Genetic Algorithm an Intelligent Approach for Routing

Noor Mohammad
Laxmi Prasanna M V
Shekhawati Engg College, Dundlod.

ABSTRACT
Genetic Algorithm approaches are adaptive heuristic search algorithms based on evolutionary ideas of natural selection and genetics. This paper focuses on different areas in computer networks their limitations and generalized solutions to optimize the areas to help the future demands. As today's Internet network does not meet the requirements of future integrated service networks that carry multimedia data traffic with high quality of service .So, it is necessary to develop high quality mechanism to check routing process, network traffic load and ensure Quality of Service (QoS) requirements. For resolving these problems Genetic Algorithm is a Global Optimized methodology, which we are going to solve using Traveling Sales Person problem.

1. INTRODUCTION AND OBJECTIVE
Routing is the process of selecting paths in a network along which to send data to physical traffic. Routing is performed to many kinds of networks, including the telephone networks, the internet and networks. Routing is the act of moving information across an internetwork from source to destination. Different routing algorithms are implemented for information transfer. As communication networks require complex resource management mechanism to stay operational, the previous routing techniques are not able to meet the requirements especially when there is traffic approaching some undesirable threshold. For better performance of network, there was a need for new solution.

No such research is done yet now in this area and still need continuous research as internet need is dynamic and increasing exponentially. Today's internet network does not meet the requirements of future integrated service networks that carry multimedia data traffic with a high quality of service(QoS). So, it is necessary to develop high quality mechanism to check the routing process and network traffic load and ensure QoS requirements. Research using various intelligent approaches is going on, so we propose intelligent approach like Genetic Algorithm in combination of other technologies which can be used for optimization as we can use the local heuristics to achieve optimal routing solution. Previously authors focused on traffic control concept related to how to minimize congestion. Under congestion, the performance of a network due to deteriorates, resources are wasted, delay increases and the predictability of network services is reduced so it is necessary to develop a new intelligent and adaptive algorithm for congestion control .For resolving this problems, we propose the approach of "Genetic Algorithm methodology" as a best solution.

2. PROBLEM DEFINITION
Communication networks require complex resource management mechanism to stay operational .So, it is considered network routing techniques is a key element of resource allocation. For better performance of network there was a need for new solution.There was a need for managing quality of service. Problem definition depends on available infrastructure on the required results. Hence to illustrate the importance of routing we have taken the example of Travelling Salesman Problem using Genetic Algorithm. In Travelling salesman problem (TSP) is a NP-Complete problem in which there are 'n' cities given with distances between them. Travelling salesman has to visit all of the cities exactly once and return to his home city. Task is to find a sequence of cities to minimize traveled distance. In other words, find a minimal Hamiltonian tour in a complete graph of N nodes.

3. PREVIOUS EXISTING TECHNIQUES
1) Off-line routing: With these routing, resulting routing solutions heavily dependent upon the accuracy of the traffic matrix and the calculations for large networks are generally very complex. Here highly dynamic networks pose problems like solutions produces offline might be very quickly out of date if the network stage diverges significantly from traffic predictions.

2) Adaptive/Dynamic Routing (fall between on and off line routing algorithm) here route calculations triggered directly by changes in network states such as: link failure, new hosts being added and routing tables is updated automatically.

3) Online routing: Here route calculations need to be made as demand arises, so incurring delay in call set up, on each route may itself be an NP-complete task depending upon the metrics used. Also, route calculations depend on extensive network state information which may be rapidly changing and needs to be accessible from throughput the network. For large network, information updates becomes very expensive to send and suffer from non-negligible delays.

4) Distributed routing (hop to hop routing): It is hard to apply complex heuristic in distributed manner and that global properties are hard to maintain like (prevention of loops and adhoc to end to constraints such as delay).

5) Source routing: First drawback is lacking of scalability. As hardware becomes larger, it becomes impractical to store and manipulate sufficient network state information in each node to obtain good routes. Second drawback is: source routing nodes will always be working with outdated information due to the propagation time of information updates in the network.

6) Hierarchical routing: It make able to useful routing decision based on aggregation. So give imprecise network state information.

7) Multicast routing: Due to need to manage the multicast group much of the difficulty in routing for multicast is with updating existing multicast trees when user join and leaves rather than generating complete tree from scratch.

8) Multi authority routing: Problem with this routing is to find a way of making use of restricted information to make adequate routing decisions when passing through domains controlled by several different authorities.

9) Quality of service routing: There are number of difficulties need to be dealt with. QoS connections often depend on many different parameters. The number of parameters along with large numbers of different applications to very heterogeneous traffic types, making traffic prediction difficult. QoS connections are very sensitive to the network states; this sensitivity reduces the usefulness of pre-determined routes. As the network state changes, the best route to take changes rapidly with it.

10) Making future reservation: Problem occurs when call times holding times are known in advance. If the holding time becomes known on connection time, the problem is considerably simplified since this information can be used to predict a clash with service reserved at some time in the future.

4. IMPLEMENTATION METHOD

Optimization:

NP-complete problems are hard and verifiable in polynomial times. We can solve these problems better with intelligent approaches these approaches are known as evolutionary approaches.

The heuristic techniques under the category "optimization" lead with maximum profits with low cost. It refers to search for better solution to a problem. Optimization deals with two branches of science-Applied Mathematics and Numerical Analysis.Eveloutinary approaches follow different techniques like GA, Simulated Annealing (SA), Tabu search (TS), Genetic programming (GP), Neural Networks (NN) and Fuzzy Logic (FL).

Genetic Algorithm uses a metaphor where an optimization problem takes the place of an environment and feasible solutions are considered as individuals living in that environment. In genetic algorithms, individuals are binary digits or of some other set of symbols drawn from a finite set. As computer memory is made up of array of bits, anything can be stored in a computer and can also be encoded by a bit string of sufficient length. Each of the encoded individual in the population can be viewed as a representation, according to an appropriate encoding of a particular solution to the problem .For Genetic Algorithm to find a best solution, it is necessary to perform certain operations over these individuals. This chapter discusses the basic terminologies and operators used in Genetic Algorithms to achieve a good enough solution for possible terminating conditions.

5. KEY ELEMENTS

The two distinct elements in the GA are individuals and populations. An individual is a single solution while the population is the set of individuals currently involved in the search process.

Individuals:

An individual is a single solution. Individual groups together two forms of solutions as given below:

1) The chromosome, which is the raw ' genetic ' information (genotype) that the GA deals.

2) The phenotype, which is the expressive of the chromosome in the terms of the model.

Solution Set Phenotype

Factor 1

Factor 2

Factor 3

Factor N

Gene 1

Gene 2

Gene 3

Gene N

Chromosome Genotype

1 0 1 0 1 0 1 1 1 0 1 0 1 1 0

6. REPRESENTATION OF A CHROMOSOME

A chromosome is subdivided into genes. A gene is the GA's representation of a single factor for a control factor. Each factor in the solution set corresponds to gene in the chromosome. A chromosome should in some way contain information about solution that it represents. The morphogenesis function associates each genotype with its phenotype. It simply means that each chromosome must define one unique solution, but it does not mean that each solution encoded by exactly one chromosome. Indeed, the morphogenesis function is not necessary bijective, and it is even sometimes impossible (especially with binary representation). Nevertheless, the morphogenesis function should at least be subjective. Indeed all the candidate solutions of the problem must correspond to at least one possible chromosome, to be sure that the whole search space can be explored. When the morphogenesis function that associates each chromosome to one solution is not injective. i.e., different chromosomes can encode the same solution. The representation is said to be degenerated. A slight degeneracy is not so worrying, even if the space where the algorithm is looking for the optimal solution is inevitably enlarged. But a too important degeneracy could be a more serious problem. It can badly affect the behavior of the GA. mostly because if several chromosomes can represent the same phenotype, the meaning of each gene will obviously not correspond to specific characteristics of a solution. It may add some kind of confusion in the search. Chromosomes are encoded by bits strings.

Genes:

Genes are the basic "instructions' for building a Genetic Algorithms. A chromosome is a sequence of genes. Genes may describe a possible solution to a problem, without actually being the solution. A gene is a bit string of arbitrary lengths. The bit string is a binary representation of number of intervals from a lower bound. A gene is the GA's representation of a single factor value for a control factor, where control factor must have upper bound and lower bound. This range can be divided into the number of intervals that can be expressed by the gene's bit string. A bit string of length 'n' can represent (2n-1) intervals. The size of the interval would be (range0/(2n-1)). The structure of each gene is defined in a record of phenotype parameters. The phenotype parameters are instructions for mapping between genotype and phenotype. It can be said as encoding a solution set into chromosome and decoding a chromosome to a solution set. The mapping between genotype and phenotype is necessary to convert solution sets from the model into a form that the GA can work with, and for converting new individuals from the GA into form that the model can evaluate.

Fitness:

The fitness of an individual in a generic algorithm is the value of an objective function for its phenotype. For calculating fitness, the chromosome has to be first decoded and the objective function has to be evaluated. The fitness not only indicates how good the solution is, but also corresponds to how close the chromosome is to the optimal one. In the case of multi-criterion optimization, the fitness function is definitely more difficult to determine. In multi-criterion optimization problems, there is often a dilemma as how to determine if one solution is better than another. What should be done if a solution is better for one criterion but worse for another? But here, the trouble comes more from the definition of a 'better' solution rather than from how to implement a GA to resolve it. If sometimes a fitness function obtained by a simple combination of the different criteria can give good result, it supposes that criterions can be combined in consistent way. But, for more advanced problems, it may be useful to consider something like Pareto optimally or others ideas from multi-criteria optimization theory.

Population:

A population is a collection of individuals. A population consists of a number of individuals being tested, the phenotype parameters defining the individuals and some information about search space. The two important aspects of population used in Generic

Algorithms are:

1. The initial population generation.

2. The population size.

For each and every problem, the population size will depend on the Complexity of the problem. It is often random initializations of population is carried. In case of a binary

Chromosome 1

Chromosome 2

Chromosome 3

Chromosome 4

1 1 1 0 0 0 1 0

0 1 1 1 1 0 1 1

1 0 1 0 1 0 1 0

1 1 0 0 1 1 0 0

coded chromosome this means, that each bit is initialized to a random zero or one. But there may be instances where the initialization of population is carried out with some known good solutions. Ideally, the first population should have a gene pool as large as possible in order to be able to explore the whole search space. All the different possible alleles of each should be present in the population. To achieve this, the initial population is, in most of the cases, chosen randomly. Nevertheless, sometimes a kind of heuristic can be used to seed the initial population. Thus the mean fitness of the population is already high and it may help the genetic algorithm to find good solutions faster. But for doing this one should be sure that the gene pool is still large enough. Otherwise, if the population badly lacks diversity, the algorithm will just explore a small part of the search space and never find global optimal solutions. The size of the population raises few problems too. The larger the population is, the easier it is to explore the search space. But it has established that the time required by a GA to converge is O(nlogn) function evaluations where n is the population size. We say that the population has converged when all the individuals are very much alike and further improvement may only be possibly by mutation. Goldberg has also shown that GA efficiently to reach global optimum instead of local ones is largely determined by the size of the population. To sum up, a large population is quite useful. But it requires much more computational cost, memory and time. Practically, a population size of around 100 individuals is quite frequent, but anyway this size can be changed according to the time and the memory disposed on the machine compared to the quality of the result to be reached. Population being combination of various chromosomes. Thus the above population consists of four chromosomes.

7. DATA STRUCTURES

The main data structures in GA are chromosomes, phenotypes, and objectiveFunction values and fitness values. This is particularly easy implemented when using MATLAB package as a numeric tool. An entire chromosome population can be stored in a single array given the number of individuals and the length of their genotype representation. Similarly, the design variables, or phenotypes that are obtained by applying some mapping from the chromosome representation into the design space can be stored in a single array. The actual mapping depends upon the decoding scheme used. The objective function values can be scalar or vectorial and are necessarily the same as the fitness values. Fitness values are derived from the object function using scaling or ranking function and can be stored as vectors.

8. SEARCH STRATEGIES

The search process consists of initializing the population and then breeding a new individual until the termination condition is met. There can be several goals for the search process, one of which is to find the global optimal. This can never be assured with the types of models that GA's work with. There is always a possibility that the next iteration in the search would produce a better solution. In some cases, the search process could run for years and does not produce any better solution than it did in the first little iteration. Another goal is faster convergence is desirable; however, the chance of converging on local and possibly quite substandard optima is increased. Apart from these, yet another goal is to produce a range of diverse, but still good solutions. When the solution space contains several distinct optima, which are similar in fitness, it is useful to be able to select between them, since combination of factors values in the model may be more feasible than others. Also, some solutions may be more robust than others.

9. ENCODING

Encoding is a process of representing individual genes. The process can be Performed using bits, numbers, trees,, arrays, lists or any other objects. The encoding depends mainly on solving the problem. For example, one can encode directly real or integer number.

Binary Encoding:

Chromosome 1	1 1 0 1 0 0 0 1 1 0 1 0
Chromosome 2	0 1 1 1 1 1 1 1 1 1 0 0

Binary encoding

The most common way of encoding is a binary string, which would be represented in figure 28. Each chromosome encodes a binary (bit) string. Each bit in the string can represent some characteristics of the solution. Every bit string therefore is a solution but not necessarily the best solution. Another possibility is that the whole string can represent a number. The way bit string can code differs from problem to problem. Binary encoding gives many possible chromosomes with a smaller number of alleles. On the other hand this encoding is not natural for many problems and sometimes corrections must be made after genetic operation is completed. Binary coded strings with 1s and 0s are mostly used. The length of the string depends on the accuracy.

In this,

- Integers are represented exactly.
- Finite number of real numbers can be represented.
- Number of real numbers represented increase with string length.

Octal Encoding

This encoding uses string made up of octal numbers (0-7).

Chromosome 1	03467216
Chromosome 1	15723314

Octal Encoding

3 Hexadecimal Encoding

This encoding uses string made up of hexadecimal numbers (0-9, A-F).

Chromosome 1	9CE7
Chromosome 1	3DBA

Hexadecimal Encoding

Permutation Encoding (Real Number Coding):

Every chromosome is a string of numbers, which represents the number in sequence. Sometimes corrections have to be done after genetic operation is completed. In permutation encoding, every chromosome is a string of integer/real values, which represents number in a sequence.

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Permutation Encoding

Permutation encoding is only useful for ordering problems. Even for this problems for some types of crossover and mutation corrections must be made to leave the chromosome consistent (i.e. have real sequence in it).

Value Encoding:

Every chromosome is a string of values can be anything connected to the problem. This encoding produces best results for some special problems. On the other hand, it is often necessary to develop new genetic operator's specific to the problem. Direct value encoding can be used in problems. Where some complicated values, such as real numbers are used. Use of binary encoding for this type of problems would be very difficult. In value encoding, every chromosome is a string of some values. Values can be anything connected to problem, form numbers, real numbers or chars to some complicated objects. Value encoding is very good for some special problems. On the other hand, for this encoding is often necessary to develop some new crossover and mutation specific for the problem.

10. BREEDING

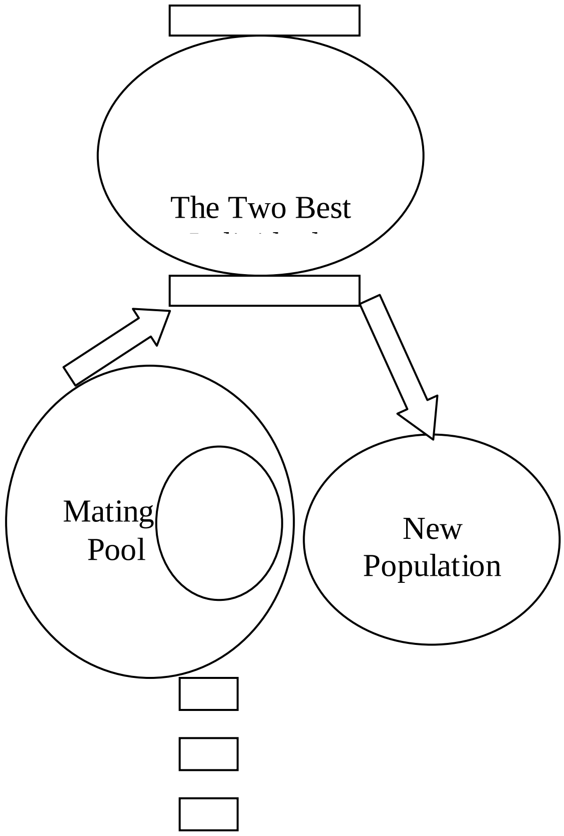
The Breeding process is the heart of the genetic algorithm. It is in this process, the Search process creates new and hopefully fitter individuals.

The breeding cycle consists of three steps:

1) Selecting parents.
2) Crossing the parents to create new individuals (offspring of children).
3) Replacing old individuals in the population with the new ones.

11. SELECTION

Selection is the process of choosing two parents from the population for crossing. After deciding on an encoding, the next step is to decide how to perform selection i.e., how to choose individuals in the population that will create offspring for the next generation and how many offspring each will create. The purpose of selection is to emphasize fitter individuals in the population in hopes that their off springs have higher fitness. Chromosomes are selected from the initial population to be parents for reproduction. The problem is how to select these chromosomes. According to Darwin's theory of evolution the best one s survive to create new offspring. The Fig.31 shows the basic selection process. Selection is a method that randomly picks chromosomes out of the population according to their evaluation function. The higher the fitness function, the more chance an individual has to be selected. The selection pressure is defined as the degree to which the better individuals are favored. The higher the selection pressured, the more the better individuals are favored. This selection pressure drives the GA to improve the population fitness over the successive generations. The convergence rate of GA is largely determined by the magnitude of the selection pressure, with higher selection pressure resulting in higher convergence rates.



Selection

Genetic Algorithm should be able to identify optimal or nearly optimal solutions under a wide range of selection scheme pressure. However, if the selection pressure is too low, the convergence rate will be slow, and the GA will take unnecessarily longer time to find the optimal solution. If the selection pressure is too high, there is an increase change of the GA prematurely converging to an incorrect (sub-optimal) solution. In addition to providing selection pressure, selection schemes should also preserve population diversity, as this helps to avoid premature convergence. Typically we can distinguish two types of selection scheme, proportionate selection and ordinal-based selection. Proportionate-based selection picks out individuals based upon their fitness values relative to the fitness of the other individuals in the population. Ordinal-based selection schemes select individuals not upon their raw fitness, but upon their rank within population. This requires that the selection pressure is independent of the fitness distribution of the population, and is solely based upon the relative ordering (ranking) of the population. It is also possible to use a scaling function to redistribute the fitness range of the population in order to adapt the selection pressure. For example, if all the solutions have their fitness in the range [999, 1000], the probability of selecting a better individual than any other using a proportionate-based method will not be important. If the fitness in every individual is brought to the range [0, 1] equitably, the probability of selecting good individuals instead of bad one will be important. Selection has to be balanced with variation form crossover and mutation. Too strong selection means sub optimal highly fit individuals will take over the population, reducing the diversity needed for change and progress; too weak selection will result in too slow evolution. The various selection methods are discussed as follows

Roulette Wheel Selection:

Roulette selection is one of the traditional GA selection techniques. The commonly used reproduction operate is the population reproductive operate where a string is selected from the mating pool with a probability proportional to the fitness. The principle of roulette selection is a linear search through a roulette wheel with the slots in the wheel weighted in proportion to the individual's fitness values. A target values is set, which is a random proportion of the sum of the fit nesses in the population. The population is stepped through until the target value is reached. This is only a moderately strong selection technique, since fit individuals are not guaranteed to be selected for, but somewhat have a greater chance. A fit individual will contribute more to the target value, but if it does not exceed it, the next chromosome in line has a chance, and it may be weak. It is essential that the population not be stored by fitness, since this would dramatically bias the selection. The above described Roulette process can also be explained as follows: The expected value of an individual is that fitness divided by the actual fitness of the population.

Each individual is assigned a slice of the Roulette wheel, the size of the slice being proportional to the individual's fitness. The wheel is spun N times, where N is the number of individuals in the population. On each spin, the individual under the wheel's marker is selected to be in the pool of parents for the next generation. 483 terminologies and operators of GA. This method is implemented as follows:

1) Sum the total expected value of the individuals in the population. Let it be T.
2) Repeat N times:

a) Choose a random integer 'r' between o and T.
b) Loop through the individuals in the population, summing the expected

Values, until the sum is greater than or equal to 'r'. The individual whose expected value puts the sum over this limit is the one selected.

Roulette wheel selection is easier to implement but is noisy. The rate of evolution depends on the variance of fitness's in the population.

Random Selection:

This technique randomly selects a parent from the population. In terms of disruption of genetic codes, random selection is a little more disruptive, on average, than Roulette wheel selection.

12. EVOLUTIONARY APPROACH

Each evolutionary application will follow the given components:

1) Representation
2) Evolution
3) Population
4) Parent Selection Mechanism
5) Variation operators, Recombination's and Mutation
6) Survivor Selection Mechanism(replacement)
7) General Scheme for Evolutionary Algorithm:

Begin

Initialize population with random candidate solutions;

Evaluate each candidate;

Repeat until (termination condition is satisfied);

Do

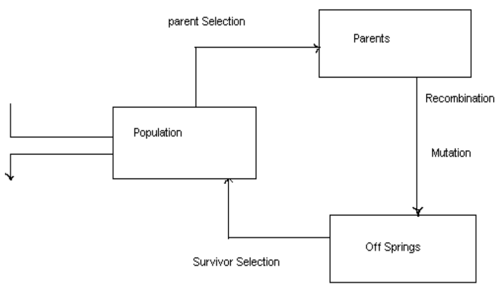
Select Parents;

Recombine pairs of parents;

Mutate the resulting offspring;

Evaluate new candidates;

Selec t individuals for next generation;



Block Diagram of Genetic Algorithm

13. CONCLUSION

Intelligent optimization techniques are very effective than traditional optimization technique. Dynamic programming problems are too complex to be solved exactly. These techniques seek for good solutions rather than best solution. This is the strength of all intelligent approaches that prevents them from myopically falling into holes in the mathematics.

In this paper we have discussed different problems and possibilities of routing optimization as a means of IP traffic engineering. Furthermore, the use of two metric types can be beneficial, enhancing network QoS even more. Although the actual gain depends on the topology and the load situation, it could still make a significant difference, especially when traffic is approaching some undesirable threshold. No much research is done yet now in this area and still need continuous research as Internet need is dynamic and increasing exponentially. Research using various intelligent approaches is going on, so we propose intelligent approach like GA in combination of other technologies which can be used for optimization as we can use the local heuristics to achieve optimal routing solution

14. REFERENCES

[1] Yadav Mahesh, Kumar Rajesh, "Limitations of Computer Networks and Optimization Approach", Proceeding of 2nd International Conference on Soft Computing, Nov 2008, Pages: 194 – 205, IET Alwar.

[2] Casetti Claudio, Mardente Gianluca, Mellia Marco and Munafo Maurizio, Cigno Lo Renato, "On-Line Routing Optimization for MPLS based IP networks" , IEEE Transaction, Jun, 2003, pages: 215-220.

[3] Curado Marilia, "Quality of Service Routing", IGI Global, 2008.

[4] Mellouk Abdelhamid, "Adaptive Routing Quality of Service Algorithm for Internet's Irregular Traffic", IGI Global, 2008.

[5] Steven Willmott, Boi Faltings, "Artificial Intelligence For Network Routing Problems", AAAI Transaction, May 18, 1999.

[6] Wang Ning, Ho Hon Kin, Pavlou George, Howarth Michael, "An Overview Of Routing Optimization For Internet Traffic Engineering" , IEEE Transaction, volume 10, No.1, Ist Quarter 2008.

[7] Riedl Anton, "A Hybrid Genetic Algorithm For Routing Optimization in IP Networks Utilizing Bandwidth and Delay Metrics" , IEEE Transaction, 2002.