

A GENETIC ALGORITHM FOR JOINT OPTIMIZATION OF SPARE CAPACITY AND DELAY IN SELF-HEALING NETWORK

Sam Kwong and H W Chong

Department of Computer Science
City University of Hong Kong, 83 Tatchee Avenue, Hong Kong

ABSTRACT

This paper presents the use of multi-objective Genetic Algorithms (mGA) to solve the capacity and routing assignment problem arising in the design of self-healing networks using the Virtual Path (VP) concept. Past research has revealed that Pre-planned Backup Protection method and the Path Restoration scheme can provide a good compromise on the reserved spare capacity and the failure restoration time. The aims to minimize the sum of working and backup capacity usage and transmission delay often compete and contradict with each other. Multi-objective Genetic algorithm is a powerful method for this kind of multi-objective problems. In this paper, a multi-objective GA approach is proposed to achieve the above two objectives while a set of customer traffic demands can still be satisfied and the traffic is 100% restorable under a single point of failure. We carried out a few experiments and the results illustrate the trade-off between objectives and the ability of this approach to produce many good compromise solutions in a single run.

1. INTRODUCTION

In "self-healing" network, what should be the optimal amount of spare capacity? Reducing network protection costs while maintaining an acceptable level of survivability is one of the main objectives of the network planners. The transmission delay from a source node to a destination node depends very much on the paths chosen. Every customer likes to have a route with minimum delay, the TELCO operator cannot promise every customer that the route is the shortest. It becomes an interesting problem for one to choose routes that will compromise the interests of the TELCO operator and different customers.

The above problem can be regarded as the combination of two sub problems:

1. The objective to minimize capacity subject to a constraint on delay with given a network topology and assigned traffic requirements.
2. The objective to minimize delay subject to a constraint on the total capacity with a given network topology and assigned traffic requirements.

Obviously, these two sub-problems are dependent and cannot be easily solved without considering each other's existence. That is an issue that makes the above problem a relatively hard one, which is not easy to be solved using classical optimization methods.

To tackle the above problems, a GA-based multi-objective optimization approach is presented in this paper. We will see how the method obtains a Pareto set of solutions in that any single set of solution can be freely chosen according to the fulfillment of the system requirements. We will also see that spare capacity and delay are jointed optimized and to provide a highly available service.

2. RELATED WORK

The capacity, routing assignment and transmission delay arising in the design of self-healing network is a combinatorial optimization problem. It involves the assignment of capacities to the links, the routing of requirements on these links and minimum of transmission delay. Ideally, all these are jointed optimized.

In [1-4], different methods are proposed to optimize capacity allocation and routing assignment in order to minimize the cost. They solve the problem using linear programming techniques. These methods suffer from the usual disadvantages of linear/integer programming: a lot of constraints need to setup, a lot of variables need to manipulate and an intensive computation is needed.

Minimizing the total capacity and total transmission delay are the same important in the self-healing networks. But previous work focus more on the capacity allocation, routing assignment than transmission delay. In some literatures, transmission delay is considered as a constraint but not an objective function for minimization. By transforming the delay constraint, the problem can be

easily formulated as a multi-objective optimization problem.

In our former work [5], we proposed a genetic algorithm based method to solve the capacity and routing assignment problem arising in the design of self-healing networks using the Virtual Path (VP) concept. This approach can avoid most of the problems inherent in linear programming and we showed that GA based approach is better than other heuristic approaches. However, we only tackled the capacity problem but we did not take the delay into consideration. In this paper we will further use a GA-based multi-objective optimization approach to solve the problem and it has a significant advantage over other approaches.

3. PROBLEM FORMULATION

In this research, we will model the network as an undirected graph $G = (V, E)$, where V is a set of nodes and E is a set of links. Each link is a pair of oppositely directed arcs or fibers. We don't model the network as a directed graph because most of the communication networks are generally bi-directional. Each site in the network will be represented as a node. Each node will be assigned a number, as its ID. From now on, if not specified, we assume the bandwidth of each link is B units.

We have made the following assumptions for this research:

- The target network of study is an *ATM* network.
- A single point of failure in the network is assumed.
- It is assumed the network can restore all the affected traffic under this single point of failure.
- Each branch of a multicast circuit requires the same bandwidth (however, different multicast circuits can have different bandwidths)
- Link transmission delay is directly proportional to the link distance

The notations used in the problem formulation are:

N	A set of nodes of the network
A	A set of directed arcs of the network
C_a	Capacity of arc a , where $a \in A$
Π	A set of source-destinations multicast traffic requirements
WVP	A set of multicast working virtual paths
BVP	A set of multicast backup virtual paths
$ \Pi $	Total number of multicast traffics demanded
$ WVP $	Total number of multicast working virtual paths
$ BVP $	Total number of multicast backup virtual paths
Π_i	i -th source-destinations multicast traffic in Π

WVP_i	i -th multicast working virtual path in WVP to satisfy Π_i
BVP_i	i -th multicast backup virtual path in BVP to backup WVP_i
$WVP_i(a)$	Capacity of arc a would be used by WVP_i , where $a \in A$
$BVP_i(a)$	Capacity of arc a would be reserved by BVP_i , where $a \in A$
$WVP_i.delay$	Total transmission delay of WVP_i
$BVP_i.delay$	Total transmission delay of BVP_i
$O(\Pi_{ij})$	Source node of the j -th branch of Π_i
$O(WVP_{ij})$	Source node of the j -th branch of WVP_i
$O(BVP_{ij})$	Source node of the j -th branch BVP_i
$D(\Pi_{ij})$	Destination node of the j -th branch of Π_i
$D(WVP_{ij})$	Destination node of the j -th branch of WVP_i
$D(BVP_{ij})$	Destination node of the j -th branch BVP_i
Π_{ij}	Bandwidth demanded by j -th branch of Π_i
$WVP_{ij}.bandwidth$	Bandwidth of j -th branch of WVP_i
$BVP_{ij}.bandwidth$	Bandwidth of j -th branch of BVP_i
$I-WVP_{ij}$	It is the intermediate virtual path of WVP_{ij} – with source and destination nodes excluded.
$I-BVP_{ij}$	It is the intermediate virtual path of BVP_{ij} – with source and destination nodes excluded.

The multicast routing problem is described as follows.

Objective one: To find WVP to satisfy the requirement Π , and the BVP to provide alternative routes that can restore the entire failed multicast working virtual paths in WVP under single point of failure of the network.

Objective two: To minimize the transmission delay between nodes.

The objectives function are given as:

$$1. \quad \text{Min} \left\{ \sum_{a \in A} \sum_{i=1}^{|\Pi|} [WVP_i(a) + BVP_i(a)] \right\}$$

$$2. \quad \text{Min} \left\{ \sum_{i=1}^{|\Pi|} [WVP_i.delay + BVP_i.delay] \right\}$$

Subject to the following constraints:

$$1. \quad |\Pi| = |WVP| = |BVP| \quad \text{AND}$$

$$O(\Pi_{ij}) = O(WVP_{ij}) = O(BVP_{ij}) \quad \text{AND}$$

$$\Pi_{ij}.bandwidth = WVP_{ij}.bandwidth = BVP_{ij}.bandwidth$$

where $\forall i = 1$ to $|\Pi|$ and $j = 1$ to degree of multicast of i -th multicast traffic demand

2. $I-WVP_{ij} \cap I-BVP_{ij} = \emptyset$ in link-disjointed or node-disjointed sense.

where $\forall i = 1$ to $|\Pi|$ and $j = 1$ to degree of multicast of i -th multicast traffic demand

$$3. \quad \sum_{i=1}^{|\Pi|} [WVP_i(a) + BVP_i(a)] \leq C_a \quad \forall a \in A$$

Constraint (1) ensures that the *WVP* found can satisfy Π , and the *BVP* can support *WVP* under a single point of failure of the network. Constraint (2) is to ensure that the backup virtual path is completely link-disjointed or node-disjointed from the working virtual path. Constraint (3) is to make sure that the capacity of any arc can satisfy the bandwidth used by *WVP* and reserved by *BVP*.

In [5], we propose a single objective approach with the objective function is given as the following,

$$\text{Min} \left\{ k \sum_{a \in A} \left[WVP_i(a) + BVP_i(a) \right] + (1-k) \sum_{i=1}^{|I|} \left[WVP_i \text{delay} + BVP_i \text{delay} \right] \right\}$$

where k is the weights to each of the objectives to indicate their importance in the problem, We set k to 1 if we are interested in the capacity assignment optimization problem, if we want to study the shortest path (i.e. delay) optimization problem, then k is set as 0. For multiple objectives, we have to tune k constantly in order to get the objectives jointly optimized. This method is very subjective, may over-simplify the behavior of the objectives, and it is often hard to find weights which can accurately reflect the situation. Thus, we propose to use mGA to solve this problem.

4. DESIGN OF GENETIC ALGORITHM

In this part, a multi-objective genetic algorithm is described. The whole process of GA can be summarized as below.

1. Initialization

- Set the population size be N_p
- Set the maximum allowed number of generation be G_{max}
- Set the crossover probability be p_c
- Set the mutation probability be p_m
- Set $G=0$ where G is the generation counter

2. The GA process

Step 1: Generate a population of chromosomes

Step 2: Fitness assignment and sharing

Step 3: Perform selection

Step 4: Perform crossover

Step 5: Perform mutation

Step 6: Fitness assignment and sharing

Step 7: Perform replacement

Step 8: Set $G = G + 1$. If $G > G_{max}$, terminate.

Otherwise, go to Step 3.

In the following, a detail description of chromosome representation, population pool initialization, fitness assignment, fitness sharing, selection, crossover, mutation and GA parameters will be provided.

4.1. Chromosome representation

In our optimization problem, a solution is a set of working virtual paths (*WVPs*) and a set of backup virtual paths (*BVPs*) for the multicast traffic, which can minimize the sum of working and backup capacity and the total transmission delay of the network. The description of our method of chromosome encoding is given as follow.

- Each chromosome consists of genes that each gene represents a set of *WVPs* and *BVPs* of the same multicast traffic that satisfy a particular customer's multicast traffic demand. Note that the number of genes in the chromosomes should be equal to the total number of customer traffic demands.
- Since multicast traffic is viewed as a number of unicast traffics leaving from the same source node but different destination nodes, we design an entity (let's call it a sub-gene) to represent a solution for a unicast traffic. Then, each gene consists of sub-genes that each sub-gene represents a *WVP* and a *BVP* that satisfy a particular customer's unicast traffic demand. As mentioned before, we defined the degree of multicast of a particular customer's multicast traffic as the total number of the unicast traffic of this customer. Then, the number of sub-genes in a gene is equal to the degree of multicast of a particular multicast traffic.

4.2. Population pool initialization

Before the genetic algorithm starts, we need to generate degrees of multicast, source nodes and destination nodes randomly based on the total number of virtual paths specified and the given topological information of an *ATM* network. Each possible path should not include an intermediate node more than once so that all the paths generated will not waste unnecessary resources. Furthermore, each source-to-destination route should have at least two paths (one for Working Virtual Path and one for Backup Virtual Path) with its minimum hop number. However, if it contains only one path with its minimum hop number, we accept all its possible paths with one more hop. In the beginning of genetic algorithm, a Working Virtual Path and a Backup Virtual Path of each source-to-destination route are randomly selected from its corresponding set of found paths and then allocated to the chromosomes.

4.3. Fitness Assignment

The fitness function links the Genetic Algorithm to the problem to be solved, it is used to measure the goodness of the chromosomes in the evolutionary process. The Pareto-based ranking method proposed by Fonseca and Fleming [6] is adapted in this paper.

Assuming that chromosome I is dominated by other p chromosomes in the population, its rank is determined as

$$Rank(I) = I + P$$

The Pareto-based ranking can correctly assign all non-dominated chromosomes with the same fitness values. However, the genetic diversity of the population can be lost due to stochastic errors in the selection process. The goal of an mGA is to find a population composed of non-dominated genotypes evenly distributed along the Pareto-front defining the trade-off between objectives. To achieve the even distribution of the population across the front, fitness-sharing methods are adopted in our design to maintain genetic diversity.

4.4. Fitness Sharing

Fitness sharing decreases the increment of fitness of densely populated solution space and shares the fitness with other space. It helps genetic algorithm search various space and generate more diverse solutions. With fitness sharing, the genetic algorithm finds more diverse solutions although some of the solutions are not good. Let f_i be the fitness of an individual i , $sh(d_{ij})$ be sharing function, and M be the population size, then the shared fitness fs_i is computed as :

$$fs_i = \frac{f_i}{\sum_{j=1}^M sh(d_{ij})}$$

The sharing function $sh(d_{ij})$ is computed using the distance value d_{ij} that means the difference between individual i and j as follows:

$$sh(d_{ij}) = \begin{cases} 1 - \frac{d_{ij}}{\sigma_s} & \text{for } 0 \leq d_{ij} < \sigma_s \\ 0 & \text{for } d_{ij} \geq \sigma_s \end{cases}$$

where σ_s describes the sharing radius. If the difference is larger than σ_s , they do not share the fitness.

4.5. Genetic operators

Selection

Elitist fitness proportionate selection, using the roulette-wheel algorithm, was implemented, using a simple fitness scaling whereby the scaled fitness f is

$$f = \frac{f - f_{worst}}{f_{best} - f_{worst}}$$

where f_{worst} is the fitness of the worst individual in the current population. Overall, the algorithm is elitist, in the sense that the best individual in the population is always passed on unchanged to the next generation, without undergoing crossover or mutation.

Crossover

We have designed three types of crossover operations, namely:

1. Chromosome Crossover
2. Direct Gene Crossover
3. Indirect Gene Crossover

The strategy of using the above three crossover operations is to use one type of crossover operation first, if the operation gives two valid offspring chromosome, then the crossover mechanism is said to be completely; otherwise, we will try the other types of crossover operations. In order to avoid using any one type of crossover operation more than the others, three operations are selected randomly.

• Chromosome crossover

The purpose of the Chromosome Crossover is to explore whether the different combinations of genes can get chromosomes with better fitness. This type of crossover involves exchange of genes between two selected chromosomes. To be fair, the crossover point of the two parents chromosomes is selected randomly.

• Direct gene crossover

For Direct Gene Crossover, it is aimed to create new combinations of sub-genes and thus new chromosomes. Only one gene is randomly selected in each chosen pair of chromosomes for performing this type of crossover.

• Indirect gene crossover

This type of crossover mechanism is not brought from the conventional Genetic Algorithm. It is dedicated to the capacity and routing assignment problem of a self-healing network. Before this type of crossover type is performed, we need to select a gene and a sub-gene inside the selected gene. If the Working Virtual Paths or Backup Virtual Paths inside a selected sub-gene of a selected gene of the two parent chromosomes have same intermediate node excluding source node and destination node, the path segment after the intermediate node (i.e. all nodes next to the intermediate node) will be exchanged between two parent chromosomes. Otherwise, the two parent chromosomes will remain unchanged.

Mutation

Mutation is another important activity in a genetic algorithm. It aims to introduce some changes on genes to the chromosomes to avoid being trapped in a local optimum. Mutation probabilities are utilized to choose a set of chromosomes to perform mutation. Before mutation is implemented, we need to select a gene and a sub-gene inside the selected gene. Then, a new Working Virtual Path and Backup Virtual Path are found to replace the current paths inside the chosen sub-gene. When a sub-gene is changed, the corresponding gene and chromosome is also changed. If the resulting chromosome is a valid one, then it will replace the selected chromosome, else no change is made.

GA Parameters

Several runs were carried out in order to evaluate the reliability of the solutions. Table 1 presents the parameters values used in this approach.

Table 1. GA parameters

Parameter	Value/Type
Population Size, N_c	50
Maximum generations, G_{max}	1500
Selection	Roulette Wheel selection
Crossover	1-point
Crossover probability, p_c	0.8
Mutation probability, p_m	0.2

5. EXPERIMENTAL RESULTS

To validate the effectiveness of the proposed approach, we have performed experiments in the network shown in figure 1. Some *ATM* network parameters are shown in Table 2. These experiments are based on customer traffic demands of 100 working virtual paths. The total delay and total capacity of those rank 1 chromosomes in the final generation are depicted in figure 2. A Pareto optimal set is clearly obtained by the multi-objective GA-based approach.

Table 2. *ATM* network parameters

Item	Value/Type
Maximum capacity of each link	100
Degree of multicast traffic	Random choice from the interval 1 – 4
Disjoint method between WVP and BVP	Node-disjoint

6. CONCLUSIONS

In this paper, instead of using Linear/Integer Programming as an optimization methodology, a multi-objective GA approach was used in our network optimization problem. Based on a set of input customer traffic demands, the algorithm can produce a set of working and backup Virtual paths which can satisfy the demands and minimizing the total capacities and total delays simultaneously.

We carried out experiments on an *ATM* self-healing network and the result illustrates trade-off between two objectives and demonstrates the ability of the approach to concurrently produce many good compromise solutions in a single run. From the result, we can conclude that GA-based multi-objective optimization method is suitable and efficient for self-healing network optimization problem.

Acknowledgement

This work is supported by the City University research grant 7001181

References

- [1] Shanzhi Chen, Shiduan Cheng, Bin Chen, and Junliang Chen, "An Efficient Spare Capacity Allocation Strategy for *ATM* Survivable Networks", *GLOBECOM 1996*, pp.442-446, 1996.
- [2] Kazutaka Murakami and Hyong S. Kim, "Joint Optimization of Capacity and Flow Assignment for Self-Healing *ATM* Networks", *Communications, 1995. ICC '95 Seattle, 'Gateway to Globalization', 1995 IEEE International Conference on*, 18-22 June 1995, vol.1, P216 – 220
- [3] Rainer R. Iraschko, M.H. MacGregor, and Wayne D. Grover, "Optimal Capacity Placement for Path Restoration in STM or *ATM* Mesh-Survivable Networks", *IEEE/ACM Transactions on Networking*, vol. 6, no. 3, p.325-336, 1998
- [4] Yijun Xiong and Lorne G. Mason, "Restoration Strategies and Spare Capacity Requirements in Self-Healing *ATM* Networks", *IEEE/ACM Transactions on Networking*, vol. 7, No.1, pp.99-110, 1999.
- [5] S.Kwong, S.S. Chan, "A Fault-tolerant Multicast Routing Algorithm In *ATM* Network", *Proceedings of the Genetic and Evolutionary Computation Conference*, Las Vegas, Nevada, July 10-12 2000, p582-589
- [6] Fonseca C. M., and Fleming, P. J., "Genetic Algorithms for multiobjective optimization: Formulation, discussion and generalization", *Proceedings of the 5th International Conference on Genetic Algorithms*, Morgan Kaufmann Publishers, Inc., San Mateo, pp. 416-423, (1993).

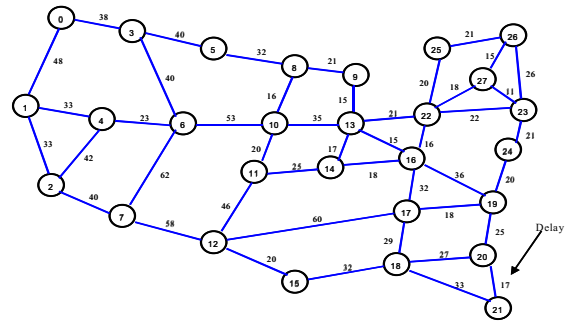


Figure 1. The 27-node, 45 link Network

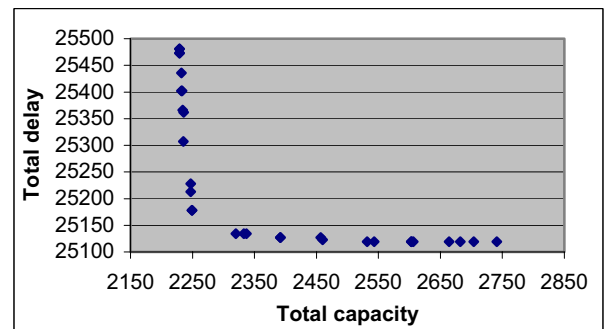


Figure 2. Final pareto front of Network