Asian Info-communications Council

Document No. 13

23rd-27th April 2001

25th Conference (Shanghai)

Working Group 1

TITLE: **Network Structure Optimization by Using GA (Genetic Algorithm)**

Theme: **Network Evolution Technologies (LTA,TAEE)**

SOURCE: **New service and automation department, Research Institute of Post and Telecom (RIPT), Ha Noi, VIET NAM**

Summary

When the telecommunication market is more competitive, network planning and network design issues are considered as very important tasks for every service provider. Finding an optimized network structure is the critical issue in the whole network planning procedure. This paper reviews available algorithms such as Heuristics, Formal optimization and Genetic Algorithm (GA) that are good for solving the network structure optimization problem. Among these solutions, the GA is a promising one and applied in developing our unified suite of network planning and design tools named NetPlan. The GA algorithm works well when dealing with traditional copper cable network, PSTN network as well as new network technologies such as access network, ATM, and SDH/SONET/WDM. Based on some case studies, the GA performance is analyzed when comparing with other optimization techniques. Finally, the future research issues are provided in this paper.

1. Introduction

Network Planning has a very important role for Telecommunication operators. The operators are not only to think about the short-term revenues, but also their advantages over their competitors in 5 years or more in the future. Planning is the first thing operators must do before any major investment is made for a new network or upgrading old networks. Network Planning will find the optimum network possible to minimize network cost while satisfying required grade of service. By minimizing network cost, cost for each phone call will be reduced and make the operator become more competitive and attracts more customers. By satisfying required grade of service, customers will be happy and stay with the operator. Planning network carefully will also have a well-structured network and make operating and maintenance task easier and cheaper.

Planning is really important and algorithm is a very important part in the planning process. The algorithm will determine how close the output network compare to the real optimum network. In this article some algorithms will be reviewed and Genetic Algorithm will be examined in detail.

2. Review of alternative approaches

Currently there are several approaches to network design problems. Among those, the most well-known and commonly used ones are Heuristics, formal optimization (e.g., linear programming) and Genetic Algorithms. In this section, we will review these approaches as well as examine their strength and weaknesses from network design point of view.

a. Heuristic Algorithms

Heuristics are design principles obtained from experience, which are incorporated into algorithms. Since heuristics can be made quantitative and repeatable, alternative heuristics can be compared by implementing them on the same problem, then observing which yields the best solution. It is therefore, possible to refine heuristics, keeping what works well and discarding what does not. With automated design, many more alternatives can be tried and objectively compared.

Although some heuristics are specific to particular types of networks, ones that are based on principles that apply across many types of networks, are the most useful. One of the most widely used heuristics is the greedy algorithm. This algorithm chooses the best one it can at each stage during the optimization process.

The greedy algorithm is a broadly applicable heuristic based on the simple observation that inexpensive networks tend to contain inexpensive links. This principle is true for all types of networks. Despite of the fact that optimal result is not expected except for some trivial cases, the algorithm is expected to yield good results. It is quite possible that the least expensive network does not contain some of the least expensive links. It happens that in the case of this simple unconstrained network, the greedy algorithm does in fact yield the optimal solution. However, with added constraints, the greedy algorithm no longer yields the optimal solution.

The main advantage of heuristics is to allow us to obtain feasible solutions to difficult problems in a reasonable amount of time[1]. Instead of an exhaustive exploration of the entire solution space, consideration is limited to solutions with characteristics that appear to be good.

However, one of the setbacks of this approach is that it is based on observations, which may become invalid over time. When the observations no longer hold, the entire procedure will need to be entirely overhauled. In addition, it is hard to evaluate or prove accuracy and/or determine deterministic bounds on the results of a specific heuristic algorithm.

b. Formal Optimization

There are algorithms, which always produce optimal solutions. The problem with these, however, is that they only apply to only a limited class of problems; for problem outside this class, they do not work at all[1]. One such algorithm is the simplex method. This algorithm only works for the class of problems called linear programming problems. Linear programs are problems where both constraints and the objectives are weighted sum of the variables.

In this case, the solution space can be searched in a very orderly way, avoiding most of the possible solutions and reaching the optimal solution in a reasonable amount of time. Variants of this class are integer-programming problems - linear programs with additional constraint that the variables must be integers, mixed integer programming, etc.

The usefulness of having exact solution to an optimization problem using this kind of approach is usually deterred by its burdensome computation. Although different methods have been employed to reduce computational load, this technique still appears to be computationally expensive, and therefore, it is normally restricted to research purposes.

c. Genetic Algorithm (GA)

Genetic Algorithms (GA) has been found extremely useful for obtaining near optimal solutions for a variety of communication and computer network design problems. Many papers on application of GA to network design in general and network synthesis in particular have been published in the recent years[2][3]

GA approach has several advantages which truly give it a competitive edge over traditional approaches mentioned above. Firstly, GA provides a flexible framework for all optimization problems due to abstraction of its implementation phases. Within this framework, there is scope for different specialized implementations matched to the selected formulation of the problem. Changes to the problem can be done on the fly without restarting the optimization process from scratch which is different from traditional approaches. Another advantage of an evolutionary approach to problem solving comes in being able to generate good enough solutions quickly enough for them to be of use. In situations where simply applying brute force to search all possible solutions to problems of even modest size is likely to fail (e.g., NP complete problems), GA provides an effective and accurate means to solve them. Like heuristics, GA searching procedure is oriented to search solutions, which is promising to contain optimal solution. However, unlike heuristics, GA is not biased to any experiences and/or observations but only have to obey some generic rules set by network designers.

GA also has it share of problems that need to be dealt with in order to make it useful. Since GA simply represents the framework, it is totally up to the designers to provide specific implementation for each phase. If randomness of each phase is not assured, the procedure may yield results which may be locally optimal or not optimal at all. Besides, GA is normally not totally repeatable in the sense that it may produce different results every time. It is also difficult to tell when solution will actually converge.

3. General GA approach for optimizing telecommunication network structures

GA is the algorithm base on the Darwin's theory of evolution. This algorithm can be used to solve many math problems that include linear and nonlinear optimization problems. So, it can be used very effectively to find the optimize structure of telecommunication networks. This algorithm can be described as bellow:

Procedure Evolution-Program;

Begin

t:=0; {Set the starting time}

Create a group of chromosome P(t) randomly ;

while (not finish) **do**

Begin

Estimate the fitness value of every chromosome in P(t);

Select 2 chromosomes from P(t) with probability directly proportional to fitness values calculate from above; *{ The larger the chromosome's fitness value, the more likely it is selected}*

Crossover the chromosomes with probability of P_c to make new generation is P(t+1);

Mutate every chromosome of P(t+1) with probability of P_m;

t = t + 1;

End;

End.

This evolution program is a tractable algorithm. It always maintains a group of chromosomes (generation) P(t) = $\{ X_1^1, X_2^1, \ldots X_n^1 \}$ for every prospect t. Each chromosome holds a solution and different programs will use different ways of coding real optimization problem to chromosome.

Each solution X_i^1 will be evolved to find out a new better solution satisfying every constraint of the specific problem. At that time, a new generation of chromosome (t+1) will be found through the selection of best chromosomes. The mutation process will make some small changes in the new chromosomes (m_i: S'-> S). The crossover will make new chromosomes by combining the old chromosomes ($C_{ij}:S_1 \times S_2 \times ... \times S_n$->S).

After some generations, the optimization process will converge and the best chromosome found represents the best solution. When applying this algorithm to find out the optimal telecommunication network structure, this algorithm works very effectively. In this part, we will describe every step of GA when applying it to network planning tool.

a. Chromosome Encoding

This step is the most important part of the algorithm. It decides the speed and accuracy of the searching process. It includes two parts.

Part 1: Chromosome encoding. Make a law that every solution can be converted to a chromosome.

Part 2: Chromosome decoding. Decode a chromosome to a solution of the problem.

We have many ways to represent a network structure as a chromosome. Access networks, switching networks, cable networks etc... We can use several methods as binary coding, order coding, coding depend on value, tree coding...

Following is the method, which had been applied in the PSTN network structure optimization tool.

First all of exchanges are numbered. The the relation matrix A are created . In this matrix, A[i][j]=1 if there is a link between exchange [i] and exchange [j] and A[i][j]=0 if there is no link. It is very easy to see that A is the symmetrical matrix. Stretch the above triangle from left to right and top to down, we get a chromosome as the bellow example:

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

Fig 1. Relation matrix with above triangle.

In this case, the chromosome we get is:

101101010100111

b. Initial Population Generation

A group of initial solutions usually are created randomly from the set of solutions. However, if we create some good solutions in this initial generation, we can shorten running time. This is very important. This step also depends on the chromosome encoding

We can select a good solution first. After that, coding it into chromosome. In another way, we can create chromosome directly, but this method is more difficult and complicated. In this step, we usually use simple algorithms to find some good solutions. Sometimes, it can speed up the optimization process by two or three times.

c. Fitness evaluation

This step is used to find the fitness value of the chromosome. When apply this algorithm in Telecommunication Network Optimization, this step usually is Network Cost Evaluation which decides the accuracy of the optimization program.

d. Crossover

This step is used to generate new chromosome from the old ones. There are many ways of crossover, such as one point crossover, two-point crossover, mask crossover, order crossover, circle crossover, transpose crossover, tree crossover etc...Depending on each case, we can use one of the mentioned methods. With some networks, that are not very complicated, we can use simple way like one point crossover or two point crossover. But for more complicated networks, other methods are more suitable. For example, tree crossover, variable-length array single point crossover, List Order-Based Crossover, work more effectively.

Here is some example:

Parent chromosome 1	1101100100110110
Parent chromosome 2	0010111000011110
Son chromosome 1	1101111000011110
Son chromosome 2	0010100100110110

Fig 2 : The one point crossover method.

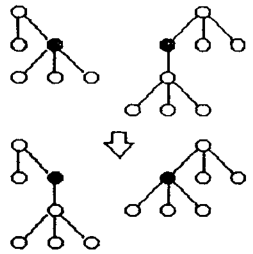


Fig 3 : The tree single point crossover method.

e. Directed mutation

This step is quite simple but very important. Because it makes the optimization process converge at near optimal result. Without it, the process will converge at a locally optimal result. The probability of the mutation must be small, otherwise, optimal process cannot converge.

f. Elitism and Pruning

This step will decide which chromosome will be kept, and which will be rejected. All chromosomes, which have higher fitness values will be kept. All chromosomes, which have lower fitness values will be rejected and be replaced by new chromosomes created from the crossover step.

g. Repair Mechanism

This step will repair most of chromosomes after crossover and mutation. After crossover and mutation. the new chromosome created which likely have a wrong structure, which means that when decoding it, the result represent an infeasible network structure. So it needs to be repaired. The number of repair in every chromosome is as small as possible otherwise optimization process cannot converge.

h. Loop Structure

After these steps, we choose the best chromosome found and store it for later use. To continue, the optimize process needs to return to the Fitness evaluation step. This is called Loop structure and after some times, if we still cannot find better solution, the best solution is assumed to be the optimal solution.

4. Case studies in the NetPlan implementation

Our unified suite of network planning and design tools named NetPlan has applied GA to solve many optimization problems in PSTN network planning, access network planning …It has been applied to plan the Ha Nam Province's Telecommunication Network and the result is quite good.

The following is a comparison of Netplan to other Planning tools.

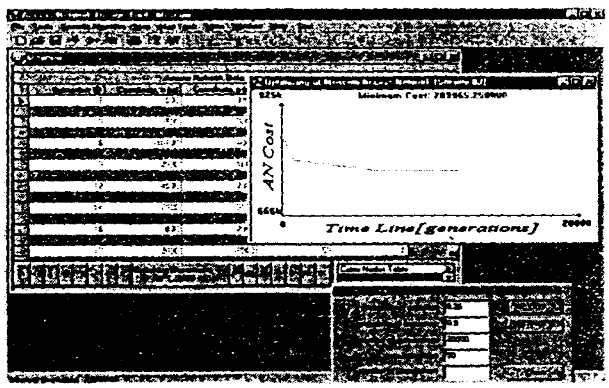
Functions	Features	PlanITU	Telstra	Planicom	NetPlan
Forecasting of Service	Voice service forecast	✓	✓	No	✓
	Non-Voice services forecast	No	No	No	✓
	Make reports with friendly GUI	*No*	No	No	✓
Forecasting of Traffic	Voice service forecast	✓	✓	✓	✓
	Non-Voice services forecast	No	No	No	✓
	Make reports with friendly GUI	No	No	✓	✓
Dimentioning of PSTN Switching Networks	*Optimarization of Network structure*	No	No	No	✓
	Dimentioning of Switching Network	✓	✓	✓	✓
	Graphic functions	✓	No	✓	✓
	Friendly GUI	No	No	No	✓
Dimentioning of ATM networks	Optimarization of Network structure	No	No	No	✓
	Optimarization of Traffic route	No	No	No	✓
Dimentioning of Transmission networks	*SDH network with Ring topo*	No	No	✓	✓
	SDH network with Mesh topo	No	No	✓	No
	SDH network simulation	No	No	✓	No
	Dimentioning of PDH/Cable network	✓	-	✓	✓
	Friendly GUI	No	-	✓	✓

Dimentioning of Access networks	*Copper cable media*	✓	No	No	✓
	Optical/ WLL / xDSL media	No	No	No	✓
	Friendly GUI	No	No	No	✓

Fig 4. Comparison of characteristics between NetPlan and other Planning tools

Here is the case study using NetPlan to design Ha Nam's Access Network. The tool we used called AND (*Access Network Dimensioning tool*) which employs GA to find optimal network structure.

Fig 5.AND Interface when running



The input data includes:

- Position and capacity of every kind of cable cabinet and cable box.
- The length, position, capacity and number of pairs of cable are not in used in all cable branches.
- The routing table for all of cable cabinets and cable boxes.
- Coordinates of exchanges and remote location units.
- Number of new subscribers.
- The distribution of new subscribers in the are
- Service requirements

The output data includes:

- The configuration of the network using selected technology.
- The number and type of equipment
- The results obtained from access network dimensioning process.
- The investment model corresponding to different access media.
- Total investment for the whole network

And this is the view of Ha Nam's access network after running *AND*.

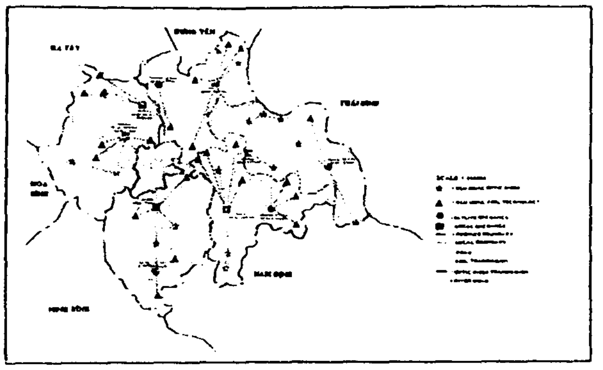


Fig 6. The result of Ha Nam's access network in 2010

5. Future research works

Although we have applied this algorithm quite widely in our unified suite of network planning, we still have to continue to research GA and the method to applied it in others tool like IP network planning tool, Mobile network planning tool etc...Furthermore, we still have to improve speed of GA when applying it. The time running of this algorithm is one of its weak points. We will need to find some other methods of encoding chromosome to shorten running time and improve optimality of result.

6. Conclusions

Genetic Algorithm (GA) is the algorithm that proves to be suitable for solving the network structure optimization problem. Furthermore it is a promising one for other optimization problems. Although it works very well in our software, we still want to assert about the optimality of result we get from GA.

References

[1] Aaron Kershenbaum, "Telecomminications Network Design Algorithms", *Mc Graw-Hill, 1993 Singapore.*

[2] L.T.M. Berry, B.A.Murtagh, S.J.Sugden and G.B.McMahon, Application of genetic based algorithm for optimal design of tree-structured communications networks in: Proceeding of Regional Teletra ± c Engineering Conference of the International Teletra ±c Conggress, South Africa, (1995) 361-370

[3] C.C Palmer, An approach to a problem in network design using genetic algorithms, Networks, 26(1995) 151-163

Author's / presenter's Profile :

Tan Ba Le was born in Vietnam in 1977. He received the B.E. in Electronic and Telecommunication engineering from Hanoi University of Technology, Vietnam, 2000. He joined Vietnam Research Institute of Posts and Telecommunications (RIPT) as a researcher in 2000. His main research areas are the network planning, software development for telecommunications systems.

Buu Hao Tran was born in Vietnam in 1972. He received the B.E, M.E. and Ph.D in electrical engineering from the Leningrad University of Technology, the former Soviet Union, in 1993, 1995, and 1999 respectively.

He joined Vietnam Research Institute of Posts and Telecommunications (RIPT) as a researcher in 1999. His main research areas are the network planning and design, software development for telecommunications systems, and Internet telephony

Hieu Trung Tran was born in Vietnam in 1976. He received the B.E. in electrical engineering with first class honors from Royal Melbourne Institute of Technology, Australia, 1998. He joined Vietnam Research Institute of Posts and Telecommunications (RIPT) as a researcher in 1999. His main research areas are the network planning and design and software development for telecommunications systems.

Vinh Quang Nguyen was born in Vietnam in 1976. He received the B.E. in electrical engineering with first class honors from Royal Melbourne Institute of Technology, Australia, 1999. He joined Vietnam Research Institute of Posts and Telecommunications (RIPT) as a researcher in 2000. His main research areas are the network planning and design and software development for telecommunications systems.

Dzung Tran Duy was born in Vietnam in 1978. He received the B.E. in Electronic and Telecommunication engineering from Ho Chi Minh University of Technology, Vietnam, 2000. He joined Vietnam Research Institute of Posts and Telecommunications (RIPT) as a researcher in 2000. His main research areas are the network planning, software development for telecommunications systems and Network Simulation.

Dzung Van Dinh was born in Vietnam in 1966. He received the B.E. in electrical engineering with first class honors from the Odessa Institute of Telecommunications Technology, the former Soviet Union, 1989. In 1996, he finished his Master of Engineering in Telecommunications Engineering course at the University of Technology, Sydney (UTS), Australia.

He joined Vietnam Research Institute of Posts and Telecommunications (RIPT) as a researcher in 1990. His main research areas are the mobility management for wireless ATM networks, communications protocols performance, network simulation, and software development for telecommunications systems.