Integer Optimization Models For Communication Network Design

F. Bu’ulolo, Herman Mawengkang
Graduate School of Mathematics, University of Sumatera Utara FMIPA USU, Medan Indonesia - 20155

Abstract :- The most important part of network design problem is to find the best way to layout components in order to minimize cost while meeting a performance criterion, such as reliability and availability of communication paths between all terminals. In this paper we address an integer optimization model. We solve the model using neighbourhood search. A computational experience is presented.

Keywords :- Network Design, Integer Optimization, Communication, Reliability, Neighbourhood Search.

I. INTRODUCTION

Communication networks are built and designed in order to distribute resources. 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. The most important part of network design is to find the best way to layout the components (nodes and arcs) to minimize cost while meeting performance criterion such as transmission delay, throughput or reliability.

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. Therefore this type of problem is termed NP – complete [10].

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. Further more, communication network design problems are not time ciritical. Meaning that, most approaches which have been designed to solve these problems based on heuristics, such as, simulated annealing, taboo search, evolutionary computing [6, 11].

The network problem can also be solved using optimization techniques. The optimization techniques are mainly variations of Genetic Algorithms [3, 2, 12, 1] and Particle Swarm Optimization [4].

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. Many studies have considered topological optimization with reliability criterion [9, 1]. Further in [7] used a decomposition method based on branch and bound to minimize total network cost under a system reliability constraint. However, their method can only solve small networks because as the number of arcs increases, the number of possible layouts grow faster exponentially.

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 linear Programming (LP) and heuristic based on neighbourhood search approach to tackle this problem; and these approaches are discussed in sections 3 and 4.

II. 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 linear cost function for carrying traffic on each (possible) link \((i, j)\) between nodes \( i \) and \( j \).

Define:

\[
F_{pq} = \text{total traffic flow between OD–pairs (p,q)}
\]  (1)
If we restrict the problem to minimizing total flow in a tree network, it can be formulated as:

Minimize

\[ \sum_{p=1}^{n} \sum_{q>p} F_{pq} \sum_{i=1}^{n-1} c_{ij} y_{ij}^{pq} \]

subject to

\[ x_{ij} \in \{0,1\} \]  \hspace{1cm} (5)

\[ \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} x_{ij} = n-1 \]  \hspace{1cm} (6)

where

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

Although these constraints 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:

\[ \varphi_{ij}^{pq} = \text{flow on directed link (i, j) due to OD-pair (p, q)} \]

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:

\[ f_{ij} = \sum_{p=q}^{m} (\varphi_{ij}^{pq} + \varphi_{ji}^{pq}) \]  \hspace{1cm} (9)

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

\[ s_{i} = \sum_{p=1}^{n} (F_{pi} + F_{ip}) \]  \hspace{1cm} (10)

The total flow, \( u_{i} \), traversing node i, is given by:

\[ u_{i} = \frac{1}{2} \left[ s_{i} + \sum_{j \neq i} f_{ij} \right] \]  \hspace{1cm} (11)

The design problem may now be expressed as:

Minimize

\[ \sum_{i=1}^{n} \sum_{j \neq i} c_{ij} f_{ij} \]  \hspace{1cm} (12)

Subject to:

\[ 0 \leq f_{ij} \leq f_{ij}^{\max} \]  \hspace{1cm} (13)

\[ u_{i} \leq u_{i}^{\max} \]  \hspace{1cm} (14)

and for all \( p = 1, \ldots, n \) and \( q > p \)
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*. This implies redundancy in the system, and is intended to give a greater level of network reliability.

### III. NEIGHBOURHOOD SEARCH

It should be noted that, generally, in integer programming the reduced gradient vector, which is normally used to detect an optimality condition, is not available, even though the problems are convex. Thus we need to impose a certain condition for the local testing search procedure in order to assure that we have obtained the “best” suboptimal integer feasible solution.

Further in [5] has proposed a quantity test to replace the pricing test for optimality in the integer programming problem. The test is conducted by a search through the neighbours of a proposed feasible point to see whether a nearby point is also feasible and yields an improvement to the objective function.

Let $[\beta]_k$ be an integer point belongs to a finite set of neighbourhood $N([\beta]_k)$ We define a neighbourhood system associated with $[\beta]_k$, that is, if such an integer point satisfies the following two requirements

1. if $[\beta]_j \in N([\beta]_k)$ then $[\beta]_j \neq j \neq k$.
2. $N([\beta]_k) = [\beta]_k + N(0)$

With respect to the neighbourhood system mentioned above, the proposed integerizing strategy can be described as follows.

Given a non-integer component, $x_k$, of an optimal vector, $x_B$. The adjacent points of $x_k$, being considered are $[x_k]$ dan $[x_k] + 1$. If one of these points satisfies the constraints and yields a minimum deterioration of the optimal objective value we move to another component, if not we have integer-feasible solution.

Let $[x_k]$ be the integer feasible point which satisfies the above conditions. We could then say if $[x_k] + 1 \in N([x_k])$ implies that the point $[x_k] + 1$ is either infeasible or yields an inferior value to the objective function obtained with respect to $[x_k]$. In this case $[x_k]$ is said to be an “optimal” integer feasible solution to the integer programming problem. Obviously, in our case, a neighbourhood search is conducted through proposed feasible points such that the integer feasible solution would be at the least distance from the optimal continuous solution.

### IV. THE BASIC APPROACH

Consider a mixed integer linear programming (MILP) problem with the following form

\[
\begin{align*}
\text{Minimize} & \quad P = c^T x \\
\text{Subject to} & \quad Ax \leq b \\
& \quad x \geq 0 \\
& \quad x_j \text{ integer for some } j \in J
\end{align*}
\]

(16) (17) (18) (19)

A component of the optimal basic feasible vector $(x_B)_k$, to MILP solved as continuous can be written as

\[
(x_B)_k = \beta_k - \alpha_k \cdot \frac{1}{x_N} - \alpha_{ij} \cdot x_j - \ldots
\]

(20)

Note that, this expression can be found in the final tableau of Simplex procedure. If $(x_B)_k$ is an integer variable and we assume that $\beta_k$ is not an integer, the partitioning of $\beta_k$ into the integer and fractional components is that given

\[
\beta_k = [\beta_k] + f_k \quad , 0 \leq f_k \leq 1
\]

(21) suppose we wish to increase $(x_B)_k$ to its nearest integer, $([\beta] + 1)$. Based on the idea of suboptimal solutions we may elevate a particular nonbasic variable, say $(x_N)_j$, above its bound of zero, provided $\alpha_{ij}$, as one of the element of the vector $\alpha_j$, is negative. Let $\Delta_j$ be amount of movement of the non variable $(x_N)_j$, such that the numerical value of scalar $(x_B)_k$ is integer. Referring to Eqn. (20), $\Delta_j$ can then be expressed as

\[
\Delta_j = \frac{1 - f_k}{\alpha_{ij}}
\]

(22)
while the remaining nonbasic stay at zero. It can be seen that after substituting (21) into (22) for \((x_N)_k\) and taking into account the partitioning of \(\beta_k\) given in (21), we obtain
\[
(x_B - )_k = \lceil \beta \rceil + 1
\]
Thus, \((x_B)_k\) is now an integer.

It is now clear that a nonbasic variable plays an important role to integerize the corresponding basic variable. Therefore, the following result is necessary in order to confirm that must be a non-integer variable to work with in integerizing process.

**Theorem 1.** Suppose the MILP problem (16) – (19) has an optimal solution, then some of the nonbasic variables, \((x_N)_j, j = 1, \ldots, n\), must be non-integer variables.

**Proof:**
Solving problem as a continuous of slack variables (which are non-integer, except in the case of equality constraint). If we assume that the vector of basic variables \(x_B\) consists of all the slack variables then all integer variables would be in the nonbasic vector \(x_N\) and therefore integer valued.

**V. DERIVATION OF THE METHOD**

It is clear that the other components, \((x_B)_j, j \neq k\), of vector \(x_B\) will also be affected as the numerical value of the scalar \((x_N)_k\) increases to \(\Delta_j\). Consequently, if some element of vector \(a_j\), i.e., \(a_j^*\) for \(i \neq k\), are positive, then the corresponding element of \(x_B\) will decrease, and eventually may pass through zero. However, any component of vector \(x\) must not go below zero due to the non-negativity restriction. Therefore, a formula, called the minimum ratio test is needed in order to see what is the maximum movement of the nonbasic \((x_N)_k\) such that all components of \(x\) remain feasible. This ratio test would include two cases.

1. A basic variable \((x_B)_j, j \neq k\) decreases to zero (lower bound) first.
2. The basic variable, \((x_B)_k\) increases to an integer.

Specifically, corresponding to each of these two cases above, one would compute
\[
\theta_1 = \min \left\{ \frac{\beta_k}{v_k} \right\}_j
\]
\[
\theta_2 = \Delta_j
\]

How far one can release the nonbasic \((x_N)_k\) from its bound of zero, such that vector \(x\) remains feasible, will depend on the ratio test \(\theta^k\) given below
\[
\theta^k = \min(\theta_1, \theta_2)
\]

Obviously, if \(\theta^k = \theta_1\), one of the basic variable \((x_B)_j, j \neq k\) will hit the lower bound before \((x_B)_k\) becomes integer. If \(\theta^k = \theta_2\), the numerical value of the basic variable \((x_B)_k\) will be integer and feasibility is still maintained. Analogously, we would be able to reduce the numerical value of the basic variable \((x_B)_k\) to its closest integer \([\beta_k]\). In this case the amount of movement of a particular nonbasic variable, \((x_N)_k\), corresponding to any positive element of vector \(a_j^*\), is given by
\[
\Delta_j = \frac{f}{\alpha}
\]

In order to maintain the feasibility, the ratio test \(\theta^*\) is still needed.

Consider the movement of a particular nonbasic variable, \(\Delta\), as expressed in Eqns.(22) and (26). The only factor that one needs to calculate is the corresponding element of vector \(a\). A vector \(a_j\) can be expressed as
\[
a_j = B^T a_j, j = 1, \ldots, n - m
\]
Therefore, in order to get a particular element of vector \(a_j\) we should be able to distinguish the corresponding column of matrix \([B]^T\). Suppose we need the value of element \(a_{kj}\), letting \(v_k^T\) be the \(k\)-th column vector of \([B]^T\), we then have
\[
v_k^T = e_k^T B^{-1}
\]
Subsequently, the numerical value of \(a_{kj}\) can be obtained from
\[
a_{kj}^* = v_k^T a_j
\]
in Linear Programming (LP) terminology the operation conducted in Eqns. (28) and (29) is called the pricing operation. The vector of reduced costs \(d_j\) is used to measure the deterioration of the objective function value.
caused by releasing a nonbasic variable from its bound. Consequently, in deciding which nonbasic should be
released in the integerizing process, the vector $d_j$ must be taken into account, such that deterioration is
minimized. Recall that the minimum continuous solution provides a lower bound to any integer-feasible
solution. Nevertheless, the amount of movement of particular nonbasic variable as given in Eqns. (22) or (26),
depends in some way on the corresponding element of vector $\alpha_j$. Therefore it can be observed that the
deterioration of the objective function value due to releasing a nonbasic variable $(x_N)_j^*$ so as to integerize a basic
variable $(x_B)_k$ may be measured by the ratio

$$\frac{d_k}{\alpha_j}$$

where $|z|$ means the absolute value of scalar $a$.

In order to minimize the deterioration of the optimal continuous solution we then use the following strategy for
deciding which nonbasic variable may be increased from its bound of zero, that is,

$$\min_{j} \left| \frac{a_k}{\alpha_j} \right|, \quad j = 1, \ldots, n-m$$

From the “active constraint” strategy and the partitioning of the constraints corresponding to basic ($B$),
superbasic ($S$) and nonbasic ($N$) variables, we can write [5].

$$\begin{bmatrix} B & S & N \end{bmatrix} \begin{bmatrix} x_B \\ x_S \\ x_N \end{bmatrix} = \begin{bmatrix} b \\ 0 \\ 0 \end{bmatrix}$$

or

$$Bx_B + Sx_N = b \tag{33}$$

$$x_N = b_N \tag{34}$$

The basis matrix $B$ is assumed to be square and nonsingular, we get

$$x_B = \beta - Wx_S - \alpha x_N \tag{35}$$

Where

$$\beta = B^T b \tag{36}$$

$$W = B^T S \tag{37}$$

$$\alpha = B^T N \tag{38}$$

Expression (34) indicates that the nonbasic variables are being held equal to their bound. It is evident
through the “nearly” basic expression of Eqn. (35), the integerizing strategy discussed in the previous section,
designed for MILP problem can be implemented. Particularly, we would be able to release a nonbasic variable
from its bound, Eqn.(34) and exchange it with a corresponding basic variable in the integerizing process,
although the solution would be degenerate. Furthermore, the Theorem (1) above can also be extended for
MINLP problem.

**Theorem 2.** Suppose the MINLP problem has a bounded optimal continuous solution, then we can always get a
non-integer $y_j$ in the optimum basic variable vector.

**Proof.**
1. If these variables are nonbasic, they will be at their bound. Therefore they have integer value.
2. If a $y_j$ is superbasic, it is possible to make $y_j$ basic and bring in a nonbasic at its bound to replace it in the
superbasic.

However, the ratio test expressed in (25) cannot be used as a tool to guarantee that the integer solution found
still remains in the feasible region.

### 5.1 Pivoting

Currently, we are in a position where particular basic variable, $(x_B)_k$ is being integerized, thereby a
_corresponding nonbasic variable, $(c_N)_j$, is being released from its bound of zero. Suppose the maximum
movement of $(x_N)_j^*$ satisfies

$$\theta^* = \Delta_j^*$$
such that \((x_B)_k\) is integer valued. To exploit the manner of changing the basis in linear programming, we would be able to move \((x_B)_k\) into \(B\) (to replace \((x_B)_k\)) and integer-valued \((x_B)_k\) into \(S\) in order to maintain the integer solution. We now have a degenerate solution since a basic variable is at its bound. The integerizing process continues with a new set of \([B, S]\). In this case, eventually we may end up with all of the integer variables being superbasic.

**Theorem 3.** A suboptimal solution exists to the MILP and MINLP problem in which all of the integer variables are superbasic.

**Proof.**
1. If all of the integer variables are in \(N\), then they will be at bound.
2. If an integer variable is basic it is possible to either
   - interchange it with a superbasic continuous variable, or
   - make this integer variable superbasic and bring in a nonbasic at its bound to replace it in the basis which gives a degenerate solution.

The other case which can happen is that a different basic variable \((x_B)_j \neq k\) may hit its bound before \((x_B)_k\) becomes integer. Or in other words, we are in a situation where

\[ \Theta^* = \Delta_1 \]

In this case we move the basic variable \((x_B)_j\) into \(N\) and its position in the basic variable vector would be replaced by nonbasic \((x_B)_j^*\). Note that \((x_B)_k\) is still a non-integer basic variable with a new value.

VI. THE ALGORITHM

After solving the relaxed problem, the procedure for searching a suboptimal but integer-feasible solution from an optimal continuous solution can be described as follows.

Let

\[ x = [x] + f, \quad 0 \leq f \leq 1 \]

be the (continuous) solution of the relaxed problem, \([x]\) is the integer component of non-integer variable \(x\) and \(f\) is the fractional component.

**Stage 1.**

Step 1. Get row \(i^*\) the smallest integer infeasibility, such that 
\[ \delta^*_i = \min \{ f_i, 1 - f_i \} \]

Step 2. Do a pricing operation
\[ v^*_j = e^T B^+ \]

Step 3. Calculate \(\sigma_j = v^T \alpha_i j\)

With \(j\) corresponds to
\[ \min \left| \begin{array}{c} d_j \\ \sigma_j \end{array} \right| \]

Calculate the maximum movement of nonbasic \(j\) at lower bound and upper bound.

Otherwise go to next non-integer nonbasic or superbasic \(j\) (if available). Eventually the column \(j^*\) is to be increased form LB or decreased from UB. If none go to next \(i^*\).

Step 4. Solve \(B\alpha_j^* = \alpha_j^*\) for \(\alpha_j^*\)

Step 5. Do ratio test for the basic variables in order to stay feasible due to the releasing of nonbasic \(j^*\) from its bounds.

Step 6. Exchange basis

Step 7. If row \(i^* = \emptyset\) go to Stage 2, otherwise

Repeat from step 1.

**Stage 2.** Do integer lines search to improve the integer feasible solution

VII. COMPUTATIONAL EXPERIENCE

The algorithm described in Section 6 applied to the tree-network synthesis problem. Problem of size up to 35 nodes, whose cost and traffic data is described in detail in [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.

We are currently investigating efficient representation of more general network 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.

VIII. 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