OPTIMIZATION MODELS FOR COMMUNICATION NETWORK DESIGN

Iryanto & Herman Mawengkang

Department of Mathematic, University of Sumatera Utara, Medan 20155

Abstract

The use of both Genetic Algorithms and Linier Programming to solve the general problem of communication system design is considered. The network synthesis problem is known to be NP–complete and the combinatorial nature of it lends itself to genetic algorithms rather than conventional mathematical programming approaches. Once a network topology is established, linier programming can be used to optimize network flows to satisfy specified origin-destination demands.

Keywords: Network Design, Genetic Algorithms

1. Introduction

With the advent of the information age there has been increased interest in the efficient design of communication networks. Network design problems occur in many field of communication, ranging from the old style analogue telephone networks to more modern applications such as wide - area and local – area computer networks, ISDN networks for multi – media transmission, and digital cellular networks for mobile phone communications. They all have a similar intent: how to carry the expected traffic flow from origin to destination at minimum cost. A relevant survey is Minoux [6].

If one imagines the communication requirement as a set of nodes representing origin and destinations, the primary problem is to join the nodes together in the most efficient manner. This is not an easy problem; for event a small number of nodes there are many thousands of ways of joining them together. This type of problem is termed NP – *complete*.

Given that we cannot investigate all possible ways of linking the nodes, the challenge is to devise a method for investigating a restricted number that nevertheless obtains a good solution. Heuristic methods are often used, which are relatively fast and can be devised to produce a good (if not optimal) solution.

A topology which has been the subject of much research interest is the tree network, which connects all nodes but does not contain any cycle. Obtaining good solutions to this type of network is a difficult task and we have used a Genetic Algorithms (GA) approach which is described in section 3.

In many applications however, it is necessary to add redundancy into the network to ensure reliability; if one link fails it would still be possible to connect from origin to destination. We have used a combination of GA and linear Programming (LP) to tacle this problem; and these approaches are discussed in sections 3 and 4.

Some success has been achieved by our efforts, and computational experience is discussed in section 5, and our approach may be compared with other approaches described in the literature, such as Gavish [3], Kershenbaum and Peng [5], Sharma et al. [8], and Balakrishnan et al. [2].

2. The network design problem

The design problem is to satisfy all traffic requirements at minimum cost. For a tree network synthesis problem involving n nodes, there are n^{n-2} possible tree structures, e.g. one hundred million possibilities for a network as small as 10 nodes. The information required to formulate the problem is the traffic demand between each origin and destination (O – D) pair, and the linier cost function for carrying traffic on each (possible) link (i,j) between nodes i and j.

Define:

$$F^{pq} = \text{total traffic flow between OD – pairs } (p,q) \tag{1}$$

$$x_{ij} = \begin{cases} 1 & \text{if there is a link joining nodes } i \text{ and } j \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$c_{ij} = \text{cost per unit of capacity on link } (i,j) \tag{3}$$

If we restrict the problem to minimizing total flow in a tree network, it can be formulated as:

Minimize

$$\sum_{p=1}^{n}\sum_{q>p}^{n} F^{pq} \sum_{i=1}^{n-1}\sum_{j=i+1} c_{ij} y_{ij}^{pq} \tag{4}$$

subject to

$$x_{ij} \in \{0,1\} \tag{5}$$

$$\sum_{i-1}^{n-1}\sum_{j=i+1}^{n} x_{ij} = n-1 \tag{6}$$

where

$$y_{ij}^{pq} = \begin{cases} 1 & \text{if link}(i,j) \text{ exist and is on the (unique) path joining OD-pair}(p,q) \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

Although these constrains do not guarantee a tree topology, the avoidance of cycle is easily handled in the computer implementation. A more general formulation would allow the cost to be a nonlinear function of flow, f_{ij}, expressed in the form c_{ij} (f_{ij}). We have recently addressed the general problem where by the tree structure of the network is removed, and the direction of flow of traffic on link (i,j) is explicitly taken into account. For this problem we need to introduce some further notation:

$$\phi_{ij}^{pq} = \text{flow on directed link } (i, j) \text{ due to OD - pair } (p, q)(\text{ie, from } i \text{ to } j) \quad (8)$$

We regard the *undirected* link between nodes i and j as two *directed* links denoted by (i,j). The total link flow f_{ij}, between nodes i and j is then given by:

$$f_{ij} = \sum_{p=1}^{n} \sum_{q>p}^{n} (\phi_{ij}^{pq} + \phi_{ji}^{pq}) \quad (9)$$

The sum of flow originating and terminating at node i is given by:

$$s_i = \sum_{p=1}^{n} (F^{pi} + F^{ip}) \quad (10)$$

The total flow, u_i, traversing node i, is given by:

$$u_i = \frac{1}{2} [s_i + \sum_{j \neq i}^{n} f_{ij}] \quad (11)$$

The design problem may now be expressed as:

Minimize

$$\sum_{i=1}^{n} \sum_{j>i}^{n} c_{ij} (f_{ij}) \quad (12)$$

Subject to:

$$0 \leq f_{ij} \leq f_{ij}^{\max} \quad (13)$$

$$u_i \leq u_i^{\max} \quad (14)$$

and for all $p = 1,\ldots,n$ and $q > p$

$$\sum_{j \neq i}^{n} \phi_{ij}^{pq} - \sum_{j \neq i}^{n} \phi_{ji}^{pq} = \begin{cases} F^{pq} & \text{if i} = p; i = 1,\ldots,n \\ -F^{pq} & \text{if i} = \text{q}; i = 1,\ldots,n \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

The relations in eq. (15) specify traffic at origin and destination nodes, and also conservation of traffic at intermediate nodes. We may add further requirements, for example that there be more than one route from node p to node q. This implies redundancy in the system, and is intended to give a greater level of network reliability.

3. The use of Genetic Algorithms

GAs are a class of heuristic methods for solving optimation problem that emulate the biological processes of natural selection. An initial population is generated by some means, in which each individual is a feasible solution of the optimization problem. An individual is represented by a *chromosom*, which consists of a string of elements, and with a particular meaning. With each individual is associated of the objective function. The process then generates a sequence of populations of solutions whose members inherit characteristics from their parents.

In order to construct the next generation, pairs of parents are selected from the current population for breeding. The selection process is partly random, with some bias towards fit individuals. Child solutions inherit parental characteristics by two major processes: crossover and mutation. Crossover : the child solutions has a chromosome consisting of a portion of each of the two parents. Mutation : a random change in the child chromosome occurs.

The size of the population normally remains constant throughout the entire process. In order to achieve this some unfit individuals must be discarded. The iterative process is normally terminated after a preassigned number of generations is reached.

The details of how a tree network is constructed from the information in each chromosome for our problem is described in detail in Berry et al. [1]. Our implementation of the genetic search for an optimal tree network has been based somewhat loosely on the ideas of Goldberg [4]. It keeps some of the fittest individuals, including the best for breeding the next generation.

The compact *predecessor node format* (linear in n) was found to be an efficient representation. Many other benefits followed from this step: more efficient calculation of the objective function, easy satisfactions of the topological constraints and an ability to solve large problems with a given amount of RAM. Based on this representation, a very efficient mating algorithm was coded which enabled mutations to be implemented in a simple manner. The objective is coded as a recursive depth-first search, which also produces link information, and mutations are made independently of crossover by removing a leaf node and attaching it randomly elsewhere in the tree.

4. The Use of Linear Programming

For the more general problem described in Section 2 it is not sufficient to use a GA to construct a near-optimal tree network. We cannot merely construct a tree (which uniquely defines the traffic flow) but must allow the existence of a more general network containing alternative paths, and optimize the flow on each of several possible routes between each OD-pair.

Each solution is generated in two stages, firstly we construct the network topology using the GA approach, and secondly we optimize traffic flows on the network, taking into consideration the constraints on maximum flow, maximum degree, maximum node capacity and the need for redundancy (diversity).

To do this we formulated and solved the following LP model.

h_j^{pq} is the partial flow along the jth route from p to q.

C_j^{pq} is cost per unit flow on route j link between p and q.

F^{pq} is the flow requirement from p to q.

The LP formulation can then be written :

$$\min \sum_{p=1}^{n} \sum_{q>p} \sum_{j} C_j^{pq} h_j^{pq} \tag{16}$$

subject to

$$\sum_j h_j^{pq} = F^{pq} \quad \forall p,q \tag{17}$$

$$h_l^{pq} \geq 0 \quad \forall l,p,q \tag{18}$$

Some additional constrains may be formulated, e.q. bounds on link flow, node flow, number of link on a route, and the degree of the node. The number of decision variable on this formulation is potentially much greater than in earlier formulations. However, current industrial practice often leads to small values for both the degree of each node and the number of links in each route, thus reducing the severity of this problem.

5. Computational Experience

The GA approach described in section 3 was initially applied to the tree-network synthesis problem. Problem of size up to 35 nodes, whose cost and traffic data is described in detail in Berry et al. [1], have been satisfactorily solved in reasonable computation times and have demonstrated improved result over those obtained using other heuristic approaches. We then took the best near-optimal solution obtained for the 35-node problem and arbitrarily added extra links to add redundancy to the system. This was achieved by joining each node which had only one link associated with it to an adjacent node at random.

Initial investigations with a 10-nodes instance of the model described by (16) – (18) of section 4 have given rise ti quite manageable sub-problem if the limit on link per route is held at 3. in performing these calculation the MINOS large-scale optimization code of Murtagh and Saunders [7] was used.

We are currently investigating efficient representation of more general network in the GA for creating the network topology, and also investigating more efficient use of previous solutions in providing a warm-start to the linear programming solution process.

6. Conclusions

The approach described in this paper has been demonstrated to be efficient on a limited range of moderately sized problem. However, the success of these investigations lends hope that it will prove to provide a valuable tool for communication network design. An advantage of the approach described here over other heuristic approaches is the ability to terminate the procedure early and still be assured of a reasonably good solution which still satisfies feasibility requirement.

References

1. Berry L. T. M. Murtagh B. A, Sugden S.J, and McMahon G.B, (1995). "Application of a genetic based algorithm for optimal design of tree-structured communications networks". Proceedings of the regional Teletraffic Engineering Conference of the international Teletraffic Congress, South Africa, September 1995,pp. 361 – 370.
2. Balakrishnan A. Magnanti T. L, Shulman A. , and Wong R. T, (1991)." Models for planning capacity expansion in local access telecommunication network", Annals of Operations Research,33, pp. 239-284.
3. Gavish B, (1985). "Augmented Lagrangian based algorithms for centralised network design", IEEE Trans. On Communicaations,33,pp. 1247-1257.
4. Goldberg D. E, (1989), "Genetic Algorithms in search, Optimization and Machine Lerning, Addison Wesley , (reading)
5. Kershenbaum A. and Peng S. L. (1986). "Neighbour finding algorithm for CMST customer calculations". IEEE INFOCOM.
6. Minoux M, (1989). "Network Synthetic and optimum network design problem: models, solutionmethods and applications", network, 19, pp. 313-360.
7. Murtagh B. A. and Saunders M. A, (1987), MINOS 5.1 Users Manual, report SOL 83-R,Dept. Of Operations Research, Stanford University.
8. Sharma U, Misra K. B, and Bhattacharji A. K, (1991), "Applications of an efficient search technique for optimal design of computer communication networks". Micro Electronics & Reliability, 31,pp.337-341.