Comparison of a Genetic Algorithm with a Simulated Annealing Algorithm for the Design of an ATM Network

Dale R. Thompson, *Member, IEEE*, and Griff L. Bilbro, *Senior Member, IEEE*

***Abstract*—The genetic algorithm (GA) and simulated annealing algorithm (SA) are empirically compared for the problem of optimizing the topological design of a network. In addition to the usual problem of optimizing only the placement of links, in this letter the number and placement of concentrators are also decision variables for a class of problems using a real set of concentrators, links, and traffic. The average GA solution cost less than the average SA solution.**

***Index Terms*—Asynchronous transfer mode, genetic algorithm, networks, simulated annealing, topology.**

I. Introduction[1]

THE topological design of a computer network specifies a low-cost network topology that satisfies traffic quality of service (QoS) constraints [1]. We characterize QoS as acceptable buffer overflow probability [2]. A network is a set of terminals and concentrators that exchange traffic over links. A terminal is a source and destination of traffic. A concentrator is any switch, hub, or router and is characterized by maximum traffic capacity, number of ports, the traffic capacity of each port, buffer size, and cost. A link is a connection between a terminal and a concentrator or between concentrators and is characterized by cost and maximum traffic capacity [3]. In a network, terminals exchange traffic with other terminals, possibly via concentrators. Traffic is a measure of the amount of information exchanged per unit of time, i.e., cells per second, and is characterized by effective bandwidth [2]. The route of the traffic is the set of links the traffic uses to reach the destination. The concentrators are connected to each other to form a backbone network. Links between concentrators may be redundant [4].

The intractability and importance of topological design has attracted heuristics [3], [4], including genetic algorithm (GA) [5]–[7] and simulated annealing (SA) [1], [8] and new algorithms are commonly compared with SA [1], [5]. Topological design of a computer network is usually formulated as the placement of links between given concentrator locations [1], [5], because the monthly cost of leasing links between locations dominates the cost of concentrators. Little attention has been given to optimizing the number and placement of concentrators [3], [4], [9]. We focus on comparing GA with the well-understood SA for the concentrator location problem. The concentrator location problem determines the number and location of concentrators, how to connect them, and the terminals to connect to which concentrators [4]. This work was motivated by upgrading a campus network to an asynchronous transfer mode (ATM) network that had an existing fiber optic physical plant. Therefore, the concentrator costs dominate.

II. Statement of the Problem

We formulate the concentrator location problem using Gavish's terminology [3]. We refer to each unique origin-destination pair of traffic as a commodity and label it with index p. We define Π to be the index set of all commodities. Let S_p be the subset of routes that are candidates to support commodity p. Let R be the index set of all candidate routes in the network, $R = U_{p\in\Pi} S_p$. Let I be the index set of feasible concentrator locations. The set of terminal locations K is a given constant. We define L_r as the index set of links used by route r and L as the index set of candidate links. A link is defined by its end points and multiple links can exist between two locations. In particular, a two-pair fiber cable between two locations may connect both a terminal and a concentrator as well as a concentrator to another concentrator.

Let Q_{ij} for $i \neq j$ be the maximum capacity of link (i, j) in cells per second. Let Q_{ii} be the maximum capacity of the concentrator at site i. Each commodity p has a set of traffic parameters Γ_p which include peak rate, average rate, burst size, and maximum acceptable loss probability. Any such Γ_p defines an effective bandwidth which we use to size link capacities.

We define three decision variables. Let Z_i be equal to one if a concentrator is assigned at location i, and zero otherwise. Let Y_{ij} be equal to one if a link exists between locations i and j, and zero otherwise. Let X_r be equal to one if route r is selected to support the appropriate commodity, and zero otherwise.

The traffic flow F_{ij} in cells per second on link (i, j) is a function of each commodities' Γ_p that traverse the link. Let Γ_r be the set of Γ_p's of the appropriate commodity that is supported by route r. Let δ_{ij}^r be one if r supports the appropriate commodity and uses link (i, j), and zero otherwise. Therefore,

Manuscript received January 12, 2000. The associate editor coordinating the review of this letter and approving it for publication was Prof. N. Shroff.

D. R. Thompson is with the Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville, AR 72701 USA.

G. L. Bilbro is with the Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, NC 27695 USA.

Publisher Item Identifier S 1089-7798(00)07667-5.

[1]The tests described and the resulting data presented herein, unless otherwise noted, were obtained from research conducted under the Engineer Research and Development Center. Permission was granted by the Chief of Engineers to publish this information

F_{ij} is a function of the type of effective bandwidth algorithm $EBW(X_r \Gamma_r \delta_{ij}^r)$.

We define C_i as the cost of locating a concentrator at location i and includes hardware, software, site preparation, and maintenance costs. The total concentrator costs are $\sum_{i \in I} C_i Z_i$. We define S_{ij} as the link cost of connecting locations i and j and includes installation and maintenance. The total link costs are $\sum_{(i,j) \in L} S_{ij} Y_{ij}$.

The objective function of the concentrator location problem is to minimize the overall system costs

$$\Psi = \text{Min} \left\{ \sum_{i \in I} C_i Z_i + \sum_{(i,j) \in L} S_{ij} Y_{ij} \right\} \tag{1}$$

subject to contraints

$$Y_{ij} \le Z_i \quad \forall i \in I, (i, j) \in L \tag{2}$$
$$Y_{ij} \le Z_j \quad \forall j \in I, (i, j) \in L \tag{3}$$
$$X_r \le Y_{ij} \quad \forall (i, j) \in L_r \tag{4}$$
$$\sum_{r \in R} X_r = 1 \quad \forall p \in \Pi, \tag{5}$$
$$F_{ij} = EBW(X_r \Gamma_r \delta_{ij}^r) \quad \forall r \in R, (i, j) \in L \tag{6}$$
$$F_{ij} \le Q_{ij} \quad \forall (i, j) \in L \tag{7}$$
$$(F_{ij} + F_{ji}) Z_i \le Q_{ii} Z_i \quad \forall i \in I, (i, j) \in L \tag{8}$$
$$Z_i, X_r, Y_{ij} = 0 \text{ or } 1 \quad \forall i \in I, (i, j) \in L, r \in R. \tag{9}$$

We make the following assumptions. We define the least expensive path as the least expensive set of link costs S_{ij} that establishes a physical path between locations i and j. We use the least expensive path to assign terminals to concentrators, to assign the links between concentrators, and to route commodities. If we were solving a network in which link costs dominated, we would adjust the routes to take capacity packing into consideration. We restrict $Q_{ij} = Q_{ji}$ so the maximum of F_{ij} and F_{ji} determines the required link capacity. For each given concentrator location, we assign the largest capacity concentrator and apply a greedy algorithm to find the least-cost concentrator that supports the traffic. Each concentrator may have multiple capacity ports so we assign links to ports by sorting the effective bandwidth of the links in ascending order and assigning them to the ports in ascending order of capacity. Given these assumptions, the problem is reduced to specifying the concentrator locations.

III. Genetic Algorithm

The initial GA population of size n was generated by using the n distinct least expensive solutions from a greedy-drop heuristic. The greedy-drop heuristic places the largest capacity concentrator at every candidate location. For each chosen location, it iteratively assigns concentrators in decreasing order of capacity until it finds the least-cost concentrator that will support the traffic. Then, it calculates each concentrator's cost per supported traffic and drops the most expensive concentrator. It repeats this process until the set of concentrators cannot support the given traffic.

We used a relatively small population size to speed convergence, a multipoint crossover operator to overcome the homogeneity of the small population [10], and a neighborhood search to refine the local minima [11]. Let max be the maximum number of generations, x be the average number of crossover points, p_c be the probability of crossover, and p_m be the probability of mutation. Note p_m is defined as the probability that a solution will be modified, not an individual variable. Therefore, $p_m = 1.0$ specifies that one or two variables in each solution will be modified.

Algorithm 1: Genetic Algorithm
Initialize population with n solutions
Evaluate fitness of each solution.
Save best solution.
Repeat max times{
  Select solutions for next generation using tournament selection of size s [12].
  After a specified number of generations, copy the best member n times, and delete exactly one distinct concentrator location from each solution.
  Pick two solutions with probability p_c and exchange genetic material with x-point segmented crossover operator.
  Mutate solution with probability p_m by randomly adding, deleting, or exchanging concentrator locations with probability 0.5, 0.25, and 0.25.
  Evaluate fitness of each solution.
  Replace worst solution of present generation with best solution found.}

IV. Simulated Annealing Algorithm

The initial feasible solution for SA is constructed by using the least expensive solution found with the greedy-drop heuristic. Let t be the temperature, t_0 be the initial temperature, max be the maximum number of iterations, and $nrep$ be the maximum number of modifications at a given t. Let s_c be the current solution and s_n be the new solution with costs $f(s_c)$ and $f(s_n)$. The function $random[0, 1]$ generates a random number between zero and one with a uniform distribution.

Algorithm 2: Simulated Annealing Algorithm
Generate $\boldsymbol{s}_c$
Evaluate $f(\boldsymbol{s}_c)$
$t \leftarrow t_0$
Repeat max times{
  Repeat $nrep$ times{
    Create $\boldsymbol{s}_n$ by randomly adding, deleting, or exchanging concentrator locations with probability 0.5, 0.25, and 0.25.
    Evaluate $f(\boldsymbol{s}_n)$
    if $f(\boldsymbol{s}_n) < f(\boldsymbol{s}_c)$, **then** $\boldsymbol{s}_c \leftarrow \boldsymbol{s}_n$
    else if $random[0, 1] < \exp\{(f(\boldsymbol{s}_c) - f(\boldsymbol{s}_n))/t\}$,
      then $\boldsymbol{s}_c \leftarrow \boldsymbol{s}_n$}
  $t \leftarrow ct, 0 < c < 1$}

TABLE I
AVAILABLE SWITCHES

Switch #	Cost $K	# 622 Mbps ports	# 155 Mbps ports
0	170	4	80
1	150	2	88
2	130	0	96
3	70	4	0
4	50	2	8
5	30	0	16

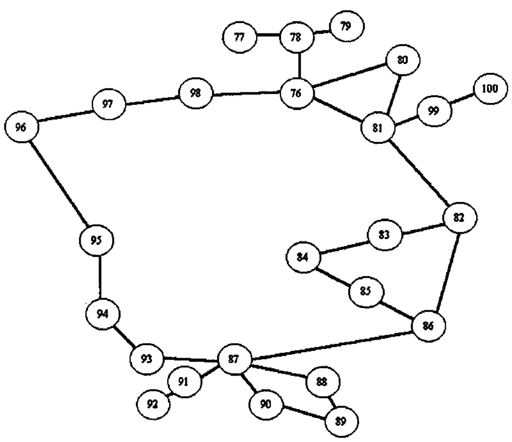

Fig. 1. Fiber topology

V. Performance Evaluation

Although we evaluate GA and SA on only one problem, it provides a realistic comparison because it uses available switches, fiber optic links, and traffic flows. The available ATM switches are shown in Table I and the fiber optic topology is shown in Fig. 1. The fiber physical topology supports 76 networks with 1423 different commodities. We set the cost per fiber at \$1.25/m. We set the maximum acceptable loss probability to 10^{-10} and use effective bandwidth to determine link capacities.

A single network evaluation requires approximately five minutes of computational time. Therefore, we restricted GA and SA to evaluate 100 sets of concentrator sites and compared the best solutions. We ran both algorithms ten times with distinct random number generator seeds. In SA, $t_o = 100\,000, c = 0.80$, $nrep = 10$, and $\max = 10$. In GA, $n = 20$, $\max = 4$, $p_c = 0.60$, $p_m = 1.0$, $x = 10$, $s = 4$, and the local search was performed after the fourth generation.

VI. Results and Conclusions

GA and SA provided better solutions than the least-cost greedy-drop heuristic solution of \$587K. The average GA solution (\$463K) cost 5% less than the average SA solution

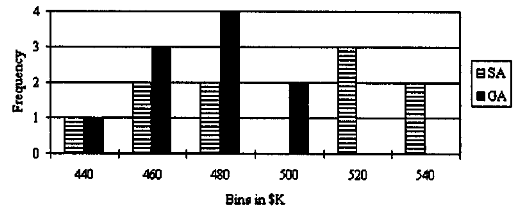

Fig. 2. Histogram of network topology costs.

(\$489K). A histogram of the costs with \$20Kbins is shown in Fig. 2. GA provides a tighter distribution of solutions than SA with a standard deviation of almost one-half SA. We propose that it is more probable that SA may discard potentially "good" solutions than GA because SA retains a single solution and GA retains a population of solutions.

Future upgrades of the network could be supported by GA. We could do a break-even analysis by rearranging the existing concentrators. For the initial population, we would copy the existing solution n times and then randomly swap used and unused locations maintaining the same number and types of concentrators. If these simulations did not support the new requirements, we would permit concentrator upgrades and additional concentrator locations.

References

[1] S. Pierre, "Inferring new design rules by machine learning: A case study of topological optimization," *IEEE Trans. Syst., Man, Cybern. A*, vol. 28, pp. 575–585, Sept. 1998.

[2] R. Guérin, H. Ahmadi, and M. Naghshineh, "Equivalent capacity and its application to bandwidth allocation in high-speed networks," *IEEE J. Select. Areas Commun.*, vol. 9, pp. 968–981, 1991.

[3] B. Gavish, "Topological design of computer communication networks-the overall design problem," *Eur. J. Oper. Res.*, vol. 58, pp. 149–172, 1992.

[4] R. R. Boorstyn and H. Frank, "Large-scale network topological optimization," *IEEE Trans. Commun.*, vol. COM-25, pp. 29–47, 1977.

[5] S. Pierre and G. Legault, "A genetic algorithm for designing distributed computer network topologies," *IEEE Trans. Syst., Man, Cybern. B*, vol. 28, pp. 249–258, Apr. 1998.

[6] L. Davis and S. Coombs, "Genetic algorithms and communication link speed design: Constraints and operators," in *Proc. 2nd Int. Conf. Genetic Algorithms*. Hillsdale, NJ: Lawrence Erlbaum, 1987, pp. 257–260.

[7] K. Tang, K. To, K. Man, and S. Kwong, "Topology design and bandwidth allocation of embedded ATM networks using genetic algorithm," *IEEE Commun. Lett.*, vol. 2, pp. 171–173, June 1998.

[8] B. Mukherjee, D. Banerjee, S. Ramamurthy, and A. Mukherjee, "Some principles for designing a wide-area WDM optical network," *IEEE/ACM Trans. Networking*, vol. 4, pp. 684–696, Oct. 1996.

[9] Z. Stanković and G. Petrović, "The design of packet switching network with the given initial graphical structure," in *Proc. 11th Mediterranean Electrotechnical Conf.*. Piscataway, NJ: IEEE Press, 1998, vol. 2, pp. 1333–1337.

[10] K. A. D. Jong and W. M. Spears, "An analysis of the interacting roles of population size and crossover in genetic algorithms," in *Int. Workshop Parallel Problem Solving from Nature*, Germany: Univ. of Dortmund, Oct. 1–3, 1990, pp. 38–47.

[11] C. R. Houck, J. A. Joines, M. G. Kay, and J. R. Wilson, "Empirical investigation of the benefits of partial Lamarckianism," *Evol. Comput.*, vol. 5, no. 1, pp. 31–60, 1997.

[12] T. Bäck, U. Hammel, and H. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 3–17, Apr. 1997.