Practical approach to the optimization of large-scale switching networks

V M Grout*, P W Sanders* and C T Stockel† detail the optimization of large-scale switching networks

There are a number of techniques that enable networks to be optimized. Neither search techniques nor heuristic methods are entirely satisfactory in solving practical problems. The authors describe an approach to network optimization that utilizes an algorithm that reduces the number of nodes and consequently the time taken to find a solution. The approach is based on two basic stages: a preparation stage and an optimization stage.

Keywords: communication networks, network optimization, large scale switching networks

In recent years there has been a dramatic increase in the amount of data being transmitted and with the appearance of many new services, such as electronic mail, viewdata and facsimile, it seems that the days of pencil and paper as the primary means of communication are numbered. This demand for communication has led to a similar increase in the number and size of both public and private networks. From the grand scale of the public telephone network, through the variety of private communication systems used by banks, shops etc., to the small system employed by a company in a single building, there is the common need for efficient terminal interconnection.

As the amount of data flowing in various guises increases, the size of new and existing networks increases. Not only is it necessary to increase the traffic carrying capacities of the main routes in such networks but the number and distribution of terminal and switch locations grows and changes. Modem electronic communication may be very flexible and efficient but consequently can be rather expensive. These costs are becoming an increasingly important part of company financial matters with growing amounts of money being put aside for such purposes, so the requirement that any network be connected in the most economical way is of extreme importance. The problem for the network planner is to determine the 'best' way of connecting the terminals together where 'best' is usually taken to mean 'cheapest' with consideration given to constraints on performance under possible failures and heavy traffic conditions.

There already exist a number of methods that may be used to achieve network optimization. Some find the exact optimum by search techniques,[1-3] while others use heuristic methods in order to arrive at an approximation of the optimum in a shorter time[4-8]. None of these is entirely satisfactory in solving the real problem in practice. Many of them take excessive amounts of processing time and/or storage resources, while others make such drastic assumptions and simplifications that the end result bears very little relevance to the practical requirements of the problem. There is clearly a need for a new approach to the problem. A method is required which runs within acceptable limits of time on a smaller machine and operates on realistic networks to give results that are within a specified amount of the true optimum.

The work presented here was initially carried out with regard to telephone networks but this distinction is of no real importance. The principles discussed will apply

*Department of Communication Engineering, Plymouth Polytechnic, Plymouth, Devon PL4 8AA, UK
†Department of Computing, Plymouth Polytechnic, Plymouth, Devon PL4 8AA, UK

equally to any problem that requires that a number of data source/sinks be linked together to form a hierarchical system in such a way as to minimize the total cost. The development of the techniques presented in this paper was achieved using an IBM PC-XT personal computer. The same machine was also used for extensive numerical testing of these techniques.

The results of this work will eventually form part of a suite of programs to be used by British Telecom CFM/NC (Communications and Facilities Management/Network Consultancy).

DESCRIPTION OF THE PROBLEM

Although there is a great variety of network types in computer and telephone systems, they can all be expressed in a common uniform description as follows.

A network N consists of n nodes, where each node i ($i = 1, 2, \ldots, n$) is uniquely defined by its coordinates in the x-y plane. For the general nationwide case, these could be Ordinance Survey grid references. In addition to the nodes there must be at least one $n \times n$ matrix which describes the amount of data (traffic) flowing between each pair of nodes in N. In other words:

$\mathbf{A} = (a_{ij})$ where a_{ij} is the amount of traffic that leaves i and is destined for j.

There may be more than one traffic matrix if there is more than one type of traffic produced in the system (voice, data etc.), but for ease of explanation it shall be assumed that there is only one. The following discussion extends without difficulty to the case where there are two or more matrices. The combination of node locations and traffic matrix completely describes the network N.

An examination of this information will yield further details of the particular network in question. There may be large or small amounts of traffic involved, the nodes may be distributed uniformly or clustered into groups, but for all networks the problem remains the same; namely 'connect up the n nodes in such a way as to minimize the total cost subject to performance constraints'. These constraints will be of reliability, security and grade of service (the percentage of data which fails to transmit). A typical network problem and solution is shown in Figure 1.

Two types of links are currently used in practice. Single analogue circuits suitable for carrying relatively small amounts of speech/data and wideband digital links (equivalent to a number of analogue links) applicable for the higher levels of the network. The nodes shown as squares are transits and are in the best places to switch traffic on the digital links between them. A typical structure would be n network nodes with M of them chosen as transits. Each network node is connected directly to only one of the transit nodes (if a node is a transit then we think of it as being connected to itself) to form M stars. The higher level (intertransit) links are determined separately to achieve the cheapest solution. For two ordinary network nodes to communicate, the traffic must be routed via the path indicated in Figure 1. These intertransit links are usually digital, whereas the

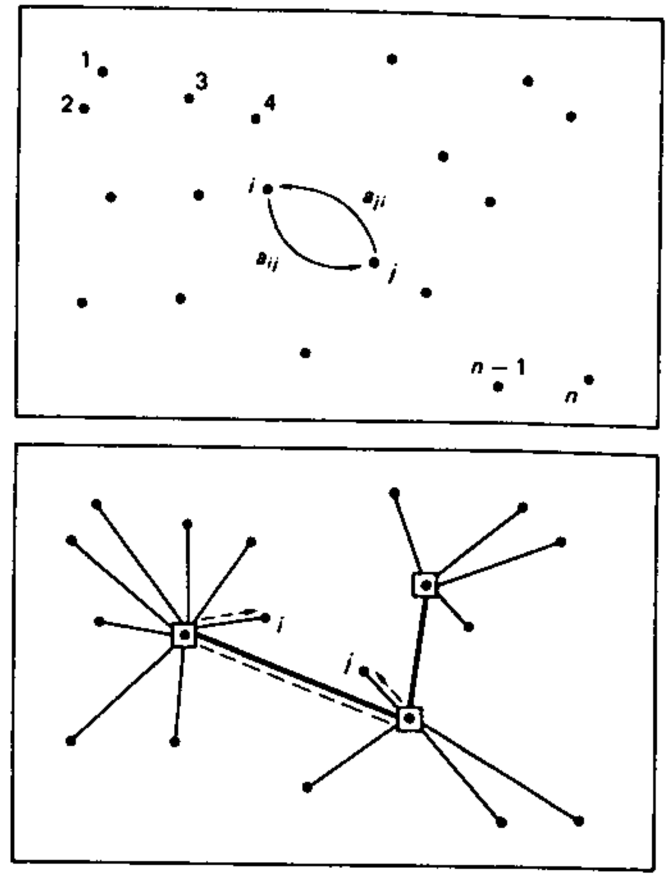

Figure 1. Typical network problem and solution pair

node–transit links are mostly analogue at present. The part of the network comprising the transits and the digital links connecting them is conveniently known as the core network, and the nodes and analogue links are generally known as the supply or service network.

In order to determine the cheapest solution it is necessary to assign costs to the individual components of a network. There are basically three component costs: cost of switches (transits), cost of analogue links and cost of digital links. Therefore, in general, the cost of a network N is given by:

$$C_N = \sum_{\substack{\text{all transits}\\ \text{in } N}} C_T + \sum_{\substack{\text{all analogue}\\ \text{links in } N}} C_A + \sum_{\substack{\text{all digital}\\ \text{links in } N}} C_D$$

where C_T, C_A, C_D are the costs of individual transits, analogue links and digital links respectively.

The cost of a switch can be considered as comprising a fixed installation charge and an additional incremental cost per port, where a port is equivalent to one input/output line entering or leaving the switch. The cost of a single (digital or analogue) line comprises a fixed connection charge plus a cost per unit distance.

$$C_T = a_T + b_{TA}\,p_A + b_{TD}\,p_D$$

where a_T is a standing charge; p_A and p_D are the number of analogue and digital input/output lines entering or leaving the transit switch; b_{TA} and b_{TD} are the costs per analogue and digital input/output port at the transit switch.

$$C_A = a_A + b_A d$$

where a_A is a standing charge; b_A the cost per unit distance and d is the length.

$$C_D = a_D + b_D d$$

where a_D is a standing charge, b_D the cost per unit distance and d is the length.

In general, the optimum solution for a network will have a similar appearance to Figure 1. The use of a partial mesh for the top levels and a set of stars for the low level is particularly suited for communication networks and we make use of this in our optimization. The two level (node-transit) situation is also commonly accepted as opposed to (say) a three level (node-transit-'super transit') configuration.

EXISTING METHODS

In order to determine the optimum arrangement of a network of the form described above, the most obvious method is simply to generate each possible network in turn, determine the total cost of that particular network and compare costs in order to find the cheapest. This, however, is not practical for large arrangements. If there are n nodes in the network and the optimum has M^* transits, we have to let M, the number of transits being 'tried out', range from 1 to n in order to determine the value of M^*. Within each of these loops it is necessary to consider every one of the

$$^nC_M = \frac{n!}{M!(n-M)!}$$

combinations of M transits from n nodes. Also every possible set of intertransit connections must be considered for each combination. This number is equivalent to the total number of graphs that can be constructed on M points, with the exception that half of them will be disconnected and thus give rise to an unacceptable solution. Taking this into account, the number of different sets of intertransit links (or alternatively, the number of possible core networks) for each combination of M transits is:

$$\frac{2^{(M(M-1)/2)}}{2} = 2^{(M(M-1)/2)-1}$$

Consequently, the total number of networks that must be examined in this form of exhaustive search is:

$$\sum_{M=1}^{n} \left[\frac{n!\ 2^{(M(M-1)/2)-1}}{M!\,(n-M)!} \right]$$

Since the summand is exponential in M, the value of the above expression becomes enormous as M increases. However, this expression takes into account only the physical topology of the network; there is no consideration given to how traffic will be routed over the links. If there is no direct link between two transits, which is the best path to take through the network?

The appreciation of the need for simplification has produced a wide variety of network optimization methods. Some of these use established mathematical techniques[2,5,8], while others have developed new algorithms for the purpose[1,4,7,9]. One of the commonest approaches is to formulate the network problem as a linear programming task and then use methods such as the simplex method to obtain a solution. In order to do this it is necessary initially to define certain values within the network. The usual formulation is as follows:

Let $y(i)$ $(i = 1, 2, \ldots, n)$ be a variable taking the value 1 if a transit is set at node i and 0 otherwise.

Let $x(i, j)$ $(i, j = 1, 2, \ldots, n)$ be a variable taking the value 1 if there is a link between i and j, and 0 otherwise.

Let $CT(i)$ be the cost of a transit at node i.

Let $CL(i, j)$ be the cost of a link between i and j.

The problem can then be defined as:

$$\text{Minimize } Z = \sum_{i=1}^{n} CT(i)y(i) + \sum_{i,j=1}^{n} CL(i,j)x(i,j)$$

subject to $x(i, j), y(i) >= 0 \qquad i, j = 1, 2, \ldots, n$

$$\text{and} \sum_{i=1}^{n} y(i) >= 1 \qquad \sum_{i=1}^{n} x(i,j) >= 1 \qquad j = 1, 2, \ldots, n$$

which can be solved using an established linear (integer) programming method.

This seems quite satisfactory on the surface but a closer inspection shows that assumptions have been made with the original problem in order to obtain it in this convenient form. The 'constants' $CT(i)$ and $CL(i, j)$ cannot be defined in this way. Referring back to the cost formulas given in the previous section, it can be seen that they are not constants at all. The cost of a transit at site i depends on its size, which in turn depends on where the other transits are in the network and how many there are of them. Similarly it is impossible to state the cost of the link joining i to j since this will depend on how much traffic it has to carry which in turn depends on the final topology of the network. This sort of information is unavailable at the beginning of the optimization process and so cannot be provided as input to a linear programming package. Unfortunately, then, this need for reality reduces the applicability of a large section of the work carried out to date on network optimization.

It is not only the mathematical programming approaches that fall into this trap. Many of the more constructional methods which deal in nodes and links rather than constants and variables also make impractical assumptions. With the increased complication that variable costs cause, the amount of work that needs to be done increases dramatically. The numbers involved are not as large as for the case of the exhaustive search but they are large, certainly increasing as an exponential function of n. Most existing methods run on medium sized or large machines and very few can deal with more than a few hundred nodes even then.

NEW APPROACH

The main problem with the optimization is the number of nodes in the network. The time taken (as well as resources used in general) to produce a solution is a function of that number, $T = f(n)$. The difficulty is that this time becomes too large as n increases. The basic objective, then, in seeking a solution is to decrease the value of T which can be done in two ways:

- Transform the function f to a (less severe) function g. Then the new time T' will be given by $T' = g(n)$ and $T' << T$.
- Transform the number n to a (smaller) number q. Then the new time T' will be given by $T' = f(q)$ and again $T' << T$.

Replacing the function f by g means simplifying the optimization process which in turn means making assumptions and approximations in finding a solution. In contrast, the second possibility requires that work is done on a reduced number of nodes with the same optimization process as before. While literature on network optimization contains many algorithms of the first type, very little has been done with respect to the second. It is our belief that it may be in this direction that a more acceptable solution lies. If somehow the number of nodes under consideration can be reduced before optimization begins in earnest then the corresponding value of T will have been reduced as well. A method is now presented which consists of two basic stages: a preparation stage and an optimization stage. (See Figure 2.)

Having obtained a reduced number of nodes (by whichever method is appropriate) optimization proceeds on the new smaller network approached in three stages, making four in all including the reduction. First, a quick approximation method is used in order to find a reasonably good starting set of transits, then a sequence of perturbations and adjustments on these transit positions is made and finally, when no further improvements are to be found and the final set of transits is established, the best set of intertransit connections (the optimum core network) is found.

The merit of this modular structure is that only as much work as is necessary is carried out at each stage before

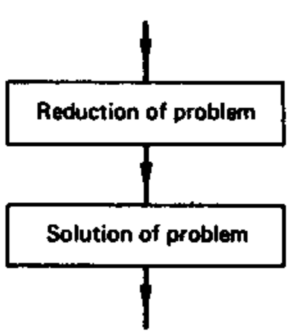

Figure 2. Presentation of a method as two basic stages

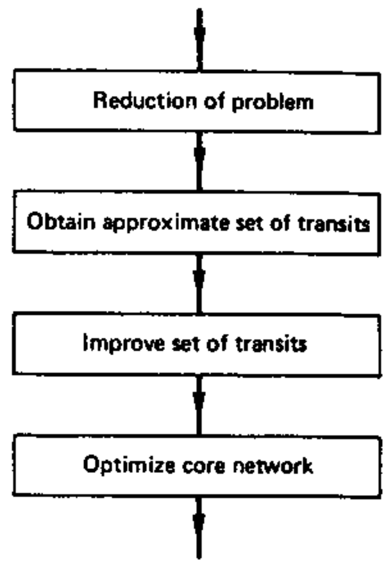

Figure 3. Modular structure of complete method

moving on to the next, but each stage is a little more rigorous and consequently more accurate than the previous one. This method is advantageous compared with existing procedures as it does not try to do everything all at once. The n original nodes are replaced by q representative ones, an initial set of transits is obtained very quickly, this set is modified as far as possible, and finally, with the transits established, the links between them are optimized.

Stage one — reduction

Consider a network with n nodes and an $n \times n$ traffic matrix to describe the traffic flow about the network. The reduction stage is used to determine the position of $q(q < n)$ nodes which represent the original network. In practical terms, q might be 25 or 50 whereas n could be over 1000. This reduction takes place one stage at a time as follows.

By considering all the n nodes in the network, select the two that are closest together. These nodes are then replaced by a single node which represents them, in terms of both their position and traffic characteristics (see Figure 4). The position of this equivalent node is found on the line joining the two points it has replaced, but weighted toward the node with the greater amount of traffic entering and leaving it. Symbolically, if a node r replaces nodes i and j then the x and y coordinates of r are given by:

$$x(r) = \frac{W(i)x(i) + W(j)x(j)}{W(i) + W(j)} \quad y(r) = \frac{W(i)y(i) + W(j)y(j)}{W(i) + W(j)}$$

where $W(i)$ and $W(j)$ are weighting factors indicating the amount of traffic there is at i and j respectively, determined at the beginning of the process. The weighting factor of the new point r, $W(r)$, is calculated as:

$$W(r) = W(i) + W(j)$$

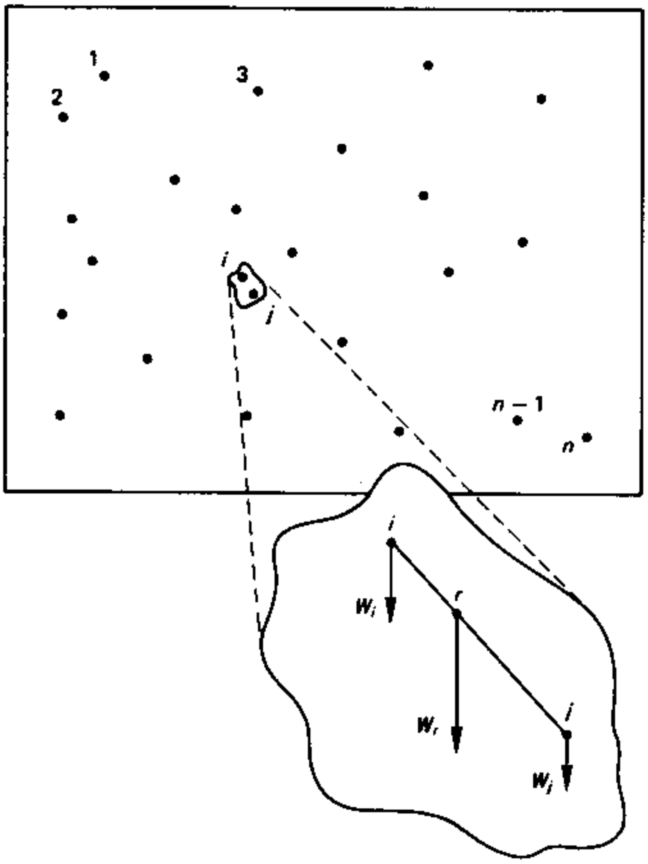

Figure 4. Single stage of reduction

The traffic matrix is updated and reduced by combining the original rows and columns for i and j to give a single row and column for r (see Figure 5).

After this reduction stage we have a network with $n - 1$ nodes and a traffic matrix of size $n - 1 \times n - 1$. The process is then repeated to give an $n - 2$ node network, an $n - 3$ node network and so on. By repeating this $n - q$ times, the required q node network is obtained. These q nodes are located and traffic loaded such that they form a contracted yet accurate picture of the original network. It is important to realize that these are not actual nodes but a representative set on which the optimization may begin.

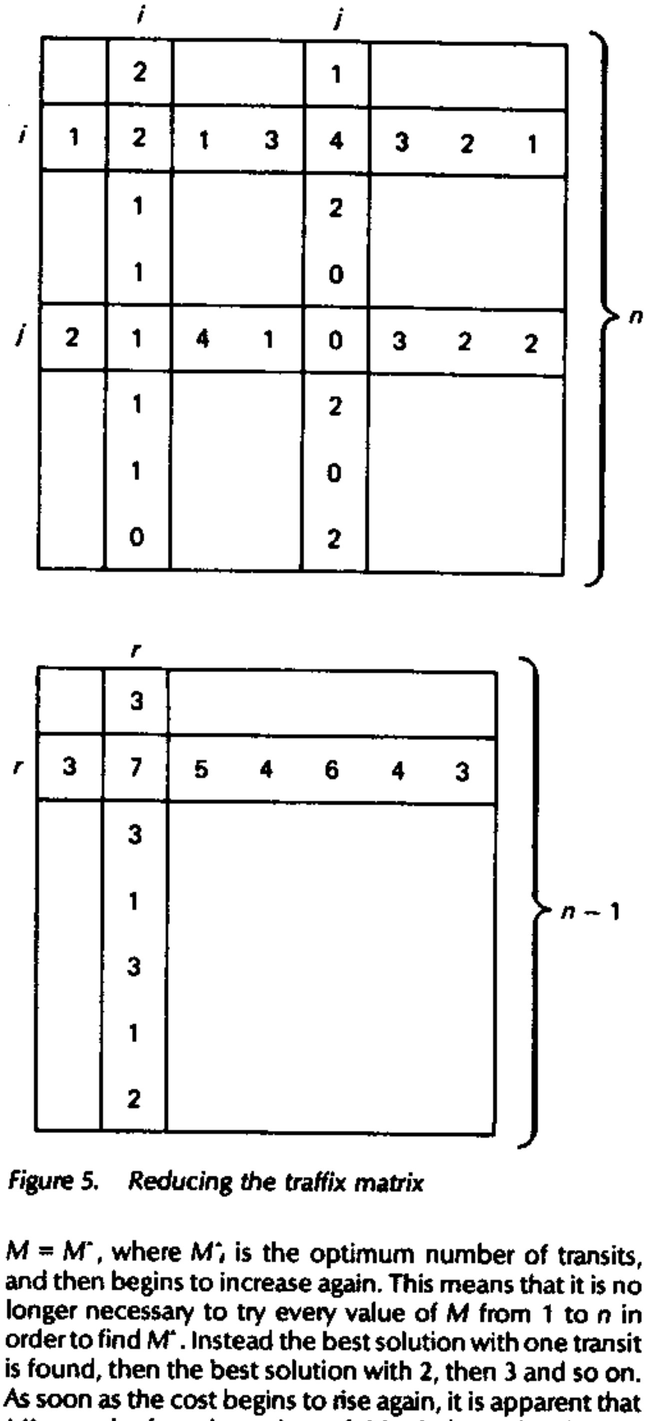

Figure 5. Reducing the traffix matrix

Stage two — star optimization

We will now find a first approximation to the optimum number of transits and their locations. First, consider what happens when the number of transits is increased. The switch costs obviously increase steadily, the cost of analogue links in the supply network decreases (sharply at first then gradually since the saving in analogue links caused by employing two transits rather than one transit is greater than the saving in analogue links caused by employing five instead of four, for example) and the digital links within the core network increase since there are more transits to interconnect. The result is the curve shown in Figure 6. The overall cost reaches a minimum at $M = M^*$, where M^* is the optimum number of transits, and then begins to increase again. This means that it is no longer necessary to try every value of M from 1 to n in order to find M^*. Instead the best solution with one transit is found, then the best solution with 2, then 3 and so on. As soon as the cost begins to rise again, it is apparent that M^* was in fact the value of M tried previously and therefore it is unnecessary to continue.

For each particular value of M, it is necessary to examine every combination of M transits from q nodes. These sets must be generated in order and tested one at a time. For each set of M transits there may be a very large number of possible core networks — far too many to be examined at this stage, so the assumption that the M transits are to be connected in star formation is made, with one at the centre and $M - 1$ connected to it as shown in Figure 7. This cuts down the number of core networks to be studied from an unmanageable number to just M each time since it must be determined which

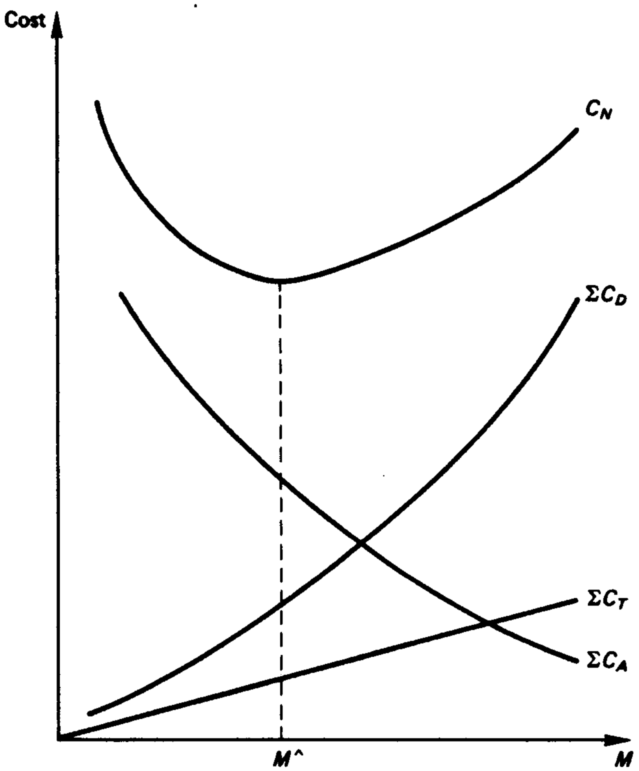

Figure 6. Variation of costs with respect to the number of transits

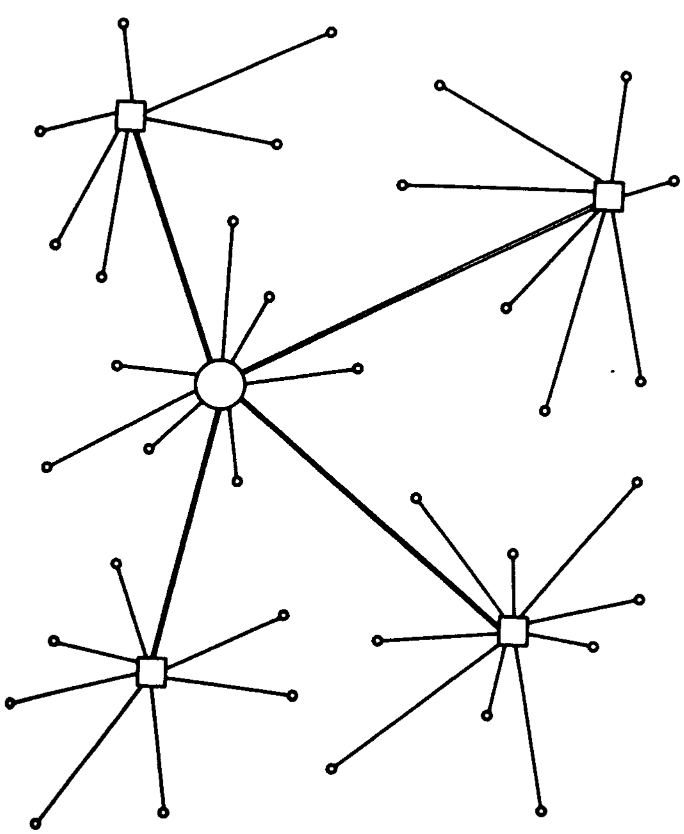

Figure 7. Star configuration core network

transit is best at the centre of the star for that set. This allows the production of a quick starting set of transits without having to search through large numbers of networks. This approximate solution is then passed on to the next section to be improved. The justification for this approach lies in the tendency for real network solutions to have star topologies within the core network or an arrangement that is very similar.

Stage three — transit perturbation

A first attempt at an optimum set of transits now exists which will be a good approximation in some cases but not so in others. The set was obtained with a specific way of connecting the transits together, so by introducing an awareness of different core network possibilities into the proceedings, an attempt to improve upon this for the network in question is made. Each transit in turn is taken from its position and moved gradually around the other nodes in the network. At the same time, different forms of interconnection are tried. If, at any stage, an improvement is encountered, then it is recorded. Having tried out all the different positions for the first node and a number of different connections as well, the choice that gave the greatest improvement is selected and the network is updated accordingly. This is now a (possibly) different set of transits with a (probably) different set of connections from the original star. The entire procedure is now repeated with the second transit, then the third and so on. Having completed the procedure for the Mth transit, the first transit is selected again and the complete cycle repeated. All of the time, continuously improved sets of transits are obtained by moving further and further away from the initial star.

Eventually, however, a stage will be reached when no further improvement can be found. As soon as one complete cycle of transits is observed without finding any improvement, the procedure is finished. The new set of transits is accepted and passed to the final section to have the links between them optimized.

Stage four — core network optimization

The number and position of the M transits are now established. It only remains necessary to determine the most efficient method of connecting the transits together (optimizing the core network). Two methods are presented, one being an extension of the other. The first is very simple and fast but not always as accurate as may be required, while the other is more complicated but can be made as accurate as necessary.

The method begins with all M transits connected to one another (a full mesh of $M(M - 1)/2$ digital links) and the removal of each link is considered in turn. The removal of a link will result in a saving of cost, but the traffic that was flowing along that link now has to be sent along a different path. This may require additional links in other parts of the network or an increase in the fill of existing digital circuits. Whichever the case, removing the link and redirecting the traffic by the cheapest alternative route is tried and the difference in cost is recorded. This is repeated for all digital links in the core network and the link whose removal caused the greatest improvement is removed permanently from the network, with the traffic being rerouted in the best way possible. This single stage procedure is repeated until either:

- there are no further links whose removal gives an improvement, or
- the removal of any further link disconnects the network.

The core network, and therefore the complete network, is now optimized.

The simple method described is quite sufficient in many cases. However, it can lose accuracy because it makes decisions on a relatively small amount of information at each stage, namely the improvement in cost caused by the removal of a single link. It is possible that a link could be removed early in the process which would turn out to be of great importance toward the end of this process. This, unfortunately, is the bane of all heuristic procedures in one form or another.

Matters can be improved considerably by a 'look-ahead' process in the core network optimization section. In this, it is necessary to consider the removal not just of one link at a time, but of a series of links. Every possible sequence of link removals up to a maximum of S stages ahead is examined, and the best sequence, on the basis of the greatest improvement in cost after S removals, is selected. However, not all of those links are removed, only the first one in the sequence. The process then repeats from this new core network with stopping criteria exactly as for the simple method. This is very similar to the way in which a chess playing program works. The machine 'looks ahead' as far as possible at each stage, finds the most advantageous board position after a certain number of projected moves, then plays the move that could lead to this position. There is no guarantee that this position will be reached because better information will become available as the game progresses and a different course could be taken.

RESULTS

In order to examine the model in detail, an exhaustive search or full optimization program was developed to run with the same network examples, but on a large Prime 9950 computer. This is a fast, powerful machine in comparison with the IBM PC-XT and is capable of making huge numbers of calculations in a short time. Clearly, such a machine is needed to run routines of this complexity. Even so, exhaustive search methods must be treated with care. Although very slow and requiring cpu times in the order of days and weeks, it produces exact reference networks by trying out every possible network and recording the cheapest. It also gives a measure of the complexity of the job in hand because the program becomes impossible to run for n greater than about 30 or 40 depending on the particular characteristics of the network concerned. Comparisons of results between the model and this full optimization program have been very reasonable indeed. Of the 20 or so networks tested simultaneously so far, the overall optimum (number as well as location of transits) was obtained in all but one case (see Network 3 in Table 1). This was an artificial network deliberately constructed in a totally symmetric form with completely uniform traffic flow. Its purpose was to test the model to the extremes of its operation. Even in this instance the difference in cost between the modelled solution and the correct one was less than 1%. If the number of transits was constrained to be artificially higher or lower than the optimum value of M^*, for the purposes of determining the optimum network with a specific number of transits, then the results obtained were exactly correct around 80% of the time with the average error in

Table 1. Six typical network comparisons between the Prime and the IBM PC-XT

Network no	No of nodes	No of transits	Cost of 'optimum' solution produced by: Prime	IBM PC-XT	Error
			(units of cost)		(%)
1	19	1	1.997	1.997	0.0
1	19	2	1.603	1.603	0.0
1	19	3	1.591	1.591	0.0
1	19	4	1.590*	1.590*	0.0
1	19	5	1.668	1.670	0.1
2	50	1	1.532	1.532	0.0
2	50	2	1.219	1.219	0.0
2	50	3	1.199	1.199	0.0
2	50	4	1.190*	1.190*	0.0
2	50	5	1.203	1.203	0.0
2	50	6	1.256	1.256	0.0
3	49	1	2.340	2.340	0.0
3	49	2	2.254	2.254	0.0
3	49	3	2.152	2.152	0.0
3	49	4	2.093	2.093*	0.0
3	49	5	2.079*	2.095	0.8
3	49	6	2.146	2.155	0.4
4	18	1	1.512	1.512	0.0
4	18	2	1.484*	1.484*	0.0
4	18	3	1.509	1.509	0.0
4	18	4	1.562	1.562	0.0
5	25	1	0.965	0.965	0.0
5	25	2	0.952*	0.952*	0.0
5	25	3	1.001	1.001	0.0
5	25	4	1.090	1.122	2.9
6	21	1	1.935	1.935	0.0
6	21	2	1.895	1.895	0.0
6	21	3	1.891*	1.891*	0.0
6	21	4	1.899	1.908	0.4

where* indicates the overall 'optimum' produced by each machine for each of the six networks the remainder of cases being less then 1% and bounded by 5%. These are considered to be excellent results for a program running so much faster than existing methods. It appears that it is only when particularly suboptimal numbers of transits are requested by the user that the model begins to lose accuracy and, even then, to within perfectly acceptable limits. Table 1 gives the results of six typical network comparisons between the Prime and the IBM PC-XT.

The precise time taken by the model to obtain a solution is somewhat flexible. The time taken for the reduction process will depend on the values of n and q but in general it does not constitute a major part of the running time. The speed of the star optimization section will depend on the value of q; the greater the reduction, the smaller the value of q and the quicker the star optimization becomes. The transit perturbation section will run only until there is no more improvement to be found in transit selection and this in turn will depend on the particular network under consideration. The run-time of the core network optimization will be determined by the number of 'look-ahead' stages. Figure 8 shows an example of the saving in time achieved by the reduction process and the assumption of a star core-network at the early stages. It shows graphically the number of different core networks that need to be considered, first by the full optimization method and then by the reduced version. A typical 100 node network with an optimum of

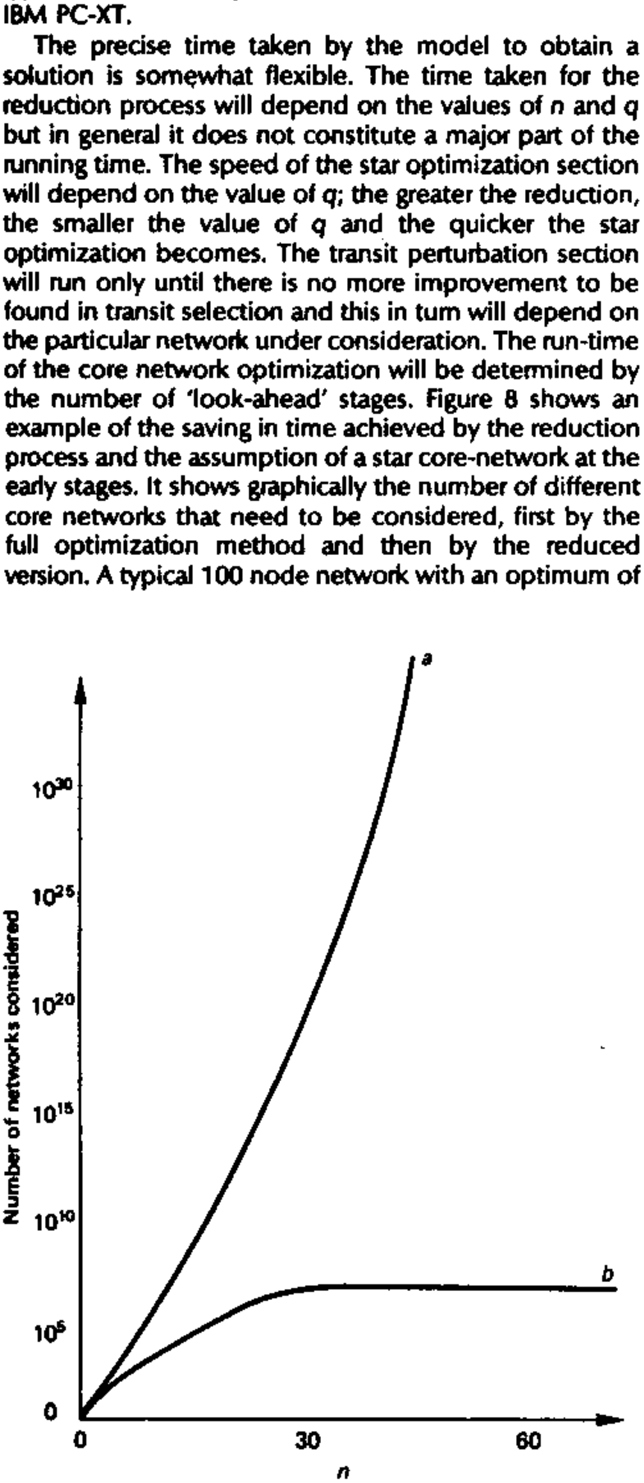

Figure 8. Comparison of computational complexities; $q = 30$ and $M^* = 4$; (a): exhaustive search; (b): model

four transits takes the following approximate times for each section:

Reduction — 10 min to $q = 30$ nodes,
Star optimization — 60 min to $M = 4$ transits,
Transit perturbation — (on average) 120 min to new $M = 4$ transits,
Core optimization — one stage 'look-ahead' — a few seconds,
two stage 'look-ahead' — 5 min,
three stage 'look-ahead' — 60 min,
etc.

These timings are the approximate average values for each section as measured using an IBM PC-XT personal computer which is a comparatively slow machine. If the program is run on the Prime, which is about 20 times faster, it can be seen that in about 10 min it is possible to obtain an approximation to a solution that was otherwise impossible.

CONCLUSIONS

This is a new and fundamentally different method of optimizing the structure of large switching networks. The results obtained to date are extremely encouraging and have apparently achieved the combined objectives of decreasing the computational work while maintaining the required level of accuracy. It soon becomes impossible to test the model for increasing values of n since full optimization programs take too long to run even on the fastest of machines. Work is proceeding at present to determine the limitations of this technique from a theoretical point of view, and, because no assumptions have been made based solely on communication networks, the application of this technique to many types of network problem seems possible.

ACKNOWLEDGEMENT

The authors gratefully acknowledge British Telecom for their sponsorship of this work and their kind permission to publish this paper.

REFERENCES

1 **Chandy, K M and Russel, R A** 'The design of multipoint linkages in a teleprocessing tree network' *IEEE Trans. on Comput.* Vol C-21 No 10 (October 1972) pp 1062–1066
2 **Matsui, S** *A network optimization by mixed integer programming* Nippon Univac, Tokyo, Japan (1978)
3 **Tabourier, Y** 'All shortest distances in a graph: An improvement to Dantzig's inductive algorithm' *Discrete Math.* Vol 4 (1973) pp 83–87
4 **Karnaugh, M** 'A new class of algorithms for multipoint network optimization' *IEEE Trans. on Commun.* Vol COM-24 No 5 (May 1976) pp 500–505

5 **Hoang, H H** 'Topological optimization of networks: a nonlinear mixed integer model employing generalized benders decomposition' *IEEE Trans. on Auto. Control* Vol AC-24 No 1 (February 1982) pp 164–169
6 **Bahl, L R and Tang, D T** 'Optimization of concentrator locations in teleprocessing networks' *Symp. on Comput. Commun. Networks and Teletraffic* Polytechnic Institute of Brooklyn, New York, USA (4–6 April 1972)
7 **Greenhop, D and Campbell, R** 'Cost 201 — a procedure for the optimization of telecommunication networks' *Br. Telecomm. Eng.* Vol 3 (April 1984) pp 47–58
8 **Mirzaian, A** 'Lagrangian relaxation for the star-star concentrator location problem: approximation algorithm and bounds' *Networks* Vol 15 (1985) pp 1–20
9 **Elias, D and Ferguson, M J** 'Topological design of multipoint teleprocessing networks' *IEEE Trans. on Commun.* Vol COM-22 No 11 (November 1974) pp 1753–1762