation can be relatively easily.

10 2012

March because they work collectively and their inter-relations are complex. The encoding process of solution as a chromosome is most difficult aspect of solving any problem using genetic algorithm. Encoding of solution as a chromosome is known as genotype and its equivalent physical representation is known as phenotype.

ii. Fitness Function

In the nature the organism's "fitness" can be measured by its ability to reproduce, to adapt and to survive. In genetic algorithm chromosomes should be measured by some technique to decide which chromosomes are good compared to other chromosomes. Fitness function is a objective or evaluation function which is used to measure how good a chromosome is. Fitness function assigns fitness value to each chromosome using genetic structure and relevant information of the chromosome. Fitness function plays a big role because subsequent genetic operators use fitness values to select chromosomes. Different fitness functions are used depending on type and solution vector of problem. For function optimization problems, fitness function may be the value of objective function.

11 iii. Reproduction

Reproduction or selection is based on the concept natural selection and it is one of the main three operators used in genetic algorithm. The main objective of reproduction operator is to emphasize good chromosomes in a population. Reproduction makes multiple copies of relatively good chromosomes at the cost of relatively bad chromosomes while keeping population size constant. The essential idea is that chromosomes having a higher fitness value have a higher probability of selection. The identification of good or bad chromosomes is done using fitness value of the chromosomes. Many selection methods are available, some of them make multiple copies of the chromosomes on the basis of probability, where as some make multiple copies deterministically.

12 1) Roulette Wheel Selection Method

In this method each chromosome in the population occupies an area of the roulette wheel proportional to its fitness value. Chromosomes with better fitness occupies large fraction of roulette wheel where as chromosomes with bad fitness occupies small fraction of roulette wheel. Then roulette wheel is spun as many times as population size. Each time roulette wheel pointer points one chromosome and that chromosome is placed in mating pool. A chromosome with a higher fitness is likely to receive more copies than a chromosome with a lower fitness. Roulette Wheel Selection method is widely used for the maximization problems but it has two main drawbacks: i.

It can handle only maximization problem so minimization problem must be converted into an equivalent maximization problem. ii.

If a population contains a chromosome having exceptionally better fitness compared to the rest of the chromosomes in the population then this chromosome occupies most of the roulette wheel area. Thus, almost all the spinning of the roulette wheel is likely to choose the same chromosome, this may result in the loss of genes diversity and population may coverage to local optima.

2) Rank Based Selection Method Rank based selection uses fitness value of the chromosomes to sort chromosomes in to ascending or descending order depending on the minimization or maximization problem. Then it assigns reproduction probability and ranked fitness to each chromosome on the basis of only rank order of the chromosome in the current population. Rank based selection also assigns some probability to the worst chromosome so that it has some chance for getting selected.

13 3) Steady State Selection

In the steady state selection in every generation a few good chromosomes are selected for creating new offspring. Then some bad chromosomes are removed and the new offspring is placed. The rest of population survives to new generation.

14 4) Elitism

After crossover and mutation new population is generated. With the help of elitism we can store the best found chromosomes. The remaining chromosomes are delivered for the next generation. In this way we cannot lose best chromosomes.

15 5) Tournament Selection

Runs a "tournament" among a few individuals chosen at random from the population and selects the winner (the one with the best fitness) for crossover. Two entities are picked out of the pool, their fitness is compared, and the better is permitted to reproduce. Advantage is decreases computing time.

Comparative Analysis of Threshold Acceptance Algorithm, Simulated Annealing Algorithm and Genetic Algorithm for Function Optimization

16 Global Journal of Researches in Engineering

Volume XII Issue v v v v I Version I

17 ( )

cannot identify the role of each gene individually organisms of the same species as well as different species. Even twins have at least some minor differences. These differences are due to genetic structure, which is called chromosomes. Chromosomes or individuals are consisting of genes. Genes may contain different possible values depending on the environment, constraints and struggle to survive. Each gene is responsible for some part of solution but we b) Operators of Genetic Algorithm i. Crossover

Two parents chromosomes are selected randomly from the mating pool, few genes of the chromosomes are exchanged between these two parents and offspring are produced. In general crossover operator recombines two chromosomes so it is also known as recombination. Crossover is intelligent search operator that exploits the information acquired by search space. Generally crossover probability is very high like 1.00, 0.95, 0.90 etc.. Different types of Crossover methods can be used i.e. 1-point crossover, n-point crossover and uniform crossover.

18 ii. Mutation

Mutation is secondary operator used in genetic algorithm to explore new points in the search space. In the latter stages of a run, the population may converge in wrong direction and stuck to local optima. The effect of mutation is to reintroduce divergence into a converging population. Mutation operator selects one chromosome randomly from the population, then selects some genes using mutation probability and flips that bit. So mutation is a random operator that randomly alters some value. Mutation either explores some new points in the search space and leads population to global optima direction or alters value of the best chromosome and losses knowledge acquired till now. So mutation should be used rarely. Generally per gene probability of mutation is 0.001, 0.01, 0.02 etc..

19 GA Flowchart

The pseudo-code of GA can be given as follows:

1. Set the values of the parameters regarding population size, probability of crossover, probability of mutation, number of generations, and all the other parameters. 2. Generate random initial population of chromosomes. 3. Select two of the chromosomes as parents, with probability proportional to their fitness. 4. If crossover is used, combine the genes of these chromosomes using the crossover operator to form two children chromosomes. In the case no crossover is applied, the children chromosomes will be initially, just copies of the parent chromosomes. 5. Then apply the mutation operator to the children chromosomes, so that some (if any) random bits of the children chromosomes are inverted. 6. Repeat steps 4-6, until children chromosomes have been formed. 7. Repeat steps 3-7 until the specified number of generations have passed.

V.

20 TEST FUNCTIONS

Different types of Nine Test Functions are considered as given below in Table **??**. Threshold Acceptance Algorithm (TA), Simulated Annealing Algorithm (SA) and Genetic Algorithm (GA) are applied to these test functions.)=?? 1 + ?? 2 [0,1] Minimize F2 f(?? 1 , ?? 2)=?? 1 2 + ?? 2 2 [0,1] Minimize F3 f(?? 1 , ?? 2)=20 + ?? 1 2 + ?? 2 2 [0,1] Minimize F4 f(?? 1 , ?? 2)=?? 1 4 + ?? 2 4 + (2 × ?? 1 × ?? 2) [0,1] Minimize F5 f(?? 1 , ?? 2)=?? 1 2 + ?? 2 2 + (20 × ?? 1 × ?? 2) [0,1] Minimize F6 f(?? 1 , ?? 2 , ?? 3 , ?? 4)=?? 1 2 + ?? 2 2 + ?? 3 2 + ?? 4 2 [0,1] Minimize F7 f(?? 1 , ?? 2)=(?? 1 × sin ? ?? 1) + (?? 2 × sin ? ?? 2) [0,1] Minimize F8 f(?? 1 , ?? 2)=??

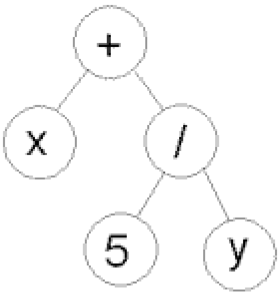
21 CONCLUSION

Genetic algorithm is probabilistic, heuristic, robust search algorithm premised on the evolutionary ideas of natural selection and genetic. Main idea behind the design of genetic algorithm is to achieve robustness and adaptiveness in real world complex problems. From the above results we have concluded that genetic algorithm is more reliable, strong and robust than threshold acceptance algorithm and simulated annealing algorithm.

22 VIII

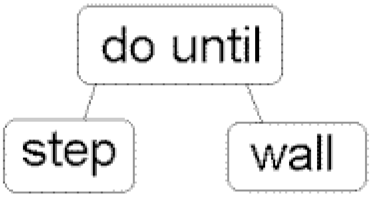
[1]

Figure 1:



1

Figure 2: Chromosome A 1



1

Figure 3: Chromosome A 1 .

I

26 I Volume XII Issue v v v v I Version ( )						
Global Journal of Researches in Engineering	Gen = 0 Create Initial Random Population Evaluate fitness of each individual in Population Reproduction Crossover Mutation	End	Function Name F1	9 Test Functions f(?? 1 , ?? 2	Constraints	Function Value
	Gen = Gen + 1	Designate best Chromosome found so far as solution				
NO	If Gen=N	YES				

[Note: © 2012 Global Journals Inc. (US)]

Figure 4: Table I :

Function Name	No. of Iterations		Time in Seconds	
	TA	SA GATA	SA	GA
F1	1506 5924 100	15	30	5
F2	1000 1637 51	9	12	2
F3	1000 1462 52	7	8	2
F4	1000 2620 51	7	13	2
F5	5918 5940 100	25	25	4
F6	2000 3483 64	17	19	3
F7	2220 1160 51	18	7	2
F8	1196 3079 51	9	12	2
F9	1000 1168 51	5	7	2

Figure 5:

[Elaine Rich and Kevin Knight 12] , www.iitk.ac.in/kangal/13.comisef.wikidot.com/concept:thresholdaccepting *Artificial Intelligence* Elaine Rich and Kevin Knight 12 (ed.) (Second Edition)

[Kaufmann ()] , Morgan Kaufmann . 1989.

[Jong and Spears ()] 'A formal analysis of the role of multi-point crossover in genetic algorithms'. K A Jong , W M Spears . *Annals of mathematics and Artificial intelligence* 1992. 5 p. .

[Davis] 'Adapting operator probabilities in genetic algorithms'. L Davis . *Proceedings of the 3rd International Conference on Genetic Algorithms*, (the 3rd International Conference on Genetic Algorithms) p. .

[Jong and Spears ()] 'An analysis of interacting roles of population size and crossover in genetic algorithms'. K A Jong , W Spears . *Parallel problem solving from nature-Proceedings of 1st workshop*, H P Manner (ed.) (Dortmund, Germany; Berlin, Germany) 1991. Springer-Verlag. 496 p. .

[Fogel ()] *Evolutionary Computation: Toward a New Philosophy of Machine Intelli¬gence*, D B Fogel . 1995. IEEE Press.

[Jong and Sharma ()] 'Generation gaps revisited'. K A Jong , J Sharma . *Foundations of Genetic Algorithms*, Darrell Whitley (ed.) 1992. Morgan Kaufmann. 2 p. .

[Goldberg and Richardson ()] 'Genetic algorithms with sharing for multimodal function optimization'. D E Goldberg , J Richardson . *Proceedings of the Second International Conference on Genetic Algorithms*, (the Second International Conference on Genetic AlgorithmsMassachusetts) 1987. Lawrence Erlbaum Associates. p. .

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

[Bramlette ()] 'Initialization, mutation and selection methods in genetic algorithm for function optimization'. M F Bramlette . *Proceedings of the Fourth International Conference on Genetic Algorithms*, R K Belew, L B Booker (ed.) (the Fourth International Conference on Genetic Algorithms) 1991. Morgan Kaufman. p. .

[Thesis and Mr (ed.)] *Nilesh Ghambhava 10. Fundamentals of Algorithmics by*, M E Thesis , Mr . Gilles Brassard and Paul Bratley (ed.)