

Backbone Network Design with QoS Requirements

Hong-Hsu Yen¹

Frank Yeong-Sung Lin

Department of Information Management

National Taiwan University

Taipei, Taiwan, R.O.C.

Tel: +886-2-2363-8423

Fax: +886-2-2758-4773

Email: d4725001@im.ntu.edu.tw

yslin@im.ntu.edu.tw

Keywords: Backbone network design, Optimization, Lagrangean relaxation, QoS.

Abstract

In this paper, we consider the backbone network design problem with a full set of QoS requirements and take the approach of mathematical programming to solve the problem. Unlike previous work, we consider both the transmission line cost and the switch cost with sophisticated cost structures. And the QoS requirements that we considered include the average packet delay, end-to-end packet delay and network reliability constraints specified by end-to-end connectivity requirements. We formulate the problem as a combinatorial optimization problem where the objective function is to minimize the total network deployment cost subject to the aforementioned QoS constraints. Besides the integrality constraints, the nonlinear and the nonconvex properties associated with the problem formulation make it difficult to develop efficient and effective solution procedures. Lagrangean relaxation in conjunction with a number of optimization-based heuristics are proposed to solve this problem. From the computational experiments, the proposed algorithms calculate credible solutions in minutes of CPU time for moderate problem sizes. Furthermore, compared with a number of sensible primal heuristics, the proposed algorithms clearly demonstrate uniform superiority in terms of solution quality. We also develop user-friendly GUI to make these algorithms easy to use.

Extended Summary

Computer network has become a strategic necessity to companies and even to countries. But how to design a usually sophisticated backbone network with the minimum deployment and operation cost subject to various and often stringent QoS requirements is a common challenge faced by network designers and managers. Intensive research has been conducted to address this issue. However, most research tackles this backbone network design problem without considering a full set of QoS requirements, including average delay, end-to-end delay and network reliability constraints.

B. Gavish [3] modeled the network topological design problem as a nonlinear combinatorial optimization problem. The objective is to minimize the network installation cost and the queueing cost imposed on the network users. However, network installation cost and the queueing cost are two different concepts such that it is not appropriate to put them together in the objective function. Hence, it is more reasonable to place the queueing cost or delay in the constraint set rather than in the objective function. In addition, end-to-end QoS constraints are not considered in [3].

N. G. Chattopadhyay [4] proposed the algorithm which combines a branch and bound

¹ Correspondence Author

method with the Ford-Fulkerson algorithm to solve the minimum cost backbone network design problem. The objective function is to minimize the fixed cost and the variable cost for setting up the communication lines in order to satisfy the end-to-end delay constraints. Hence, the cost for network access point (e.g. switch) is not considered in [4]. Furthermore, system performance constraint (e.g. average delay constraint) and reliability constraint (e.g. node disjoint path constraint) are not considered in [4].

A bunch of heuristics based on genetic algorithms (GA) and Tabu-Search algorithms are proposed to tackle the backbone network design problem [5, 6, 8, 9]. S. T. Cheng [5] proposes GA to find a network topology for a set of nodes whose total link cost is minimized, subject to the condition that the backbone network can accommodate 1 link failure. B. Ombuki [9] propose the GA to minimize the total link connection cost in backbone network design with the 3-connectivity constraints. In a similar study, A. Konak [8] proposed GA to minimize the backbone link installation cost with end-to-end delay and reliability (K node connectivity) constraints. S. Pierre [6] proposes heuristic algorithm based on Tabu-Search algorithm to minimize the computer topological network cost with considering average delay and reliability constraint. However, in [5, 6, 8, 9], the cost for network access point is not considered in the network installation cost. Furthermore, average and end-to-end delay constraints and reliability constraint are not jointly considered.

K. S. Song [7] formulated the link capacity problem with the end-to-end delay and cell loss rate constraints and solved by heuristic algorithm. This heuristic algorithm first determined the network topology based on the end-to-end delay requirement and then assigned the link capacity and network flow to limit the link utilization. However, a more rigorous approach should be taken to deal with this complex problem. And the network access point cost is also not considered in [7].

As a matter of fact, to ensure QoS requirement is the most important task in modern network services, and it require sophisticated design of the routing and capacity management. The delay QoS requirement is crucial to modern application services (e.g. VOD, tele-conferencing). And in particular, Backbone network usually requires high availability, that is, redundant links and switching nodes are needed in case of failure. As a result, unlike more of previous research, the QoS requirement considered in this paper can be classified into two parts. The first part is the delay QoS (including the average end-to-end delay and end-to-end delay for each O-D pair) and the second part is the reliability and availability of services. On the other hand, the network construction cost includes both the switch and link installation cost to reflect the cost structure of real network. This problem is a well-known difficult NP-hard problem.

This QoS based backbone network design problem is modeled as the graph where the users and switches are depicted as nodes and the communication channels are depicted as arcs. We show the definition of the following notation.

L	the set of candidate local loop links and backbone links in the communication network.
W	the set of origin-destination (O-D) pairs in the network.
λ_w	the traffic requirement for each O-D pair $w \in W$.
\bar{C}_l	a capacity upper bound in the candidate capacity configurations for link $l \in L$.
P_w	a given set of simple directed paths from the origin to the destination of O-D pair w .
U_k	a set of potential incoming links to switch k .
δ_{pl}	the indicator function which is one if link l is on path p and zero otherwise.
\mathcal{E}_{pk}	the indicator function which is one if switch k is on path p and zero otherwise.
g_l	the aggregate flow over link $l \in L$, which is $\sum_{p \in P_w} \sum_{w \in W} x_p \lambda_w \delta_{pl}$.
D_w	the maximum allowable end-to-end delay requirement for O-D pair w .
K	the maximum allowable average end-to-end delay requirement.

T_w	the minimum number of node disjoint paths required for O-D pair w .
O	the set of candidate locations for switches.
H	the set of link pairs that are with the same end points but in opposite directions.
A_l	the set of candidate capacity configurations for link l .
R_k	the set of admissible switching fabric configurations for switch at location k .
E_k	the set of candidate port configurations for switch at location k .
$\varphi_l(C_l)$	the cost for installing capacity C_l on link l , including the fixed and variable cost.
ξ_l	the fixed link installation cost for link l .
$Q_k(J_k, S_k)$	the cost for installing a switch at location k with switching fabric capacity J_k and number of ports S_k .
$F_l(f_l, C_l)$	the average delay on link $l \in L$, which is a function of f_l and C_l .
$B_l(f_l, C_l)$	the average number of packets on link $l \in L$, which is a function of f_l and C_l , and by the Little's results, which is equal to $\lambda_w * F_l(f_l, C_l)$.

And the decision variables are depicted as follows.

x_p	1 when path $p \in P_w$ is used to transmit the packets for O-D pair $w \in W$ and 0 otherwise.
z_p	1 when path $p \in P_w$ is the node disjoint path for O-D pair $w \in W$ and 0 otherwise.
y_{wl}	1 when link $l \in L$ is on the path chosen for O-D pair $w \in W$ and 0 otherwise.
f_l	the estimated aggregate flow on link $l \in L$.
M_l	1 when a link is installed at location $l \in L$ and 0 otherwise.
C_l	the capacity assignment for link $l \in L$.
J_k	the switching fabric capacity assignment for switch at location k .
S_k	the number of ports for switch at location k .

To determine the optimal network topology with consideration of the users' end-to-end QoS requirement and system QoS requirement, which is a NP-hard problem, is formulated as a nonlinear and nonconvex combinatorial optimization problem, as shown below.

$$\min Z_{IP} = \sum_{l \in L} \varphi_l(C_l) + \sum_{k \in O} Q_k(J_k, S_k) - \sum_{(l,l) \in H} \xi_l M_l M_{\bar{l}} \quad (\text{IP})$$

subject to:

$$\sum_{w \in W} \lambda_w \sum_{l \in L} B_l(f_l, C_l) \leq K \quad (1.1)$$

$$\sum_{l \in L} y_{wl} F_l(f_l, C_l) \leq D_w \quad \forall w \in W \quad (1.2)$$

$$\sum_{p \in P_w} x_p = 1 \quad \forall w \in W \quad (1.3)$$

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W \quad (1.4)$$

$$\sum_{p \in P_w} x_p \delta_{pl} \leq y_{wl} \quad \forall w \in W, l \in L \quad (1.5)$$

$$y_{wl} = 0 \text{ or } 1 \quad \forall w \in W, l \in L \quad (1.6)$$

$$g_l = \sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pl} \lambda_w \leq f_l \quad \forall l \in L \quad (1.7)$$

$$f_l \leq C_l \quad \forall l \in L \quad (1.8)$$

$$C_l \in A_l \quad \forall l \in L \quad (1.9)$$

$$M_l = 0 \text{ or } 1 \quad \forall l \in L \quad (1.10)$$

$$C_l \leq \bar{C}_l M_l \quad \forall l \in L \quad (1.11)$$

$$J_k \in R_k \quad \forall k \in O \quad (1.12)$$

$$\sum_{l \in U_k} M_l \leq S_k \quad \forall k \in O \quad (1.13)$$

$$S_k \in E_k \quad \forall k \in O \quad (1.14)$$

$$\sum_{w \in W} \sum_{p \in P_w} x_p \varepsilon_{pk} \lambda_w \leq J_k \quad \forall k \in O \quad (1.15)$$

$$\sum_{p \in P_w} z_p = T_w \quad \forall w \in W \quad (1.16)$$

$$\sum_{p \in P_w} z_p \delta_{pl} \leq M_l \quad \forall w \in W, l \in L \quad (1.17)$$

$$z_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W. \quad (1.18)$$

The objective is to minimize the link and switch installation cost. There are three terms in the objective function. The first term is to compute the total link installation cost, including the fixed cost and the variable cost. The second term is to compute the total switch installation cost. The third term is to compute one fixed cost for each installed opposite links. The necessity of subtracting the third term is to ensure that only one rather than two fixed cost is calculated for two links with the same attached nodes but in opposite direction. Constraint (1.1) requires that the average end to end packet delay should be no larger than maximum allowable average end-to-end delay requirement for all users. Constraint (1.2) requires that the end-to-end packet delay should be no larger than predetermined end-to-end delay requirement for each O-D pair. Constraints (1.3) and (1.4) require that the all the traffic for each O-D pair should be transmitted over exactly one path. The decision variable y_{wl} in Constraint (1.6) is an auxiliary decision variable, which is equal to $\sum_{p \in P_w} x_p \delta_{pl}$. Hence, the equality in Constraint (1.5) is replaced by inequality due to the

ease use of the Lagrangean relaxation. Constraints (1.7) and (1.8) are the link capacity constraints, which means the aggregate flow on the link should not exceed the link capacity. Constraint (1.9) determines the possible capacity configurations of all links. Constraints (1.10) and (1.11) require that the link must be installed first before link capacity assignment. Constraints (1.12) and (1.14) determine the possible switching fabric and number of ports of all switches. Constraint (1.13) is the switch termination constraint, which means the number of incoming links to the switch must not exceed the number of ports on that switch. Constraint (1.15) is the switch capacity constraint, which means the total flow incoming to any switch cannot exceed its switching fabric. Constraints (1.16) and (1.18) are the path diversity (node disjoint) requirement for each O-D pair. In other words, Constraints (1.16) and (1.18) denote the node and link fault tolerance constraint. Constraint (1.17) guarantees that link must be installed first before it could be adopted on the node disjoint path for each O-D pair.

The algorithm development is based upon the Lagrangean relaxation. We dualize Constraints (1.1), (1.2), (1.5), (1.7), (1.8), (1.11), (1.13), (1.15) and (1.17) of Problem (IP) to get the following Lagrangean relaxation problem (LR).

$$\begin{aligned} \min Z_D(a, b, c, d, h, e, m, n, q) \\ = \sum_{l \in L} \varphi_l(C_l) + \sum_{k \in O} Q_k(J_k, S_k) - \sum_{(l,j) \in H} \xi_l M_l M_j + a \left[\frac{1}{\sum_{w \in W} \lambda_w} \sum_{l \in L} B_l(f_l, C_l) - K \right] + \sum_{w \in W} b_w \left[\sum_{l \in L} y_{wl} F_l(f_l, C_l) - D_w \right] \\ + \sum_{w \in W} \sum_{l \in L} c_{wl} \left[\sum_{p \in P_w} x_p \delta_{pl} - y_{wl} \right] + \sum_{l \in L} d_l \left[\sum_{w \in W} \sum_{p \in P_w} x_p \delta_{pl} \lambda_w - f_l \right] + \end{aligned}$$

$$\begin{aligned}
& \sum_{l \in L} h_l (f_l - C_l) + \sum_{l \in L} e_l [C_l - \bar{C}_l M_l] + \sum_{k \in O} m_k [\sum_{l \in U_k} M_l - S_k] + \sum_{k \in O} n_k [\sum_{w \in W} \sum_{p \in P_w} x_p \varepsilon_{pk} \lambda_w - J_k] + \\
& \sum_{w \in W} \sum_{l \in L} q_{wl} [\sum_{p \in P_w} z_p \delta_{pl} - M_l]
\end{aligned} \tag{LR}$$

subject to:

$$\sum_{p \in P_w} x_p = 1 \quad \forall w \in W \tag{2.1}$$

$$x_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W \tag{2.2}$$

$$y_{wl} = 0 \text{ or } 1 \quad \forall w \in W, l \in L \tag{2.3}$$

$$C_l \in A_l \quad \forall l \in L \tag{2.4}$$

$$M_l = 0 \text{ or } 1 \quad \forall l \in L \tag{2.5}$$

$$J_k \in R_k \quad \forall k \in O \tag{2.6}$$

$$S_k \in E_k \quad \forall k \in O \tag{2.7}$$

$$\sum_{p \in P_w} z_p = T_w \quad \forall w \in W \tag{2.8}$$

$$z_p = 0 \text{ or } 1 \quad \forall p \in P_w, w \in W. \tag{2.9}$$

We can decompose (LR) into five independent subproblems.

Subproblem 2-1: for x_p :

$$\min \sum_{w \in W} \sum_{p \in P_w} [\sum_{l \in L} (c_{wl} + d_l \lambda_w) x_p \delta_{pl} + \sum_{k \in O} n_k \lambda_w x_p \varepsilon_{pk}]$$

subject to: (2.1) and (2.2).

Subproblem 2-2 : for C_l, y_{wl} and f_l :

$$\begin{aligned}
\min \quad & \sum_{l \in L} \varphi_l(C_l) + a \sum_{w \in W} \lambda_w \sum_{l \in L} B_l(f_l, C_l) + \sum_{w \in W} b_w \sum_{l \in L} y_{wl} F_l(f_l, C_l) - \sum_{w \in W} \sum_{l \in L} c_{wl} y_{wl} - \sum_{l \in L} d_l f_l \\
& + \sum_{l \in L} e_l C_l + \sum_{l \in L} h_l f_l - \sum_{l \in L} h_l C_l
\end{aligned}$$

subject to: (2.3) and (2.4).

Subproblem 2-3: for M_l :

$$\min - \sum_{l \in L} e_l \bar{C}_l M_l + \sum_{k \in O} \sum_{l \in U_k} m_k M_l - \sum_{(l, \bar{l}) \in H} \zeta_l M_l M_{\bar{l}} - \sum_{w \in W} \sum_{l \in L} q_{wl} M_l$$

subject to: (2.5).

Subproblem 2-4: for J_k and S_k :

$$\min \sum_{k \in O} [Q_k(J_k, S_k) - n_k J_k - m_k S_k]$$

subject to: (2.6) and (2.7).

Subproblem 2-5: for z_p :

$$\min \sum_{w \in W} \sum_{l \in L} q_{wl} \sum_{p \in P_w} z_p \delta_{pl}$$

subject to: (2.8) and (2.9).

In order to deal with the nodal weight of Subproblem 2-1, the node spitting technique [1] is

used. As a result, Subproblem 2-1 could be further decomposed into $|W|$ independent shortest path problem with nonnegative arc weights. It can be easily solved by the Dijkstra's algorithm. Subproblem 2-2 could also be decomposed into $|L|$ independent subproblems. For each link $l \in L$,

Subproblem 2-2.1: for C_l, y_{wl} and f_l :

$$\min \varphi_l(C_l) + a \sum_{w \in W} \frac{1}{\lambda_w} B_l(f_l, C_l) + \sum_{w \in W} b_w y_{wl} F_l(f_l, C_l) - \sum_{w \in W} c_{wl} y_{wl} - d_l f_l + e_l C_l + h_l f_l - h_l C_l$$

subject to: $y_{wl} = 0$ or $1 \quad \forall w \in W$ and $C_l \in A_l$.

Subproblem 2-2.1 is a complicated problem due to the coupling of three decision variables: C_l, y_{wl} and f_l . Since the possible capacity configurations of links are finite, such as 64kbps, 128kbps, 256kbps, 512kbps, T1 and T3 for example. We can exhaustive search all different possible link configuration by finding the best y_{wl} and f_l . In [2], Frank Y. S. Lin proposed an efficient algorithm to solve y_{wl} and f_l at a given link capacity under $M/M/1$ queuing model. Therefore, the algorithm to solve Subproblem 2-2.1 under $M/M/1$ queuing model is proposed as bellow. It is remarkable to address that the formulation could be extended to any non $M/M/1$ model with monotonically increasing and convexity performance metrics.

Step 1. For each possible link capacity configuration, applying the algorithm developed in [2] to solve Subproblem 2.1 as to find the optimal y_{wl} and f_l .

Step 2. Finding the minimum objective value of Subproblem 2.1 from the objective value associated with each possible link capacity configuration. Then y_{wl} and f_l can be determined from the optimal link capacity.

Subproblem 2-3 can be decomposed into $|H|$ independent subproblems. For each pair of bi-directional links $(l, \bar{l}) \in H$,

Subproblem 2-3-1: for M_l and $M_{\bar{l}}$:

$$\min -e_l \bar{C}_l M_l - e_{\bar{l}} \bar{C}_{\bar{l}} M_{\bar{l}} + G_1 M_l + G_2 M_{\bar{l}} - \xi_l M_l M_{\bar{l}} - \sum_{w \in W} q_{wl} M_l - \sum_{w \in W} q_{w\bar{l}} M_{\bar{l}}$$

subject to: $M_l = 0$ or 1 and $M_{\bar{l}} = 0$ or 1 .

In the above formulation, the G_1 and G_2 are calculated as follows.

1. If the link l is the incoming link to any potential switch, say k_1 , then assign G_1 to m_{k_1} , else assign G_1 to zero.
2. If the link \bar{l} is the incoming link to any potential switch, say k_2 , then assign G_2 to m_{k_2} , else assign G_2 to zero.

From the formulation of Subproblem 2-3-1, two opposite direction links are considered at the same time. As a result, the algorithm to optimally solve Subproblem 2-3-1 is proposed as follows.

Step 1. Let $N_1 = 0, N_2 = -e_l \bar{C}_l + G_1 - \sum_{w \in W} q_{wl}, N_3 = -e_{\bar{l}} \bar{C}_{\bar{l}} + G_2 - \sum_{w \in W} q_{w\bar{l}},$

$$N_4 = -e_l \bar{C}_l + G_1 - e_{\bar{l}} \bar{C}_{\bar{l}} + G_2 - \xi_l - \sum_{w \in W} q_{wl} - \sum_{w \in W} q_{w\bar{l}}.$$

Step 2. Identify the N_i with the minimum value, where $i = 1, 2, 3, 4$.

Step 3. If $i = 1$, then assign $M_l = 0$ and $M_{\bar{l}} = 0$, else if $i = 2$, then assign $M_l = 1$ and $M_{\bar{l}} = 0$, else

if $i = 3$, then assign $M_l = 0$ and $M_j = 1$, else if $i = 4$, then assign $M_l = 1$ and $M_j = 1$.

Subproblem 2-4 can be further decomposed into $|O|$ independent subproblems. For each independent subproblem, due to the number of possible switch configurations (including number of ports and switching fabric) is finite and manageable within computational time, we can exhaustively search all possible combination of switch configurations as to find the optimal J_k and S_k .

Subproblem 2-5 can be further decomposed into $|W|$ independent node disjoint shortest path problem with nonnegative arc weights. Suurballe [10] propose an efficient algorithm to optimally solve link disjoint path problem. Hence, Subproblem 2-5 could be optimally solved by the Suurballe's algorithms in conjunction with the node splitting technique.

According to the algorithms developed above to solve each subproblem, we could successfully solve the Lagrangean relaxation problem optimally. By using the weak Lagrangean duality theorem (for any given set of non-negative multipliers, the optimal objective function value of the corresponding Lagrangean relaxation problem is a lower bound on the optimal objective function value of the primal problem), $Z_D(a, b, c, d, h, e, m, n, q)$ is a lower bound on Z_{IP} . We could construct the dual problem to calculate the tightest lower bound and solve the dual problem by using the subgradient method.

To obtain the primal solutions to the Z_{IP} , solutions to the Lagrangean relaxation problems (LR) is considered. We develop sophisticated getting primal heuristics to getting the primal feasible solutions. According to the computational results, this primal heuristic can get a good primal feasible solution. This getting primal heuristic start with the routing assignment obtained from the Subproblem 2-2.1. From the routing assignment in Subproblem 2-2.1, the aggregate flow on each link can be calculated. In order to satisfy the end-to-end delay requirement for each O-D pair, the tightest end-to-end delay for all O-D pairs is located by searching the minimum end-to-end delay requirement among all O-D pairs. From the tightest end-to-end delay, the tightest link delay can be calculated by dividing the tightest end-to-end delay to the maximum hop number in any routing path. The maximum hop number for any O-D pair is equal to the number of potential switches plus one, since the source node must home to the switch first, and then route to the other switches, and finally route to the destination node. From the tightest link delay, we can determine the minimum link capacity in order to satisfy the tightest link delay requirement. From the above statement, we can see that the upper bound are overestimated, since the link delay occurs in any link must be at least as the tightest link delay.

In order to satisfy the node disjoint requirement for each O-D pair, the node disjoint path assignment from Subproblem 2-5 is used. If the associated link on any node disjoint path did not install at the above procedure, the minimum nonzero capacity is installed on that link. After the link capacity is determined, the number of links incoming to each potential switch can be determined. Also from the aggregate flow on each link, the total aggregate flow incoming to each potential switch can also be determined. As a result, the minimum cost switch configuration in order to satisfy the number of ports and switch fabric constraints can be determined as well.

In order to make the above algorithms to be easily used by telecommunication industry. We have developed the software package which bundles the network planning algorithms developed above with the user-friendly graphical user interface (GUI). The network planning algorithms are coded in C and performed at PC with INTEL™ PIII-500 CPU. The GUI are written with Microsoft Visual Basic and the running platform is on the Microsoft NT or Window 2000.

In this package, the input parameters include the locations for the users and the potential switches, admissible configurations and cost structures of potential switches and links, traffic requirements and survivability/connectivity requirements. And the output parameters include the switch and link configuration assignment, routing assignment, node disjoint paths assignment, average end-to-end delay and individual end-to-end delay for each O-D pair.

The maximum number of iterations for the proposed dual Lagrangean algorithm developed above are 1000, and the improvement counter is 30. The step size for the dual Lagrangean algorithm is initialized to be 2 and be halved of its value when the objective value of the dual algorithm does not improve for 30 iterations.

Two sets of computational experiments are performed. The computational time for these two sets of computational experiments are all within fifteen minutes under the network size of 30 user/switch nodes and 300 potential links. Hence, the proposed algorithms are efficient in time complexity.

In these computational experiments, the cost of the link assignment is divided into two parts, fixed cost and variable cost. The fixed cost is calculated from the Euclidean distance between two end points that the link connected, and the variable cost is based on the link capacity configuration. There are fifteen discrete potential link capacity configurations, from 0 to 500, for the computational experiments. And the cost associated with these potential capacity configurations is a concave function to reflect the economy-of-scale effect. On the other hand, the switch installation cost is based upon two switch configurations, switching fabric and the number of ports on the switch. There are nineteen discrete potential switch configurations in the computational experiments. And the cost associated with these potential switch configurations is also a concave function to reflect the economy-of-scale effect.

In the first set of computational experiment, we want to test the solution quality when the input delay requirements are loose as compared to the output of the delay requirement. And node disjoint path requirement is not considered. In the second set of computational experiment, we want to test the solution quality when the input delay requirements are tight as compared to the output of the delay requirement. And the 2-connected node-disjoint-path requirement for each O-D pair is considered. In order to measure the solution quality of our proposed algorithms, a number of sensible primal heuristics were proposed. The cost for backbone network installation cost, as computed by the proposed algorithm, could save up to 50% as compared to these primal heuristics. Based on the solution quality and the computational time, the proposed algorithm based on the mathematical formulation could provide effective and efficient solutions.

References

- [1] F. Y.-S. Lin, "Link Set Sizing for Networks Supporting SMDS", IEEE/ACM Trans. on Networking, pp. 729-739, Vol. 1, No. 6, Dec. 1993.
- [2] K. T. Cheng and F. Y.-S. Lin, "Minimax End-to-End Delay Routing and Capacity Assignment for Virtual Circuit Networks," Proc. IEEE Globecom, pp. 2134-2138, 1995.
- [3] B. Gavish, "Topological design of computer communication networks – The overall design problem", European Journal of Operational Research, 58, pp. 149-172, 1992.
- [4] N. G. Chattopadhyay, T. W. Morgan and A. Raghuram, "An Innovative Technique for Backbone Network Design", IEEE Trans. on System, Man, and Cybernetics, Vol. 19, No. 5, pp. 1122-1132, Sep./Oct., 1989.
- [5] S. T. Cheng, "Topological Optimization of a Reliable Communication Network", IEEE Trans. on Reliability, Vol. 47, No. 3, pp.225-233, Sep. 1998.
- [6] S. Pierre and A. Elgibaoui, "A Tabu-Search Approach for Designing Computer-Network Topologies with Unreliable Components", IEEE Trans. on Reliability, Vol. 46, No. 3, pp. 350-359, Sep. 1997.
- [7] K. S. Song and A. K. Somani, "Fault Tolerant ATM Backbone Network Design Considering Cell Loss Rates and End-to-End Delay Constraints", IEEE 13th Annual International Phoenix Conference on Computers and Communications, pp. 119-126, April 1994.
- [8] A. Konak and A. E. Smith, "A Hybrid Genetic Algorithm Approach for Backbone Design of Communication Networks", Proc. of the Congress on Evolutionary Computation, Vol. 3, pp. 1817-1823, 1999.

- [9] B. Ombuki, M. Nakamura, Z. Nakao and K. Onaga, "Evolutionary Computation for Topological Optimization of 3-Connected Computer Networks", IEEE International Conference on Systems, Man, and Cybernetics, Vol. 1, pp. 659-664, 1999.
- [10] J. W. Suurballe and R. E. Tarjan, "A Quick Method for Finding Shortest Pairs of Disjoint Paths", Networks, Vol. 14, pp. 325-336, 1984.