ROUTING IN A NETWORK WITH UNRELIABLE COMPONENTS

by

Irina Neuman
Information Systems Area
New York University
624 Tisch Hall
New York, NY 10003

and

Bezalel Gavish
Dept. of Administrative Sciences
Naval Postgraduate School
Monterey, California

April 1988

Center for Research on Information Systems
Information Systems Area
Graduate School of Business Administration
New York University

Working Paper Series

CRIS #181
GBA #88-34

Abstract

A new approach to the joint selection of primary and secondary routes in a network with unreliable components is presented. The mathematical model captures the changes in the operational characteristics of the network when it adapts to failures. Lagrangean relaxation and subgradient optimization techniques are used to obtain good heuristic solutions to the problem, as well as lower bounds to be used as benchmarks against which the quality of the solution is assessed. Results of numerical experiments are reported, and directions for further enhancements of the model are discussed.

1 Introduction

The accepted standard architecture for large computer communication systems is a hierarchical structure, consisting of a backbone network and a number of local access networks. The traffic originating in a given user site is collected by the local access network and passed to the communication processor that represents its entry point to the backbone. The backbone network is responsible for carrying the traffic to the appropriate destination switches on its boundary. The traffic is next forwarded to the corresponding local access networks, responsible for delivery to the final destination. Local access networks are usually a combination of tree structures and local loops, while the backbones are mesh-like topologies, with high capacity links and sophisticated switches. Such an architecture suggests a 'divide and conquer' two step approach to the design of complex computer networks, namely to separate between the very intricate problems of global significance that arise in the design of the backbone network, and the better understood issues that pertain to the local access networks.

This paper focusses on one of the most complex aspects of backbone network design. Most formal design tools suggested in the literature implicitely assume a 'perfect' network whose components are always functional. In reality, both the links and the nodes that are part of a communication network, though generally highly reliable, will ocasionally fail. Moreover, such failures could significantly impact the level of service the network offers to its users.

The extensively studied flow assignment problem is addressed here from a different perspective. The traditional approach separates the availability issues from the routing decisions, and concentrates on finding an optimal or near optimal flow assignment only for the no failure case. We choose instead a more natural approach that explicitly takes into account the effect of possible component breakdowns on the choice of routes. Moreover, primary and backup routes are concurrently chosen, in an attempt to minimize the average delay experienced by messages in the system, both under normal conditions, and when failures occur.

The paper is organized as follows: section 2 critically surveys the existing literature. The relevant problem background and model assumptions are presented in section 3. Section 4 introduces a new mathematical formulation of the problem, and the corresponding solution procedure is outlined in section 5. Results of computational experiments are reported in section 6, while section 7 contains some concluding remarks and suggests some possible directions for further research.

2 Related Research

The importance of producing a design that offers a high level of reliability to the users of the system is recognized by both practitioners and researchers in the field. An interesting fact that underscores the complexity of the issues involved is the lack of general agreement on how to best incorporate reliability considerations into the design procedure. Some authors, for instance, view them as being the essence of topological design, while others define stand alone reliability problems, sometimes only loosely related to the other aspects of network design. Another general characteristic of the field is that, in spite of the quite extensive existing body of knowledge on network performance analysis (e.g [1], [3], [4], [5], [17], [18], [19], [26], and [27]), and due no doubt to the inherent difficulty of the pertinent issues, the literature focussing on the design of reliable networks is quite limited. Moreover, since the probabilistic behavior of such systems is complex and hard to represent, deterministic reliability measures, usually related to the topology of the underlying network, are often used, both in analysis and in design problems, as substitutes for the more appropriate probabilistic measures. Unfortunatelly, the usefulness of purely deterministic measures is limited, since by their very nature, they cannot fully capture the effect of random failures on network performance.

An approach taken by several authors is to model the problem in terms of cost-reliability functions, i.e estimates of the direct relation between the probability of failure of a link and its cost. Special cost-reliability functions are defined in [6] for terrestrial and radio links. The network reliability measure used is global availability, defined as the probability that the network is at least *simply connected*. This criterion belongs to a family of related probabilistic measures (see [4] for a thorough discussion) that have been used extensively in the reliability literature. In spite of their popularity, such criteria overlook the issue of the level of performance experienced by users when failures occur, and therefore do not appropriately represent the operational characteristics of the network. The synthesis problem is defined as finding the link unavailability values that minimize the sum of the cost-reliability functions over all the links in the network, subject to an upper bound on the global availability. Due to the complex mathematical structure of the model, a gradient technique is used to solve it numerically. The authors suggest that the model can be used not only for choosing link unavailabilities for a given topology, but also for the topological design of the network, by starting with a complete graph (or with a graph with a high connectivity), and then eliminating links that have an unavailability of one in the final solution. The combinatorial structure of the problem renders this approach feasible only for very small networks (a few nodes).

A similar model appears in [23]. The design problem is defined as maximizing the terminal reliability, either for all or only for a subset of origin-destination pairs, under a budget constraint, an approach that mirrors the one in the earlier paper mentioned above. It suffers therefore from the same drawback, i.e it only indirectly and insufficiently takes into account the level of performance under failures. The relationship that exists between the reliability of a link and its cost is considered, the decision being to determine optimal values for investment variables, that define how much to spend on each link in the network. Two of the three iterative algorithms suggested for solving the model generate optimal solutions, and their computational requirements renders them impractical for all but very small networks. The third is a heuristic, based on an approximate evaluation of the objective function. The algorithms were tested on three small size networks, and for linear and exponential cost-reliability functions. On these examples, the heuristic provided good approximations of the optimal solutions.

A different approach is taken in [24] and [25]. The impact on network reliability of the random character of failures is completely ignored, and the connectivity of the underlying graph is used as the sole measure of network invulnerability. The end result is that, under the stated assumptions and conditions, a family of graphs of diameter two is found to be optimal whenever the average line utilization is not higher than .5, while a complete graph is the best configuration otherwise. Though the conclusions are interesting from a theoretical point of view, they have little practical relevance. Networks often operate above a .5 average line utilization, but the cost of a complete topology is prohibitive for all but the most trivial cases.

Finally, [20] contains an extensive analysis of several criteria, deterministic as well as probabilistic, that could be used in the selection of a primary route, together with a set of alternative routes to operate as backups, or for load balancing. An unrealistic premise for the analysis is the assumption that no traffic estimates are available. As a result, the selection of paths for each origin-destination pair is done independently of the other communicating pairs in the network. This significantly reduces the relevance of the suggested solutions, since the interaction that takes place inside the network between the traffic belonging to different origin-destination pairs is one of the main determinants of the overall performance.

As evidenced by the preceding discussion, the existing models for the synthesis of backbone networks overlook the importance of operational aspects, or at most consider them only in an indirect and incomplete manner. One important example of such an operational characteristic is the degradation in the performance level the network experiences in the presence of failures. For specified link capacities and routing tables, the routing mechanism may no longer be able to guarantee communication between certain origin-destination pairs, in spite of the network still being physically connected. This is due to the overload that failures may induce on some of the links that are still operational. Such congested situations can eventually be solved by an appropriate flow control mechanism, but their cost, in terms of network overhead as well as of user disatisfaction can be high. The argument we make here is that failure related congestion can be reduced by taking into account the effect of breakdowns when choosing primary and alternate routes, and thus explicitly expressing the relationship that exists between component failures, and the corresponding degradation in the network performance level.

3 Problem Definition

We address the problem of *simultaneously* selecting fixed primary and secondary routes for all the origin-destination pairs in a network with unreliable links. The secondary routes serve as backup, and are used whenever the corresponding primary is not available. The network topology, the link capacities, and estimates of external traffic requirements are are assumed to be known (see [12] for a justification of the choice of a static routing strategy, and for a short description of their implementation).

The following is a typical 'scenario', describing the sequence of events that are likely to be generated in the network as a result of a link failure (specific details will differ from implementation to implementation). Notice that at least three different levels of protocols, data link, network, and session, may be actively involved in coping with the failure:

- after a few unsuccessful attempts at transmitting over the link, the data link protocol eventually recognizes the failure and notifies the network layer protocol operating in the same switch. Packets queuing at the switch for transmission over the unavailable link are discarded from the buffers.
- the network layer protocol generates control messages destined for the appropriate origin nodes situated on the network boundary, and informing them that the route currently being used has become unavailable.
- upon reception of the control message, the network layer protocol in the origin nodes attempts to set up alternate (secondary) routes, without disrupting the already established sessions. Also, the protocol is responsible for insuring that the messages that were in transit when the failure occured are not lost. Instead, they are retransmitted along the newly set up route.

- if no alternate route can be established, the higher level protocol is informed that a disconnection has taken place. Eventually, the sessions are interrupted, and the users notified. Moreover, all new requests for session establishment arriving during the failure are rejected until the corresponding boundary nodes are informed that at least one of the failed links has again become operational.

This 'flurry of activity' that results from a simple link failure can represent a significant overhead. Obviously, the higher the number of active primary routes that were using the failed link, the higher the overhead incurred, since a larger number of connections are affected by the failure.

In terms of costs directly attributable to the failure, it is possible to distinguish between a time dependent component, corresponding to the additional delay experienced by the messages in the network due to the degradation in performance when alternate routes are used, on one hand, and a time independent component, comprising the overhead incurred in establishing a new route and in retransmitting the lost messages, on the other. Other difficult to capture components, such as the cost associated with the additional control traffic generated by a user who repeatedly (and unsuccesfully) tries to reestablish an aborted session, could be made part of the time independent component as well.

Two sets of assumptions are needed for the model. First, the queuing phenomena are captured by viewing messages arriving at a link as customers, while the link itself is a server, with a rate determined by its capacity and by the average message length. The following standard assumptions are used for modeling the resulting network of queues: Poisson message arrivals, exponentially distributed message lengths, negligible propagation delay, unlimited buffering space and no processing delay at the network nodes. Kleinrock's independence assumption [16] is also used.

In addition, the following assumptions are used to model the stochastic behavior of link failures:

Assumption 1: link failures are independent

Assumption 2: the time between successive failures of link l is exponentially distributed with parameter β_l

Assumption 3: only one link may be down at the same time, i.e no link may fail while another link is being repaired

The repair time for each link l in the network follows a general distribution, with an average failure duration of $1/\gamma_l$.

Notice that assumption 3 is not fully consistent with assumption 2. The fact that any new failure is prevented from occuring while a link is being repaired affects the characteristics of the failure arrival process, which as a result is no longer a real Poisson process.

Another approximation that must be introduced for the sake of tractability is to assume that, as a result of an event that modifies its state, the system instantly switches to another steady state, i.e the transient phenomena in the network are ignored. Thus, for example, upon the completion of a repair, the average flow supported by the link is assumed to immediately return to its normal value, determined by the number of different origin-destination pairs that use the link as part of their primary route. This ignores the fact that during the failure some or all of the communicating pairs were using alternate routes, and that a certain amount of time will pass until the knowledge about the availability of the link filters through the network and results in the traffic flow regaining its normal, steady state, pattern.

The above mentioned problems are not likely to have a significant effect on the behavior of the network whenever the following conditions are satisfied:

Assumption 4: the mean times between failures are much larger than the mean repair times.

Assumption 5: the message interarrival times are much shorter than the repair times.

The last two assumptions are not very restrictive and apply to many real life situations. For instance, in [28], operational data collected and averaged over a 30 days period for the IBM Information Network (IBM/IN) shows that the average length of an outage is of .55 hours, three orders of magnitude smaller than the reported mean time between failures of 231.69 hours. Similar values are given in [14] for ARPANET.

Assumptions 4 and 5 not only guarantee that steady state can be quickly reached, but also support assumption 3, by reducing the probability of another failure occuring during the downtime of a link. They do not however affect the probability of joint component failure, which could occur in the case of catastrophic events (e.g major storm, war, etc.), situations that are not captured by the models presented here. Hopefully, the frequency of such incidents is low enough to justify concentrating only on regular operating conditions.

[9,7,22]

4 Model Formulation

The following notation is used throughout the paper:

L =the index set of links in the network

D =unit cost of delay [$/month/message]

R =a set of candidate routes, that are part of the input to the model. The routes may be automatically generated, and/or may be provided by the users

Π =the set of communicating origin-destination pairs in the network.

S_p =the set of candidate routes for p, $p \in \Pi$. We assume that $S_p \cap S_q = \emptyset$ for $p \neq q$.

λ_r =the message arrival rate [messages/sec] of the unique origin-destination pair associated with route r, $r \in R$. We define $\lambda_p = \lambda_r$, $\forall r \in S_p$.

δ_{rl} =an indicator function, taking the value one if link l is used in route r, and zero otherwise.

$1/\mu$ =the average message length [bits/message]

F_l =the average bit flow on link l, $l \in L$

b_l = the probability that link l, $l \in L$ is down, i.e the proportion of time that link l is not operational

b_0 = the probability that all links are up, i.e the proportion of time the network is fully operational

Δ_{li} = the flow increment on link l, $l \in L$ due to the failure of link i, $i \in L$, $i \neq l$

C_p = time independent cost associated with the overhead caused by the failure of a link in the primary route used by origin-destination pair $p \in \Pi$

x_r =a decision variable, which is one if route $r \in R$ is chosen to carry the flow of its associated origin-destination pair, and zero otherwise

u_r = a decision variable, which is 1 if route $r \in R$ is chosen as an alternate route for its associated origin-destination pair, and zero otherwise

The steady state values of b_l and b_0 are:

$$b_l = \frac{\beta_l/\gamma_l}{1 + \sum_{i \in L} \beta_i/\gamma_i}$$

$$b_0 = \frac{1}{1 + \sum_{i \in L} \beta_i/\gamma_i}$$

The network layer protocol reacts to the failure of a link by attempting to re-route the traffic. The implicit assumption here is that the protocol is such that only the sessions that use the failed link as part of their primary route, and whose traffic must be diverted to appropriate alternate routes, are affected. Therefore, as a result of the failure of link i, a link l in the network incurs a surge in the flow traversing it equivalent to the traffic associated with those origin-destination pairs that use link i as part of the primary route, and link l as part of the alternate route, provided that neither link is common to both routes. Formally, the flow increment on link l due to the failure of link i is given by the flow of those $p \in \Pi$ that satisfy both of the following conditions:

- $x_r \delta_{ri}(1 - \delta_{qi}) = 1 \; \forall r, q \in S_p, \; i \in L$
- $u_q \delta_{ql}(1 - \delta_{rl}) = 1 \; \forall r, q \in S_p, \; l \in L$

Δ_{li}, the increment of flow on link l due to the failure of link i can then be expressed in terms of the decision variables as:

$$\Delta_{li} = \sum_{p \in \Pi} \sum_{r \in S_p} \sum_{\substack{q \in S_p \\ q \neq r}} \lambda_r x_r u_q \delta_{ri} \delta_{ql} (1 - \delta_{qi})(1 - \delta_{rl})/\mu \tag{1}$$

In addition, the origin-destination pairs that use the failed link as part of both their primary and secondary routes are not able to communicate during the breakdown. The average flow generated by these pairs is:

$$\Delta_{ll} = \sum_{p \in \Pi} \sum_{r \in S_p} \sum_{\substack{q \in S_p \\ q \neq r}} \lambda_r x_r u_q \delta_{ql} \delta_{rl}/\mu$$

Therefore, the problem of simultaneosly selecting primary and alternate routes for all the origin-destination pairs in the network is equivalent to that of finding the x_r and u_q values that satisfy:

Problem *P*

$$Z = \min \left\{ \sum_{l \in L} D b_0 \frac{F_l}{Q_l - F_l} + D \sum_{\substack{i \in L \\ i \neq l}} b_i \frac{F_l + \Delta_{li}}{Q_l - F_l - \Delta_{li}} + D \frac{b_l}{\gamma_l} \Delta_{ll} + \right. \tag{2}$$

$$\left. \sum_{p \in \Pi} C_p \beta_l \sum_{r \in S_p} x_r \delta_{rl} \right\}$$

subject to:

$$F_l + \Delta_{li} \leq Q_l \; \forall l \neq i \in L \tag{3}$$

$$\sum_{r \in S_p} x_r = 1 \; \forall p \in \Pi \tag{4}$$

$$\sum_{q \in S_p} u_q = 1 \; \forall p \in \Pi \tag{5}$$

$$x_r, u_q = 0, 1 \; r, q \in R \tag{6}$$

where: $F_l = \sum_{r \in R} \lambda_r x_r \delta_{rl} / \mu$, and β_l corresponds to the average number of failures of link l over the planning horizon.

The objective function captures the following cost components:

1. *time dependent* costs, for each link l in the network:
   - total average queuing cost when all links are operational;
   - total average queuing cost induced by the increased flow link l must accomodate when some other link fails; and
   - the penalty incurred as a result of the failure of link l
2. *time independent* cost, for each origin-destination pair p in the network.

The purpose of the constraints in (3) is to avoid congested situations, i.e to ensure that the flow on each link is still feasible in terms of its capacity, even when surges in traffic occur. The constraints in (4) and (5) ensure that only one primary and one alternate route are chosen for each origin-destination pair, respectively.

The above general model allows for any pair of routes to be chosen as primary and alternate, even for a single route to serve both purposes. On the other hand, over the relevant parameter ranges the structure of the objective function is such that choices of non-disjoint routes are likely to incurr heavy penalties, and therefore good feasible solutions to the problem will consist of only disjoint route pairs. This hypothesis was in fact confirmed by initial experiments. As a result, the general model was dropped from further consideration, and we concentrated instead on the link disjoint case. Notice that under the single failure assumption, choosing link disjoint paths guarantees that no sessions have to be aborted, since the alternate route is always available.

The objective function of the link disjoint problem is:

$$Z_P = \min \left\{ \sum_{l \in L} D b_0 \frac{F_l}{Q_l - F_l} + D \sum_{\substack{i \in L \\ i \neq l}} b_i \frac{F_l + \Delta_{li}}{Q_l - F_l - \Delta_{li}} + \sum_{p \in \Pi} C_p \beta_l \sum_{r \in S_p} x_r \delta_{rl} \right\}$$

subject to (3)-(6), and:

$$\sum_{l \in L} x_r u_q \delta_{rl} \delta_{ql} = 0 \ \forall r, q \in S_p, \ p \in \Pi \tag{7}$$

where the new constraint enforces the link disjoint condition.

The structure of the problem becomes mathematically more tractable when expressed in terms of a derived set of decision variables. $f_l = F_l/Q_l, \ l \in L$ is the utilization of link l under normal conditions, while $f_{li} = \Delta_{li}/Q_l, \ l \neq i$, is defined as the increase in the utilization of link l corresponding to the flow deviated from link i, while the latter is not operational. The problem then becomes:

Problem P

$$Z_P = \min \left\{ \sum_{l \in L} D b_0 \frac{f_l}{1 - f_l} + D \sum_{\substack{i \in L \\ i \neq l}} b_i \frac{f_l + f_{li}}{1 - f_l - f_{li}} + \sum_{p \in \Pi} C_p \beta_l \sum_{r \in S_p} x_r \delta_{rl} \right\}$$

subject to:

$$\sum_{p \in \Pi} \sum_{r \in S_p} \lambda_r x_r \delta_{rl} / \mu Q_l \leq f_l \ \forall l \in L \tag{8}$$

$$\sum_{p \in \Pi} \sum_{r, q \in S_p} \lambda_r x_r u_q \delta_{ri} \delta_{ql} / \mu Q_l \leq f_{li} \ \forall l \neq i \in L \tag{9}$$

$$f_l + f_{li} \leq 1 \ \forall l \neq i \in L$$

$$f_l, f_{li} \geq 0 \ \forall l \neq i \in L$$

(4) - (6) and (7).

5 Solution Procedure

The constraints in (8) and (9) are equivalent to those of the multiconstraint knapsack problem, a well known problem in the combinatorial optimization literature, shown to belong to the NP-complete class. Hence, **Problem P** is at least as complex as the multiconstrained knapsack problem, not an encouraging fact when its solution is considered. Fortunatelly, the very nature of the problem, i.e the fact that input data consisits primarily of traffic estimates, makes finding the optimal solution less critical. The approach adopted here is instead to devise heuristic procedures for obtaining good feasible solutions (that also correspond to upper bounds on the value of the optimal solution), together with a method for generating lower bounds on the optimal value. Since the value of the optimal solution lies somewhere between the best upper and lower bounds obtained, this bounding technique provides for an effective way to ascertain the quality of the heuristic solution.

A Lagrangean problem is obtained by associating a Lagrange multiplier with each of the constraints in (8) and (9) and adding them to the objective function. The resulting problem is decomposable over the links and the origin-destination pairs in the network, leading to $|L| + |\Pi|$ much simpler subproblems.

For each link l, $l \in L$, the following subproblem is obtained:

Subproblem $P(\alpha, l)$

$$L(\alpha, l) = \min \left\{ Db_0 \frac{f_l}{1 - f_l} + \alpha_l f_l + \sum_{\substack{i \in L \\ i \neq l}} Db_i \frac{f_l + f_{li}}{1 - f_l - f_{li}} + \alpha_{li} f_{li} \right\}$$

subject to:

$$\begin{aligned} f_l + f_{li} &\leq 1 \; \forall i \neq l \in L \\ f_l, f_{li} &\geq 0 \; \forall i \neq l \in L \end{aligned} \tag{10}$$

where $\{\alpha_l \leq 0, \alpha_{li} \leq 0, i \neq l\}$ are the Lagrange multiplier values.

The above is a n variables continous minimization over a closed domain (where $n =$ the number of links in the network), and its relatively complex structure precludes an analytic solution. A numerical algorithm of polynomial complexity, that very efficiently reaches the optimal solution by exploiting the special structure of the problem, is used instead.

The subproblem for origin-destination $p \in \Pi$ is of the form:

Subproblem $P(\alpha, p)$

$$L(\alpha, p) = \min \left\{ \sum_{r \in S_p} (C_p N_r + a_r) x_r + x_r \sum_{q \in S_p} b_{rq} u_q \right\}$$

subject to:

$$\sum_{r \in S_p} x_r = 1$$

$$\sum_{r \in S_p} u_r = 1$$

$$\sum_{l \in L} x_r u_q \delta_{rl} \delta_{ri} = 0 \; \forall r \neq q \in S_p$$

where a_r, b_{rq} and N_r are used for ease of notation, and are defined as :

$$a_r = \sum_{l \in L} -\alpha_l \lambda_r \delta_{rl} / \mu Q_l$$

$$b_{rq} = \sum_{l \in L} \sum_{\substack{i \in L \\ i \neq l}} -\alpha_{li} \lambda_r \delta_{ql} \delta_{ri} / \mu Q_l$$

$$N_r = \sum_{l \in L} \beta_l \delta_{rl} = \text{the average number of failures of route } r$$

A simple two-step procedure, that determines the best x_r value when each of the u_q variables are set to one, is used to solve these subproblems.

The optimal Lagrangean objective function value is then:

$$L(\alpha) = \sum_{l \in L} L(\alpha, l) + \sum_{p \in \Pi} L(\alpha, p)$$

A standard result in optimization theory [13] shows that for any nonpositive vector of multipliers, $L(\alpha)$ is a lower bound on the value of the original objective function Z_P. Given the importance ofthe lower bound, used as a benchmark against which the quality of the heuristic solutions are judged, two methods were used in order to improve the value of the Lagrangean, i.e to reach a value as close as possible to:

$$L(\alpha^*) = \max_{\alpha \leq 0} L(\alpha) \tag{11}$$

1. Based on the candidate routes each link l is part of, upper bounds on the utilization variables f_l, are computed. These constraints would be redundant in the original problem, but are often binding in the relaxed one, thus reducing the feasible region over which the Lagrangean is defined, and increasing its objective function value.

2. A subgradient optimization procedure, a technique that iteratively finds an estimate of the vector α^* defined in (11), is used to further tighten the lower bound. Examples of previous succesful applications of this method can be found in [2], [8],[11], [15].

Further details on the mathematical analysis and algorithmical implementation of the problem can be found in [??? **** In the past, you talked about two versions of this paper. This would be the somewhat less technical one. The other one could go to....? Seems too narrow focussed for Manag. Sc. and not original enough from an oper. res. point of view for OR... IEEE Comm. again? Networks? ****). **I don't think we talked about this. If this will be the only version we try to get out of this model, I will put some additional algorithmical details here, else I will leave it for the 2nd iteration**

Feasible solutions are generated by a heuristic procedure that uses the Lagrangean solution as a starting point. At each subgradient iteration, in addition to checking the Lagrangean for feasibility, a list of potentially good primary and secondary routes is also obtained for each origin-destination pair. Routes are then randomly chosen from each list, and the resulting assignment is checked for feasibility. Whenever a lower cost solution is reached, its value and associated route choices become the best current feasible solution, and the search continues. This simple and efficient procedure proved quite effective in generating upper bounds to the problem.

6 Computational Results

The procedure presented in the previous section is implemented in a flexible system that allows the user to define the network characteristics and model parameters, as well as to control the number of subgradient iterations performed. The program produces a comprehensive output that, in addition to the lower and upper bounds, also provides the full details of the best feasible solution generated.

To better understand the behavior of the algorithm and the interactions between the various model parameters, the procedure was first tested on the small network shown in figure 1. Two sessions are active at each source node, each generating an average of one message per second, resulting in an average traffic of four messages per second for both directions. Table 1 shows the primary and secondary routes selected for each communicating pair, as well as the choices when only primary routes are considered. Notice that the values for the single route case ignore the effect of failures, which explains the lower values for the average message delay. On the other hand, once the impact of failures is taken into account, the average message delay increases almost tenfold, to more than 15000 msec. The change is easily explained by the fact that, under the set of assumptions used here, a failed link causes the sessions belonging to the communicating pairs that use it as part of their route to be interrupted until the link becomes operational again. Figures 2 and 3 compare the average and the maximum link utilization corresponding to the best feasible solution. Even for such a small example, there is a considerable gap between the average load supported by a link, and the maximum amount of traffic it has to accomodate as a result of failures. Since the size of the gap is determined by the link probability of failure and by the number of sessions that have to be rerouted, it will become even more significant as the number of origin-destination pairs in the network increases. This general characteristic of feasible solutions is a direct consequence of the way the design problem is defined. The capacity constraints in (3) are quite restrictive, since they imply that the routing decisions have to be such that, even when short and very low probability failures take place, the network must still be able to support the additional traffic without any interruption of service. In many cases, this constraint significantly reduces the number of feasible solutions to the problem. Moreover, it is easy to envisage situations where such a condition is far too restrictive, i.e where the average long term network performance is only marginally affected by the occasional interruption of some active sessions. It should be mentioned that the models presented here represent a first attempt at tackling a particularly complex problem. Future work with the problem will attempt to address additional aspects of problem.

Next, the algorithm was tested on four larger network topologies. The fixed link capacities used in these experiments, as well as the statistics characterizing the number of candidate routes defined for each origin-destination pair, appear in figures 4 through 7. Each node is assumed to communicate with each other node in the network, and generates the same amount of traffic as in the previous example. Without loss of generality, the fixed failure cost C_p was ignored in the experiments, i.e $C_p = 0, \ \forall p \in \Pi$, and the unit cost of delay $D = 2000$. The convergence of the algorithm is satisfactory on an average, however for some cases it is significantly worse than the results reported in [10],[11],[12] and [21], where similar procedures were applied to related problems. We postulate that the poorer performance is caused mainly by a feasible solution which is relatively far away from the optimum, rather than by an inappropriate lower bound value. As the previous discussion shows, this is chiefly due to the relative difficulty with which feasible solutions to the problem can be generated.

The results summarized in table 2 correspond to the cases where the average message length varies from 400 to 500 bits. For higher values of the average message length, the algorithm failed to identify feasible solutions. The average failure arrival rate β_l and the average repair time $1/\gamma_l$, have the same value for all links in the network, and are kept constant at 10^{-4} failures per second and 1000 seconds, respectively. The performance of the algorithm is sensitive to the total load in the network, and it tends to worsen as the load increases, a fact that agrees with the earlier statements about the characteristics of the feasible solutions. For reduced loads, when there is less of a variance in the values assumed by the average link utilization, the performance is consistently good.

For the problems presented in table 3, the average message length $1/\mu$ and the average repair duration $1/\gamma_l$ are kept constant at 450 bits and 1000 seconds, respectively, while the average failure rate β_l is allowed to vary. Expectably, as the average time between failures increases, the average message delay in the network tends to decrease. The convergence of the algorithm does not seem to be affected by the values of β_l.

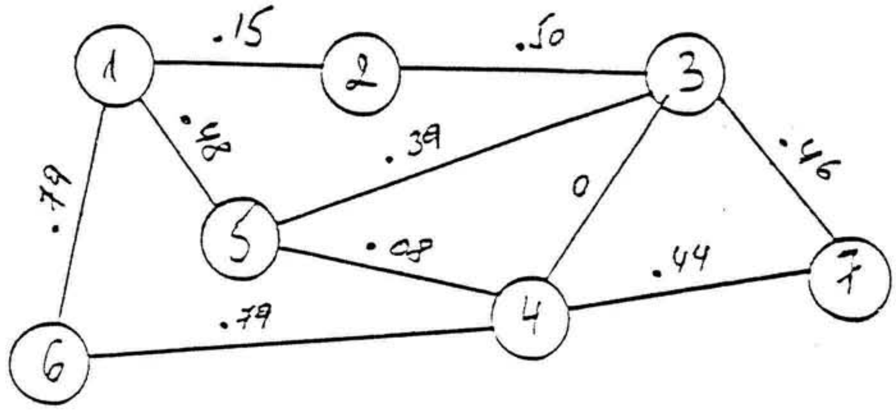
Table 4 contains the results for varying average repair durations, for fixed message lengths $1/\mu = 450$ and failure arrival rates $\beta_l = 10^{-4}$ values. The average message delay in the network steadily increases with the average link repair time. As in the previous case, the performance of the algorithm is insensitive to variations in the value of the $1/\gamma_l$ parameter.

Since the external traffic arrival values used in generating a solution may sometimes be only rough estimates of the actual values, it is important to assess the robustness of the model with respect to this parameter. Table 5 reports results of sensitivity analysis experiments, in which errors were randomly generated within intervals ranging from $\pm 10\%$ to a high value of $\pm 50\%$. The robustness measure adopted is the ratio $C(\Lambda_a, A_e)/C(\Lambda_a, A_a)$, where $C(\Lambda_a, A_e)$ is the cost under real traffic conditions of the solution obtained based on the estimated values of the external traffic arrivals, and $C(\Lambda_a, A_a)$ is the cost of the 'ideal' solution, i.e the solution that could have been obtained if the real traffic rates were available. Additional details about our approach to the sensitivity analysis issue can be found in [12]. In most cases, the solutions generated by the algorithm proved to be robust. The heuristic nature of the solution procedure explains the ratio values that are less than one. These correspond to cases where the solution based on the estimates is actually better than the solution based on the real (but unavailable) traffic values. Notice that in some cases the solution obtained based on the estimates proved to be infeasible when the actual traffic values were plugged in. This again suggests that the model formulation is too restrictive in terms of the traffic feasibility conditions in (3).

7 Conclusions

A new perspective on the problem of effective message routing in a network subjected to link failures is offered. The objective of the design process is defined as the simultaneous selection of two routes for each origin-destination pair, a primary route to be used under normal conditions, and an alternate or secondary route, to be adopted whenever the primary is rendered inoperational by failures. By concurrently choosing the two routes, it is possible to control the increment in congestion that failures induce in the system. A procedure that generates feasible solutions, together with lower bounds on the optimal objective function value, is presented. The numerical experiments show a relatively high variance in the gap between the best feasible solution provided by the heuristic and the best lower bound obtained from the Lagrangian, the gap being generally higher for longer average message lengths, i.e when failures in the network cause some active links to operate close to saturation.

Link disjointedness is a strong and possible unnecessary restrictive condition, whose enforcement may result in the problem being infeasible. The impact of this constraint, together with alternative problem definitions, will be examined in the future.



Origin-destination pairs:

1 <—> 3
1 <—> 4
1 <—> 7
2 <—> 7

Figure 1: Topology and average link utilization for the EX network (link capacities = 19200 bps)

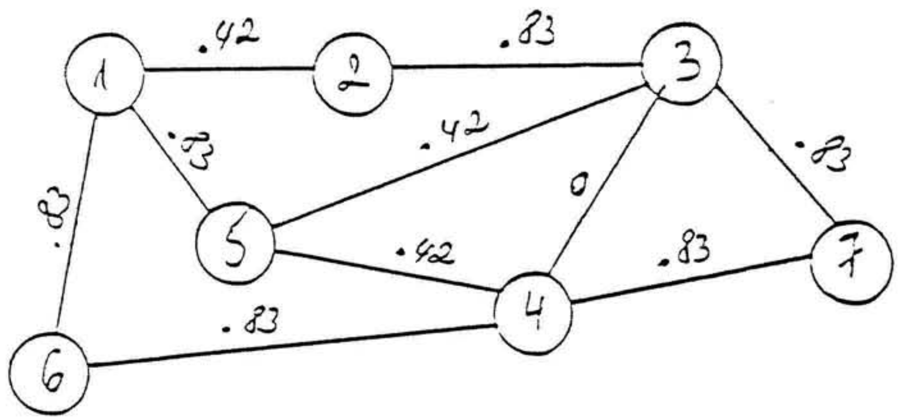
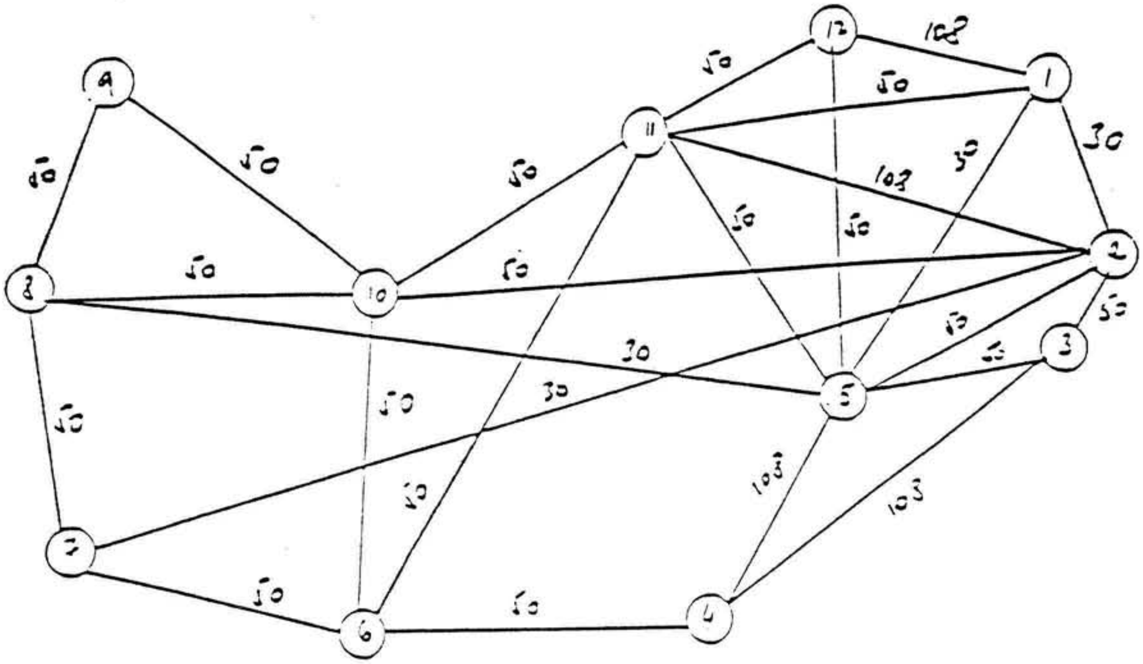


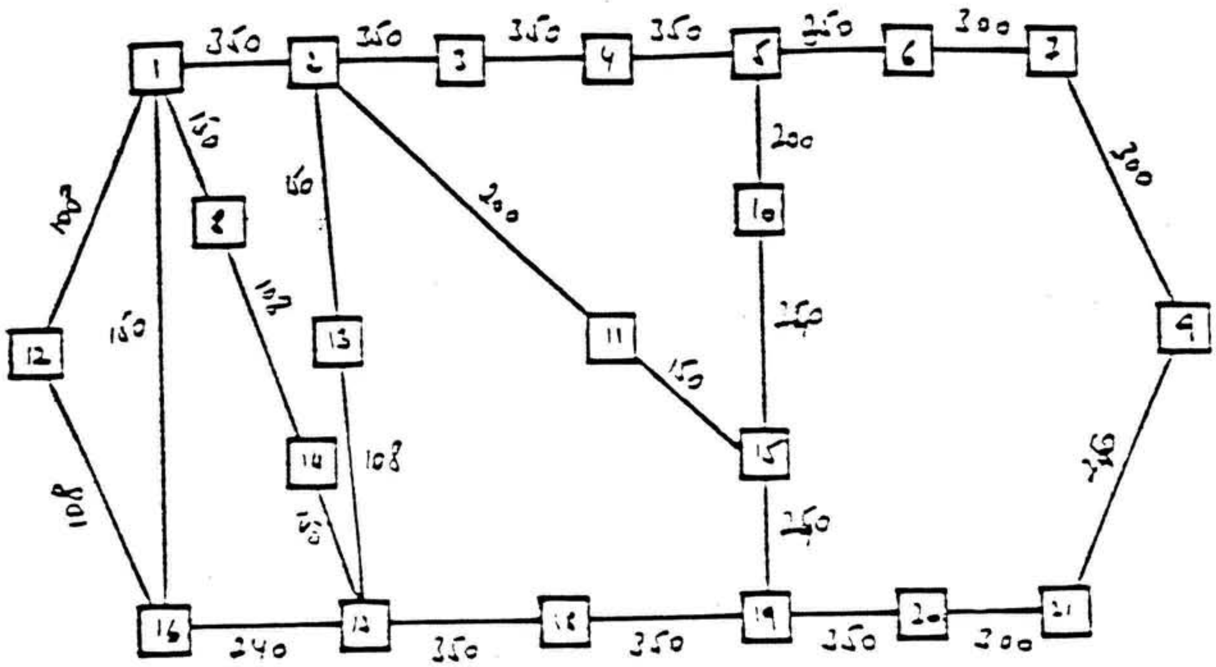
Figure 2: Topology and maximum link utilization for the EX network (link capacities = 19200 bps)



No. of Routes/Communicating Pair:

Average = 4.0
Maximum = 6
Minimum = 3

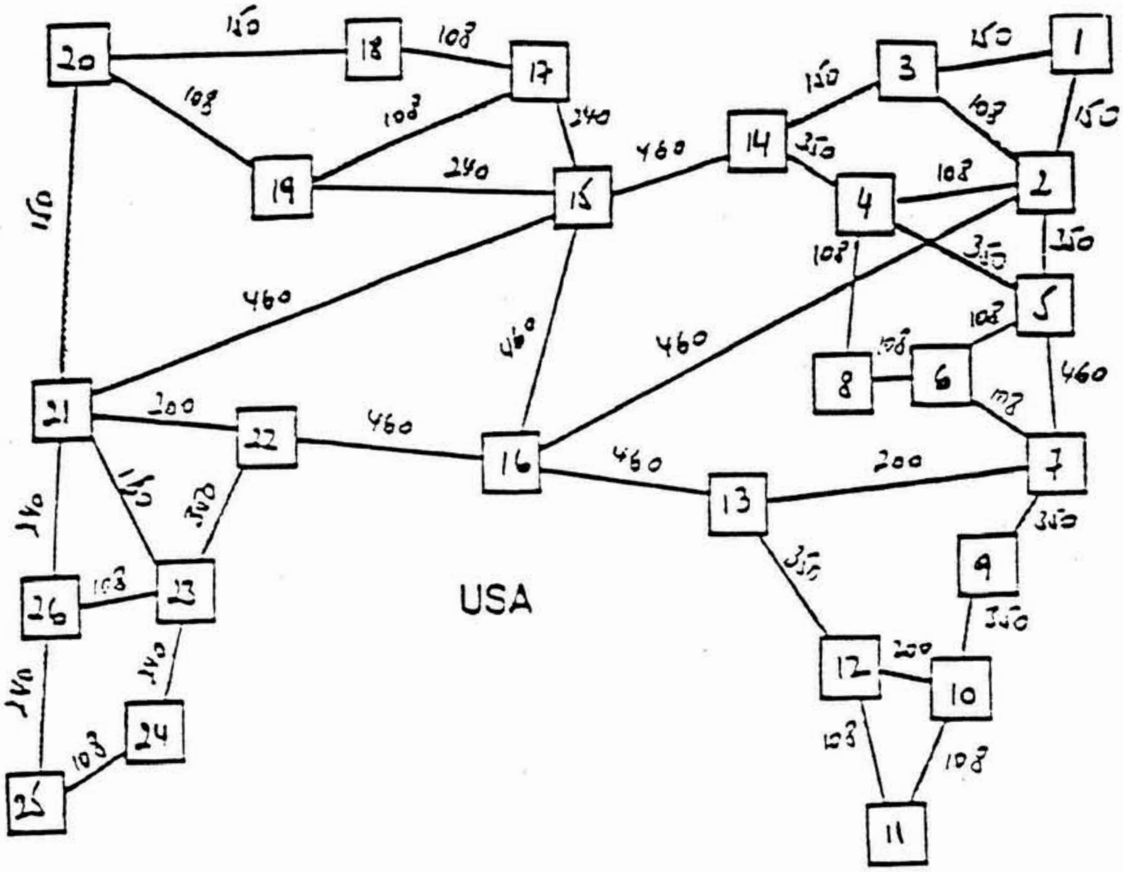
Figure 3: Topology and capacity values for the GTE network



No. of Routes/Communicating Pair:

Average	=	2.7
Maximum	=	4
Minimum	=	2

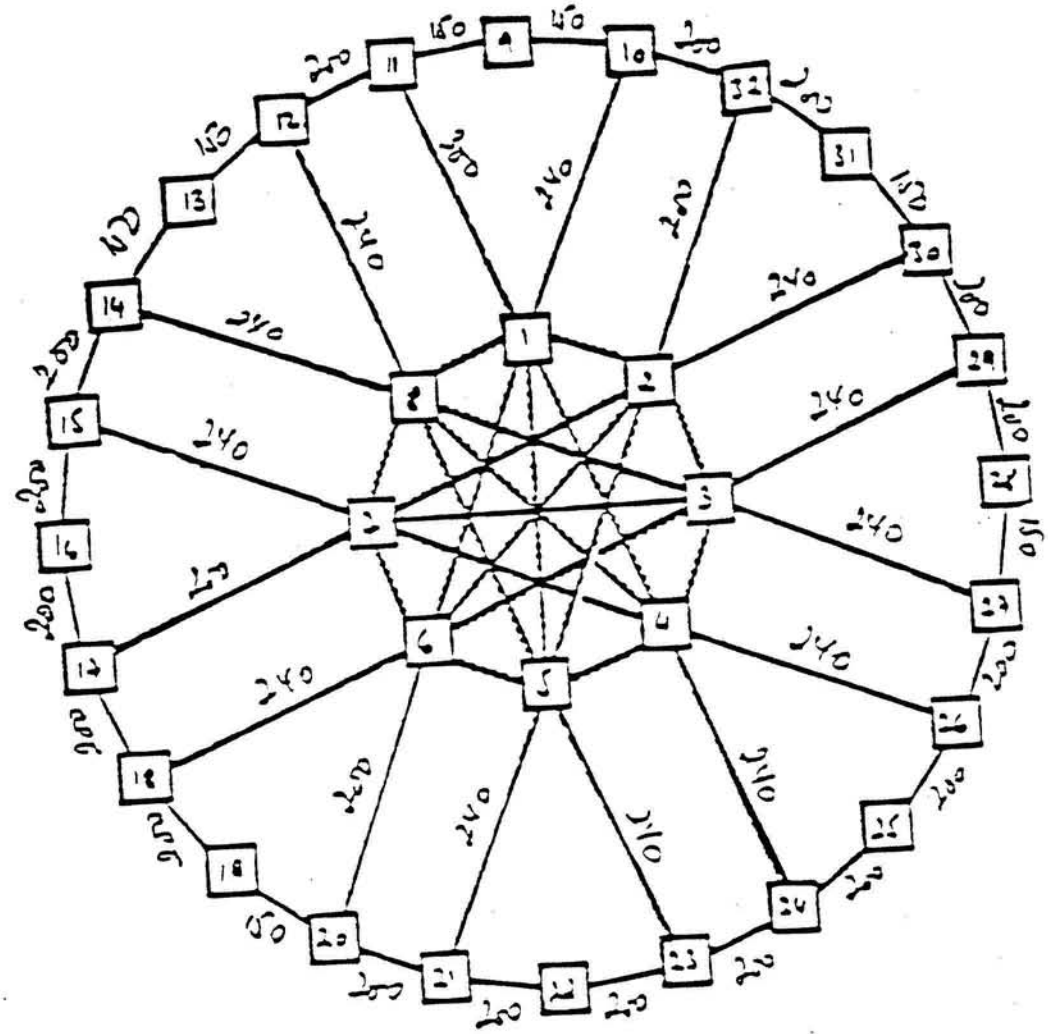
Figure 4: Topology and capacity values for the ARPA network




```
No. of Routes/Communicating Pair:

Average  =  3.2
Maximum  =  6
Minimum  =  2
```

Figure 5: Topology and capacity values for the USA network



No. of Routes/Communicating Pair:

Average = 2.9
Maximum = 6
Minimum = 2

Figure 6: Topology and capacity values for the RING network

Origin - destination	Single route (no failures)	Primary - secondary (with failures)
1↔3	1,2,3	P: 1,5,3 S: 1,2,3
1↔4	1,2,3,4	P: 1,6,4 S: 1,5,4
1↔7	1,2,3,7	P: 1,6,4,7 S: 1,2,3,7
2↔7	2,3,7	P: 2,3,7 S: 2,1,5,4,7
Average message delay	178.6 msec	508.9 msec

Table 1: Comparative results for the single route and the primary + secondary cases (EX network, average message length = 1000 bits)

Network ID	Average message length	Lower Bound	Upper Bound	Upper/ Lower	Average message delay
GTE	500	41345	45292	1.09	42.9
GTE	450	32278	35330	1.09	33.7
GTE	400	27361	27571	1.01	26.1
ARPA	500	60175	86414	1.43	25.7
ARPA	450	50650	59182	1.17	17.6
ARPA	400	41500	44443	1.07	13.2
USA	500	99749	149364	1.50	28.7
USA	450	77959	83034	1.06	15.9
USA	400	59860	62950	1.05	12.1
RING	500	163276	229224	1.40	28.9
RING	450	122903	135815	1.120	17.1
RING	400	100787	104400	1.03	13.2

Table 2: Summary of computational results for different average message lengths ($\beta = 10^{-4}, 1/\gamma = 1000$)

Network ID	Failure rate	Lower bound	Upper bound	Upper/ lower	Average message length
GTE	10^{-3}	33902	36888	1.09	34.9
GTE	10^{-4}	32278	35330	1.09	33.7
GTE	10^{-5}	30942	31250	1.01	29.6
ARPA	10^{-3}	57024	66225	1.16	19.7
ARPA	10^{-4}	50650	59182	1.17	17.6
ARPA	10^{-5}	37377	43774	1.17	13.0
USA	10^{-3}	87089	89001	1.02	17.1
USA	10^{-4}	77959	83034	1.06	16.0
USA	10^{-5}	56015	69450	1.24	13.4
RING	10^{-3}	123281	139403	1.13	17.6
RING	10^{-4}	122903	135815	1.10	17.1
RING	10^{-5}	102848	124663	1.21	15.71

Table 3: Summary of computational experiments for different failure rates $(1/\mu = 450, 1/\gamma = 1000)$

Network ID	Average repair length	Lower bound	U pper bound	Upper/ lower	Average message delay
GTE	500	30777	33993	1.10	32.2
GTE	1000	32278	35330	1.09	33.7
GTE	3000	33441	36452	1.09	34.5
ARPA	500	46745	54369	1.16	16.2
ARPA	1000	50650	59182	1.17	17.6
ARPA	3000	54610	64046	1.17	19.1
USA	500	65548	80040	1.22	15.4
USA	1000	77959	83034	1.06	16.0
USA	3000	77692	87050	1.12	16.7
RING	500	116313	132772	1.14	16.7
RING	1000	122903	135815	1.10	17.1
RING	3000	128839	142006	1.10	17.9

Table 4: Summary of computational results for different repair lengths ($1/\mu = 450, \beta = 10^{-4}$)

Network ID	Error range (%)	$C(\lambda_a, A_e)$	$C(\lambda_a, A_a)$	$C(\lambda_a, A_e)/C(\lambda_a, A_a)$
GTE	(-10,+10)	42853	36160	1.185
GTE	(-30,+30)	50399	39230	1.279*
GTE	(-50,+50)	37816	39230	0.964*
ARPA	(-10,+10)	59087	58570	1.009
ARPA	(-30,+30)	55520	55375	1.003
ARPA	(-50,+50)	53201	53124	1.001
USA	(-10,+10)	88775	84809	1.047
USA	(-30,+30)	79401	78538	1.011
USA	(-50,+50)	75319	75049	1.004
RING	(-10,+10)	134299	136192	0.986
RING	(-30,+30)	141455	137545	1.028*
RING	(-50,+50)	133083	131250	1.014

Note: For the cases marked with an *, in 2 out of the 5 cases $C(\Lambda_a, A_e)$ corresponded to infeasible solutions

Table 5: Impact of the errors in estimating the external arrival rates

References

[1] J.A. Abraham. An improved algorithm for network reliability. *IEEE Trans. Reliab.*, R-28, np.1:58–61, 1979.

[2] P. Afentakis and B. Gavish. Computationally efficient optimal solutions to the lot sizing problem in multistage assembly problems. *Manag. Sc.*, 30:222–239, 1984.

[3] K.K. Aggarwal and S. Rai. Reliability evaluation in computer communication networks. *IEEE Trans. Reliab.*, R-30, no. 1:32–35, 1981.

[4] M.O. Ball. Computing network reliability. *Oper. Res.*, 823–838, 1979.

[5] J. Cavers. Cutset manipulation for communication and networks reliability estimation. *IEEE Trans. Reliab.*, COM-23:569–575, 1975.

[6] L. Fratta and U. Montanari. A recursive method based on case analysis for computing network terminal reliability. *IEEE Trans. Commun.*, COM-26:1166–1177, 1978.

[7] B. Gavish. A general model for the topological design of computer networks. In *Globcom'86*, pages 1584–1588, 1986.

[8] B. Gavish. Topological design of centralized computer networks: formulation and algorithms. *Networks*, 12:355–377, 1982.

[9] B. Gavish and K. Altinkemer. *Backbone network design tools with economic tradeoffs.* Working Paper, Krannert Graduate School of Management, Purdue University, West Lafayette, IN, 47907, 1987.

[10] B. Gavish and S.L. Hantler. An algorithm for the optimal route selection in SNA networks. *IEEE Trans. Commun.*, COM-31:1154–1161, 1983.

[11] B. Gavish and I. Neuman. Capacity and flow assignment in large computer networks. In *Proceedings, INFOCOM'86, Miami, Florida*, pages 275–284, april 1986.

[12] B. Gavish and I. Neuman. A system for routing and capacity assignment in computer communication networks. *IEEE Trans. on Commun.* (to appear).

[13] A.M. Geoffrion. Lagrangean relaxation and its uses in integer programming. *Mathematical Programmg Study*, 2:82–114, 1974.

[14] D.S. Grubb and I.W. Cotton. Criteria for evaluation of data communication services. *Computer Networks*, 1:325–340, 1979.

[15] M. Held and R.M. Karp. The travelling salesman problem and minimum spanning trees, part ii. *Math. Program.*, 1:6–25, 1971.

[16] L. Kleinrock. *Communication nets: stochastic message flow and delay.* McGraw-Hill, 1964.

[17] P. Kubat. Reliability analysis for integrated networks with application to burst switching. *IEEE Trans. Commun.*, COM-34, no. 6:564–568, 1986.

[18] R.V. Laue. A versatile queuing model for data switching. In *Symp. on Data Commun.*, pages 118–128, 1981.

[19] V.O.K. Li and J.A. Silvester. Performance analysis of networks with unreliable components. *IEEE trans. Commun*, COM-32, no. 10:1105–1110, 1986.

[20] K.S. Natarjan, D.T. Tang, and K. Maruyama. *On the selection of communication paths in computer networks.* Technical Report RC 8002, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, 1979.

[21] I. Neuman. *Routing in a network with diffrent classes of messages.* Technical Report, Center for Research on Information Systems, Information Systems Area, Graduate School of Business Administration, New York University, 1986.

[22] H. Pirkul and S. Narasimhan. *A new algorithm for the design of backbone networks.* Working Paper, College of Business, Ohio State University, Columbus, OH, 1987.

[23] C.S. Raghavendra and S. Hariri. Reliability optimizaton in the design of distributed systems. *IEEE Trans. Soft. Eng*, 1184–1193, 1985.

[24] I. Rubin. The delay-capacity product for message -switched communication networks. *Jour. of Combinatorics, Inf. and Syst. Sc*, 1, no.2:48–68, 1976.

[25] I. Rubin. On reliable topological structures for message switching networks. *IEEE Trans. Commun.*, COM-26:62–74, 1978.

[26] A. Satyanarayana and A. Prabhakar. New topological formula and rapid algorithm for topological analysis of complex networks. *IEEE Trans. Reliab.*, 82–100, 1978.

[27] A.W. Shogan. Sequential bounding of the reliability of a stochastic network. *Oper. Res.*, 34, no. 6:1027–1044, 1976.

[28] R.C. Soucy and R.M. Bailey. IBM information network performance and availability measurement. In *Proc. Nat. Comp. Conf.*, pages 655–662, 1983.