Pergamon

0952-1976(94)00041-7

Engng Applic. Artif. Intell. Vol. 8, No. 1, pp. 61–69, 1995

0952-1976/95 $9.50 + 0.00

Contributed Paper

Topological Design of Computer Communication Networks Using Simulated Annealing

SAMUEL PIERRE
Université du Québec

MICHEL-ANGE HYPPOLITE
Université du Québec

JEAN-MARIE BOURJOLLY
Concordia University

OUMAR DIOUME
DTI Télecom, Montréal

(Received April 1994; in revised form July 1994)

This paper presents an application of the simulated annealing heuristic to the problem of designing computer communication networks. This problem essentially consists in finding the least-cost network topologies that satisfies a given set of performance and reliability constraints. The results of the computational experiments show that simulated annealing is a suitable approach for solving this very difficult combinatorial optimization problem, in the sense that it provides feasible and low-cost solutions within reasonable CPU times.

Keywords: Simulated annealing, topological design, communication network, knowledge-based system, heuristic method.

1. INTRODUCTION

The computer communication network design problem can be described as the process of generating network topologies that satisfy a given set of performance and reliability constraints at least cost. This problem is known in the literature as the topological design problem of distributed computer networks.[1–4]

This paper deals with packet-switched backbone networks. They are usually modeled by a graph $G = (N, E)$, where N is the set of switching nodes and E is a set of edges representing communication links between pairs of nodes.[5–9] More specifically, the nodes and edges respectively represent computers and full duplex communication links between pairs of computers (full duplex links allow information to be simultaneously carried in both directions). Each message is broken at the source node into ordered small blocks called *packets*, that are labeled with the destination address of the entire message. The resulting packets, instead of traveling along paths reserved in advance, adaptively find their way through the network independently of each other, in a store-and-forward fashion, until they reach the destination node where the original message is reassembled.

The problem is known to be "NP-hard".[10] It follows that it cannot be solved in polynomial time. Therefore, it is reasonable to look for a quick "good" feasible solution to this problem rather than attempting to solve it to optimality.

This paper presents an application of the simulated annealing method to the problem of designing computer network topologies. Simulated annealing is an optimization heuristic. It can be viewed as a generalization of local search in the sense that it occasionally moves in directions that temporarily deteriorate the value of the objective function, which sometimes allows the search to move away from a local optimum. Section 2 describes models, parameters, variables and constraints generally used in the context of computer network topological design, then analyzes conventional

Correspondence should be sent to: Dr S. Pierre, LICEF Laboratory, Télé-université, Université du Québec, Case Postale 5250, Succ. C, Montréal, Québec, Canada H2X 3M4.

approaches often used to solve this problem. Section 3 outlines intelligent heuristic approaches, then examines how simulated annealing can be adapted to the context of this specific problem. Section 4 discusses the results of computational experiments performed with the simulated annealing approach on various network topologies. Section 5 summarizes the most important issues of this paper and suggests new research directions.

2. TOPOLOGICAL DESIGN OF COMPUTER NETWORKS

An extended discussion of the network topological design problem is beyond the scope of this paper. Several aspects and considerations of this problem have been reported in Refs 9–13. In this paper, the focus is on the backbone topological design problem which can be stated as follows:[14]

Let $G=(N, E)$, $|N|=n$ and $|E|=m$.

Given:

(1) the n node locations
(2) the traffic requirements between every node pair
(3) the technological constraints (capacity limits, the discreteness of the capacity values, etc.)
(4) the cost elements (line tariff structures, switch costs, etc.)
(5) an upper limit on the average packet delay in the network
(6) the reliability requirements

Objective:

Find a least-cost network such that the traffic, delay, and reliabilty requirements are satisfied within the technological constraints

Over:

(1) all possible topological configurations
(2) all possible capacity assignments
(3) all possible flow assignments and routings

Denote by γ_{ij} $(i, j=1, 2, \ldots, m, i \neq j)$ the number of packets per second exchanged between nodes i and j, called the *end-to-end traffic* of the node pair (i, j); denote by γ the *total traffic* of the network. The traffic requirements can be represented by a traffic matrix $\Gamma=(\gamma_{ij})$. The estimation of Γ is generally inaccurate *a priori* because, over time, the true value of γ_{ij} depends on network parameters such as the allocation of resources to computers, the demand for resources and so on, which are difficult to forecast and are subject to changes with time and network growth.[1]

Each link k is characterized by two typical attributes: flow f_k, defined as the effective data rate on link k (in bits/second or bps), and capacity C_k representing an upper limit on the rate of data transmission along link k (in bps). The flow vector $f=(f_k)$ is uniquely determined by the traffic matrix, and by the routing strategy.[15,16] The capacity vector $C=(C_k)$ must be large enough to support the flow, i.e. $f_k \leq C_k$. Because of technological constraints, the value of C_k must be calculated as a combination of basic capacity units chosen from among a finite set of available commercial options: 1.2, 2.4, 4.8, 19.2, 56.0, 100 kbps (kilobit/second), etc.

The *average packet delay* T is the mean time that packets take to travel from origin to destination in the network. It can be expressed as follows:[1,3]

$$T=\frac{1}{\gamma}\sum_{k=1}^{n(n-1)/2}\frac{f_k}{(C_k-f_k)}. \tag{1}$$

The average packet delay is the most useful measure to evaluate the performance of the communication network. The upper limit on the average packet delay in the network is denoted by $T_{\max}$.

The reliability of a computer network can be stated in a number of ways. It can be defined as the probability of having a successful communication between every pair of nodes.[17,18] It is often associated with the concept of *graph connectivity* which refers to the fact that there is at least one path or route between any two nodes of the network.[17–20] In fact, for connecting two nodes, a path is required whenever there is no direct edge linking them. The paths between two nodes can be edge-disjoint or node-disjoint. Two paths between source i and destination j are said to be *edge-disjoint* if and only if they do not share any edge. Two paths between source i and destination j are said to be *node-disjoint* if and only if they do not share any node, except i and j considered as origin and destination. *Edge-connectivity* can be defined as the minimum number K of edge-disjoint paths over all node pairs, whereas *node-connectivity* measures the minimum number of node-disjoint paths over all node pairs. In order to satisfy high reliability requirements, a node-connectivity greater than 2 is recommended, i.e. K-node-connectivity, with $K>2$.[14,21,22]

The risk of combinatorial explosion related to blindly selecting links of topological configurations and to assigning capacities to these links, the nonlinearity of relevant functions such as network cost and average packet delay T, the discreteness of link capacities available on the telecommunication market, are examples of the difficulties related to this combinatorial optimization problem. It follows that in practice the topological backbone design problem can only be solved by heuristic methods that severely reduce the search space of candidate topologies.

Most conventional design methods are essentially simple-minded local search heuristics that work as follows. An initial feasible network topology is devised by some *ad hoc* procedure and, subsequently, the current topology undergoes a small modification and is updated if the effect of that modification is to lower the cost of the network without making it infeasible; the process is terminated when the first local minimum is reached.

Once the topological configuration and the capacities are determined, the flow deviation method can be used to compute the link flows.[23]

A variety of flow deviation algorithms have been developed to iterate between capacity assignment and flow assignment until a local minimum is reached.[14] The branch-exchange,[24] concave branch elimination[25] and cut-saturation[21] algorithms, and the method reported in Ref. 26, are examples of such optimization techniques which are search procedures aiming at optimizing the network structure by sequentially modifying small sections of a larger network.

3. INTELLIGENT HEURISTIC APPROACHES

Artificial intelligence and knowledge-based systems are characterized by their ability to represent and manipulate heuristic knowledge that can be used to solve complex decision problems. There exists significant works which adopt a hybrid strategy, combining powerful optimization algorithms and flexible heuristic procedures for solving difficult combinatorial optimization problems. For instance, Dutta and Mitra[14] have developed a method for integrating heuristic design knowledge with optimization models, to create a tool for the topological design of computer communication networks. In the same vein, Samoylenko[27] has applied heuristic problem-solving methods based on artificial intelligence techniques to computer communication network design. Sykes and White[28] have presented the conceptualization of, and specifications for, a knowledge system to assist an experienced network designer in the design of packet-switched data network backbone topologies.

On the other hand, a new approach based on the artificial intelligence paradigm has recently been developed for the topological design of computer networks. It consists in using a knowledge-based system which essentially attempts to make progressive improvements on a starting topology. This approach can be summarized as follows: an initial topology is generated by a module called an *initial topology generator* from data specified by the user. This topology is fed into another module called an *example generator* which then applies a sequence of perturbation cycles to it, in order to produce examples, i.e. topologies satisfying a given set of performance and reliability constraints. To improve the solution resulting from the perturbation cycles, a learning cycle may be started: through this process, the system mainly tries to infer new perturbation rules from the set of generated examples. The implementation of this approach results in an intelligent system, called SIDRO, capable of auto-modifying its initial knowledge by gaining additional knowledge through inductive rules. More details on this system can be found in Refs 29 and 30.

As mentioned in Ref. 29, this approach can be considered as an attempt to generalize a local search. In this sense, it is similar to heuristic methods such as the simulated annealing and the genetic algorithms. In the sequel, this paper will focus on the application of simulated annealing to designing communication network topologies.

3.1. Simulated annealing method

Simulated annealing is a search procedure in which the current solution is continually compared to configurations that are "close" to it, i.e. that can be obtained by carrying out a small perturbation.[31,32] If a perturbation results in an improved solution, it is accepted and the current solution is updated accordingly; otherwise it has a probability of being accepted which is close to 1 at the beginning, but decreases and converges to zero as the search progresses. By accepting an occasional worsening of the current solution, one avoids being trapped too early in a local optimum. On the other hand, decreasing the probability of accepting a worsening perturbation guarantees that the algorithm will eventually converge and will be less likely to move away from a global optimum after having approached it. The algorithm stops with a local optimum and the solution provided is the best among all configurations found by the algorithm. In the search for as good a solution as possible, one may execute the simulated annealing algorithm a large number of times with various initial solutions. Figure 1 presents a brief description of the simluated annealing algorithm in pseudo-PASCAL.[33]

The name of this algorithm comes from an analogy between its behavior and that of a well-known physical process of thermodynamics and metallurgy, called annealing, by which a metal is first melted at a very high temperature and then slowly cooled until it solidifies; when cooled slowly enough, it tends to solidify in a structure of minimal energy.[34] It is customary for simulated annealing practitioners to use the word *temperature* to designate the parameter θ that controls the probability of accepting a worsening perturbation over time. At the beginning, θ is set to a very high value; later it is multiplied by a factor $\alpha (0<\alpha<1)$, called the *cooling rate* or the *annealing factor*, after every L iterations or trials. L represents the length of a plateau, and the rule according to which θ is defined is called a *cooling schedule*.

3.2. Topological design by simulated annealing

In the context of topological design of packet-switched computer networks, simulated annealing starts with an initial topological configuration chosen to be K-connected, with $K=2, \ldots, 5$, depending on the desired level of reliability. At each subsequent step, one examines new configurations that are "close" to the current solution in the sense that they can be obtained from it by performing a small perturbation. A new configuration replaces the current one if either its total link cost is less than the total link cost associated with

```
Procedure Simulated_annealing;
BEGIN
    Initialize(Topo0,θ0, L, α);
    k := 0;
    SolutionI := Topo0;
    REPEAT
      FOR  I := 1  TO  L  DO
      BEGIN
        GENERATE(SolutionJ ∈ Neighborhood of SolutionI);
          IF Cost(SolutionJ) ≤ Cost(SolutionI) THEN
            SolutionI := SolutionJ
          ELSE
            IF exp[Cost(SolutionI) - Cost(SolutionJ)/θk] > Random[0,1) THEN
              SolutionI := SolutionJ;
      END;
      θk+1 := θk*α
      k := k + 1;
    UNTIL Stop_criterion;
END;
```

Fig. 1. Simulated annealing algorithm.

the current configuration, or the acceptance rule forces one to accept it. Before applying simulated annealing to solve a specific problem, decisions must be made about a number of specific questions such as the manner in which to generate an initial configuration, the cost of a configuration, the neighborhood of a configuration (i.e. the set of configurations that are considered close to it), and the definition of a feasible solution, and about the values of some control parameters.

The generation of the initial configuration is achieved by the following procedure:

Step 1: Fix n the number of nodes of the network, and their coordinates.
Step 2: Fix K the desired degree of connectivity.
Step 3: Compute Euclidean distances for every pair of nodes and sort the distances in increasing order.
Step 4: Build a minimum spanning tree.
Step 5: Sequentially add shortest links between nodes of smallest degrees until all nodes have degree greater than or equal to K.
Step 6: If the network is K-connected, go to Step 7; else add shortest links between nodes of smallest degrees until the network becomes K-connected.
Step 7: Draw up the initial topological configuration.

The above-mentioned test of K-connectivity (Step 6) is performed according to the algorithm described in Ref. 18.

The cost of a configuration is defined as the total link cost D of the topology associated with this configuration. A topology is a neighbor of another topology if and only if it satisfies the following criteria:

(1) It is obtained by replacing two links $\{i_1, j_1\}$ and $\{i_2, j_2\}$ of the current topology by the pair of new links $\{i_1, j_2\}$ and $\{i_2, j_1\}$, or the pair $\{i_1, i_2\}$ and $\{j_1, j_2\}$.
(2) It is K-connected, with $2 \leq K \leq 5$.
(3) Its average delay T is less than or equal to the maximum allowable delay T_{max}.

4. COMPUTATIONAL EXPERIMENTS AND RESULTS

In order to test the efficiency of simulated annealing in the context previously described, two numerical applications have been undertaken on an IBM PC 386 with a set of twenty nodes or sites. In both cases, the problem was to find 3-connected topologies which guarantee a maximum allowable delay $T_{max} = 50$ ms per packet. For the sake of simplicity and without loss of generality, the traffic is assumed to be uniform and equal to 10 packets/s for every pair of nodes. The average packet length has been taken to be 1000 bits. The set of capacity options available and used is shown in Table 1.

The temperature parameter must have a very high initial value.[35] Good results have been obtained with a value of T equal to the difference between the cost of the complete topology and the cost of the initial topology. Annealing factors equal to 0.9, 0.7 and 0.5 have been used to decrement the level of the temperature

Table 1. Capacity options and costs

Capacity (kbps)	Variable cost [($/month)/km]	Fixed cost ($/month)
9.6	3.0	10.0
19.6	5.0	12.0
56.0	10.0	15.0
100.0	15.0	20.0
200.0	25.0	25.0
560.0	90.0	60.0

Table 2. Node locations for the first network

No.	X	Y	No.	X	Y
1	205	230	11	220	370
2	170	420	12	310	410
3	580	150	13	500	390
4	480	80	14	320	240
5	90	130	15	205	85
6	40	210	16	280	20
7	400	370	17	590	310
8	435	185	18	20	400
9	530	250	19	365	40
10	275	165	20	50	30

down to 0.05, considered as a stop criterion. For each fixed temperature value, a number of attempts at changing the current solution were made.

4.1. First application

Table 2 shows the node locations for the first network represented by the set of Cartesian coordinates X and Y. The initial topology corresponding to the set of data is shown on Fig. 2. Its total link cost $D=$ \$208,991/month and its average delay $T=37.11$ ms per packet. Table 3 gives the values of the link attributes for this initial topology.

From this initial topology, the simulated annealing algorithm performs a number of cycles. Each cycle corresponds to 5 trials ($L=5$) to modify the current topology; after a cycle, the value of the temperature is decremented by an annealing factor of 0.7. Figure 3

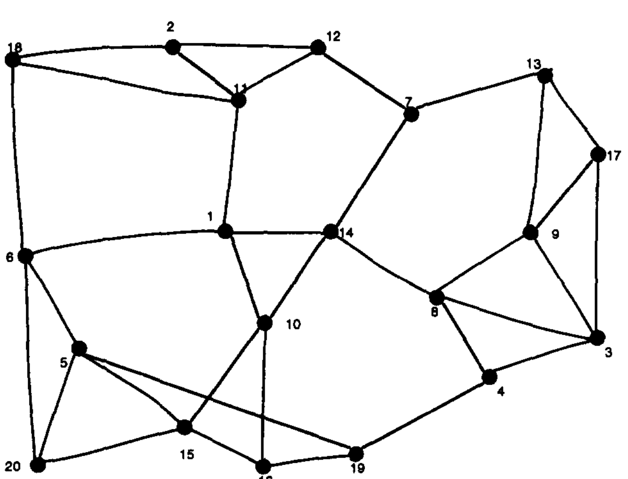

Fig. 2. Initial topology for the first network. $D=\$208,991$/month and $T=37.11$ ms.

Table 3. Link attributes of the initial topology (1st network)

Link	Length	Flow (kbps)	Capacity (kbps)
1–6	166	170	200.0
1–10	96	150	200.0
1–11	141	300	560.0
1–14	115	250	560.0
2–11	71	90	100.0
2–12	140	100	200.0
2–18	151	90	100.0
3–4	122	90	100.0
3–8	149	70	100.0
3–9	112	50	56.0
3–17	160	30	56.0
4–8	114	140	200.0
4–19	122	170	200.0
5–6	94	140	200.0
5–15	123	130	200.0
5–19	289	60	100.0
5–20	108	10	19.2
6–18	191	80	100.0
6–20	180	50	56.0
7–12	98	210	560.0
7–13	102	250	560.0
7–14	153	220	560.0
8–9	115	200	560.0
8–14	127	380	560.0
9–13	143	120	200.0
9–17	85	80	100.0
10–14	87	280	560.0
10–15	106	240	560.0
10–16	145	140	200.0
11–12	98	100	200.0
11–18	202	60	100.0
13–17	120	50	56.0
15–16	99	100	200.0
15–20	164	90	100.0
16–19	87	130	200.0

shows the evolution of network costs as a function of the number of trials, and Fig. 4 presents the topological configuration or local solution obtained at the second attempt of the third cycle, whereas Table 4 gathers results related to the length, the flow and the capacity of links which define the corresponding topology. For this topology, the total link cost is \$186,376/month, which represents a saving of \$22,615/month in comparison with the total link cost of the initial topology. Figure 5 shows the variations of the total link cost D with the average delay T for the different topologies generated during the perturbation process. Table 5

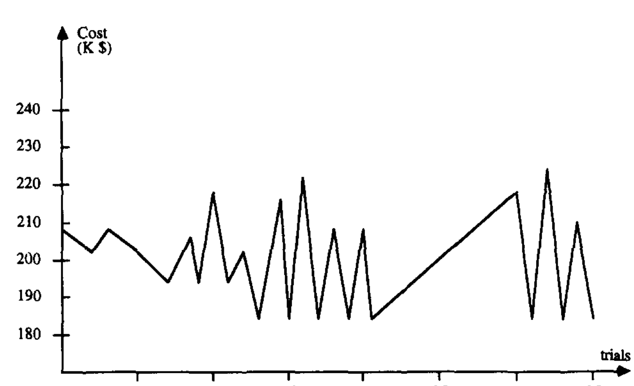

Fig. 3. Evolution of the network costs as a function of the number of trials.

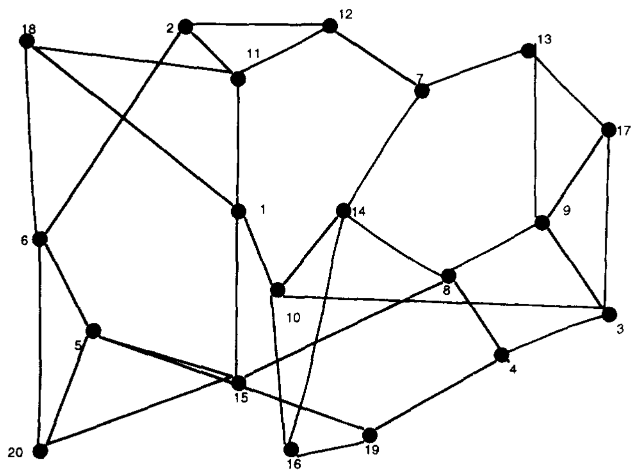

Fig. 4. Local solution obtained with an annealing factor of 0.7. $D = \$186,376$/month and $T = 47.34$ ms.

summarizes the results obtained as a function of the size of the network, whereas Table 6 presents the results as a function of the annealing factor and the number of trials per cycle.

4.2. Second application

Table 7 shows the Cartesian coordinates X and Y of the 20 nodes defining the second network. Figure 6 presents the topological configuration related to the

Table 4. Link attributes of the local solution

Link	Length	Flow (kbps)	Capacity (kbps)
1–10	96	370	560.0
1–11	141	160	200.0
1–15	145	190	200.0
2–6	247	90	100.0
2–11	71	80	100.0
2–12	140	140	200.0
3–4	122	90	100.0
3–9	112	80	100.0
3–10	305	40	56.0
3–17	160	30	56.0
4–8	114	160	200.0
4–19	122	170	200.0
5–6	94	140	200.0
5–15	123	130	200.0
5–19	289	60	100.0
5–20	108	10	19.2
6–18	191	80	100.0
6–20	180	50	56.0
7–12	98	210	560.0
7–13	102	250	560.0
7–14	153	220	560.0
8–9	115	200	560.0
8–14	127	380	560.0
9–13	143	120	200.0
9–17	85	80	100.0
10–14	87	280	560.0
10–15	106	240	560.0
10–16	145	140	200.0
11–12	98	100	200.0
11–18	202	60	100.0
13–17	120	50	56.0
15–16	99	100	200.0
15–20	164	90	100.0
16–19	87	130	200.0

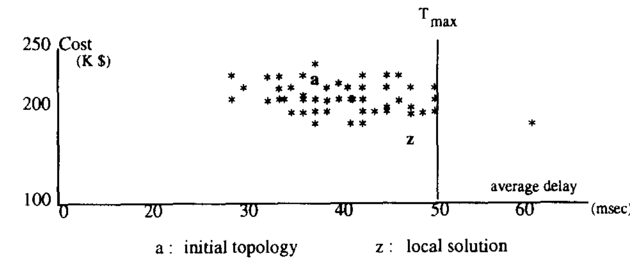

Fig. 5. Total link cost vs average delay for various topologies.

initial topology, whereas Table 8 gives the attribute values of the links which define this topology.

Figure 7 presents the local solution obtained with an annealing factor of 0.7. Table 9 gathers results related to the flow and the capacity of links which define the topology of the local solution.

4.3. Result analysis

As shown in Figs 2 and 3, the total link cost of the initial topology is reduced from \$208,991/month to \$186,376/month, i.e. an improvement of 11% for this network. In the case of the second application network, this improvement is of 8%, for a total link cost of \$197,776/month for the initial topology of Fig. 6, and of \$182,662/month for the local solution of Fig. 7. In both cases, average delay and 3-connectivity constraints are satisfied, since T is always $\leqslant T_{max} = 50$ ms and each node pair is linked by at least three node-disjoint paths.

With reference to Tables 3 and 8, the results show that link flows are not uniformly distributed, even if all traffic requirements between pairs of nodes are equal. This illustrates the role played by routing in determining flow and capacity assignments.

On the other hand, it is observed in Fig. 5 that some network topologies have the same total link cost but different values of average delay, whereas there are some others which have the same average delay but different values of total link cost. Hence, there is no deterministic relationship enabling one to forecast the behavior of the cost function when the average delay varies.

With reference to Table 5, one can observe that the CPU time and the total link cost increase quasi-linearly with the average delay which decreases with the number of nodes. Moreover, it appears that the number of trials L has a significant effect on the total link cost which decreases when L increases, for a constant value of α, as indicated in Table 6.

Table 5. Results obtained as a function of the size of the network

No. of nodes	CPU time (min)	Initial cost (\$/month) Initial delay (ms)	Final cost (\$/month) Final delay (ms)
10	3	32,953	32,953
		87.90	87.90
15	4	87,835	71,005
		42.22	65.44
20	6	208,991	186,376
		39.99	47.34

Table 6. Results obtained as a function of the annealing factor and the number of trials per cycle (plateau length)

		Initial cost: 208,991 Initial delay: 37.11	Initial cost: 87,835 Initial delay: 42.22	Initial cost: 32,953 Initial delay: 87.90
α	L	Final cost ($/month) Final delay (ms) $n=20$	Final cost ($/month) Final delay (ms) $n=15$	Final cost ($/month) Final delay (ms) $n=10$
0.5	5	199,138	77,916	32,953
		39.78	51.63	87.90
0.5	10	174,093	74,985	32,953
		42.80	47.71	87.90
0.7	5	186,376	71,005	32,953
		47.34	65.44	87.90
0.7	10	180,814	68,260	32,953
		40.32	55.57	87.90
0.9	5	198,973	76,065	32,953
		50.34	53.34	87.90
0.9	10	186,808	68,260	32,953
		39.58	55.57	87.90

Table 7. Node locations for the second network

No.	X	Y	No.	X	Y
1	250	360	11	100	300
2	165	420	12	350	420
3	600	100	13	550	360
4	480	55	14	320	240
5	90	130	15	205	85
6	150	250	16	365	40
7	400	325	17	595	310
8	435	185	18	15	260
9	530	250	19	415	80
10	275	165	20	50	25

5. CONCLUDING REMARKS

This paper has shown that the application of simulated annealing to designing distributed computer networks is promising. In order to verify the quality of solutions generated by this method, several computational experiments have been performed. The results have demonstrated that this approach can be used for solving this very hard combinatorial optimization problem, since good, low-cost feasible solutions have been obtained in reasonable CPU times. Like the artificial inteligence approach described in Ref. 29, simulated annealing is shown to be robust and flexible, in terms of convergence speed and range of design issues and constraints that it can accommodate.

In the near future, the implementation of this approach will be modified in order to make it interactive, and to facilitate human intervention during the

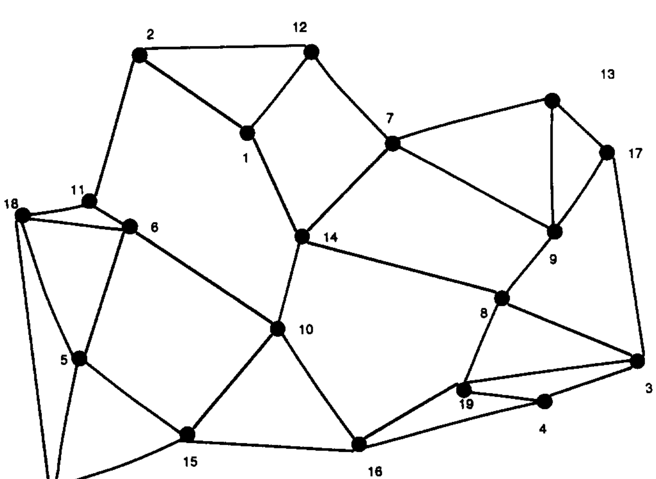

Fig. 6. Topological configuration of the initial topology generated. $D=\$197,776$/month and $T=42.40$ ms.

Table 8. Link attributes of the initial topology (2nd network)

Link	Flow (kbps)	Capacity (kbps)
1–2	120	200.0
1–12	50	56.0
1–14	170	200.0
2–11	130	200.0
2–12	90	100.0
3–4	100	200.0
3–8	70	56.0
3–17	30	100.0
3–19	10	19.2
4–19	90	100.0
5–6	100	200.0
5–15	20	200.0
5–18	170	56.0
5–20	20	56.0
6–10	260	560.0
6–11	110	200.0
6–18	20	56.0
7–9	120	200.0
7–12	200	560.0
7–13	140	200.0
7–14	220	560.0
8–9	350	560.0
8–14	320	560.0
8–19	190	200.0
9–13	40	56.0
9–17	90	100.0
10–14	420	560.0
10–15	260	560.0
10–16	150	200.0
11–18	40	56.0
13–17	60	100.0
15–16	130	200.0
15–20	130	200.0
16–19	90	100.0
18–20	10	19.2

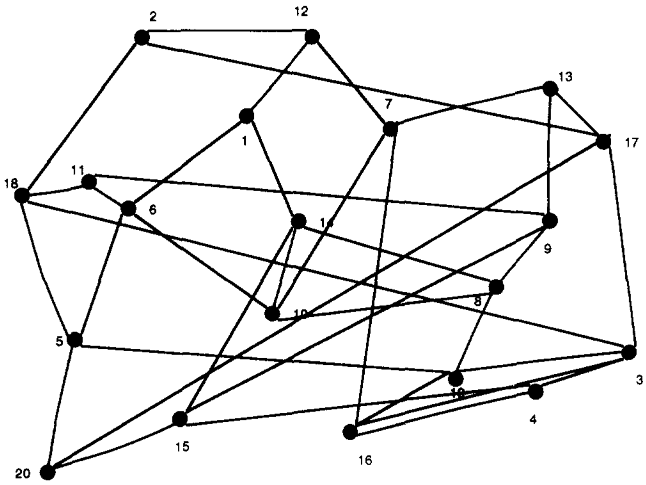

Fig. 7. Topological configuration of the local solution obtained with an annealing factor of 0.7. D = \$182,662/month and T = 39.72 ms.

Table 9. Link attributes of the local solution (2nd network)

Link	Flow (kbps)	Capacity (kbps)
1–6	90	200.0
1–12	140	200.0
1–14	150	200.0
2–12	150	200.0
2–17	20	56.0
2–18	80	100.0
3–4	30	56.0
3–16	40	56.0
3–18	10	19.2
3–19	110	200.0
4–15	50	56.0
4–16	170	200.0
5–6	160	200.0
5–15	20	200.0
5–18	80	100.0
5–19	10	200.0
5–20	120	200.0
6–10	270	560.0
6–11	170	200.0
7–12	210	560.0
7–13	160	200.0
7–16	120	200.0
8–9	170	200.0
8–10	180	200.0
8–14	170	200.0
8–19	340	560.0
9–11	50	56.0
9–13	210	560.0
9–15	50	56.0
10–14	90	560.0
11–18	100	200.0
13–17	110	200.0
14–15	90	100.0
15–20	110	200.0
16–19	280	560.0
17–20	10	19.2

process of designing network topologies. In fact, the addition of such a module which operates as a user interface module of a knowledge-based system will make it easier to perform sensitivity analysis of the results.

REFERENCES

1. Gerla M. and Kleinrock L. On the topological design of distributed computer networks. *IEEE Trans. Commun.* **COM-25**, 48–60 (1977).
2. Boorstyn R. R. and Frank H. Large-scale network topological optimization. *IEEE Trans. Commun.* **COM-25**, 29–47 (1977).
3. Gerla M. Approximations and bounds for the topological design of distributed computer networks. *Procedings of the Fourth Data Communications Symposium*, Québec City, pp. 4.9–4.15 (1975).
4. Kronz R., Lee S. and Sun M. Practical design tools for large packet-switched networks. *Proceedings of IEEE INFOCOM*, pp. 591–599 (1983).
5. Yokohira T., Sugano M., Nishida T. and Miyahara H. Fault tolerant packet-switched network design and its sensitivity. *IEEE Trans. Reliab.* **40**, 452–470 (1991).
6. Chen K. J., Stach J. F. and Wu T. H. A new method for topological design in large, traffic laden packed-switched networks. *Proceedings of the Ninth Data Communications Symposium*, Whistler Mountain, pp. 115–121 (1985).
7. Saksena V. R. Topological analysis of packet networks. *IEEE J. Selected Areas Commun.* **17**, 1243–1252 (1989).
8. Kamimura K. and Nishino H. An efficient method for determining economical configurations of elementary packet-switched networks. *IEEE Trans. Commun.* **39**, No. 2 (1991).
9. Monma C. L. and Sheng D. D. Backbone network design and performance analysis: a methodology for packet switching networks. *IEEE J. Selected Areas Commun.* **Sac-4**, 946–965 (1986).
10. Garey M R. and Johnson D. S. *Computers and Intractability: a Guide to the Theory of NP-Completeness.* Freeman, New York (1979).
11. Newport K. T. and Varshney P. K. Design of survivable communications networks under performance constraints. *IEEE Trans. Reliab.* **40**, No. 4, 433–440 (1991).
12. Boesch F. T. and Frisch I. T. On the smallest disconnecting set in a graph. *IEEE Trans. Circuit Theory* **CT-15**, 286–288 (1986).
13. Wong J. W., Sauve J. P. and Field J. A. A study of fairness in packet-switching networks. *IEEE Trans. Commun.* **30**, 346–353 (1982).
14. Dutta A. and Mitra S. Integrating heuristic knowledge and optimization models for communication network design. *IEEE Trans. Knowledge Data Engng* **5**, 999–1017 (1993).
15. Chang S.-G. Fair integration of routing and flow control in communication networks. *IEEE Trans. Commun.* **40**, 821–834 (1992).
16. Thaker G. H. and Cain J. B. Interactions between routing and flow control algorithms. *IEEE Trans. Commun.* **COM-34**, 269–277 (1986).
17. Fiol M. A. and Llado A. S. The partial line digraph technique in the design of large interconnection networks. *IEEE Trans. Comput.* **41**, 849–857 (1992).
18. Schumacher U. An algorithm for construction of a *K*-connected graph with minimum number of edges and quasiminimal diameter. *Networks* **14**, 63–74 (1984).
19. Noakes M. D., Cain J. B., Nieto J. W. and Althouse E. L. An adaptive link assignment algorithm for dynamically changing topologies. *IEEE Trans. Commun.* **41**, 694–705 (1993).
20. Newport K. T. and Schroeder M. A. Network survivability through connectivity optimization. *Proceedings of the 1987 IEEE International Conference on Communications*, Vol 1, pp. 471–477 (1987).
21. Gerla M., Frank H., Chou W. and Eckl J. A cut saturation algorithm for topological design of packet-switched communication networks. *Proceedings of the National Telecommunication Conference*, San Diego, pp. 1074–1084 (1974).
22. Chen H. and Zhou J. Reliability optimization in generalized stochastic-flow networks. *IEEE Trans. Reliab.* **40**, 92–97 (1991).
23. Fratta L., Gerla M. and Kleinrock L. The flow deviation method: an approach to store and forward communication network design. *Networks* **3**, 97–133 (1973).
24. Frank H and Chou W. Topological considerations in the design of the ARPA computer network. In *Conference Record 1970 Joint Computer Conference, AFIPS Conference Proceedings*, Vol. 36. AFIPS Press, Montvale, N,J. (1970).
25. Gerla M. The design of store and forward networks for computer communications. Ph.D dissertation, School of Engineering and

Applied Science, University of California, Los Angeles (1973).
26. Ju Ng M. and Hoang D. B. Joint optimization of capacity and flow assignment in a packet switched communication network. *IEEE Trans. Commun.* **COM-35**, 202–209 (1987).
27. Samoylenko S. I. Application of heuristic problem-solving methods in computer communication networks. *Mach. Intell.* **3**, 197–210 (1985).
28. Sykes E. A. and White C. C. Specifications of a knowledge system for packet-switched data network topological design. In *Proceedings of Expert Systems in Government Symposium*, McLean, Va, pp. 102–110 (1985).
29. Pierre S. Application of artificial intelligence techniques to computer network topology design. *Engng Applic. Artif. Intell.* **6**, 465–472 (1993).
30. Pierre S. and Hoang H. H. An artificial intelligence approach for improving computer communications network topologies. *J. Opl Res. Soc.* **41**, 405–418 (1990).
31. Kirkpatrick S., Gelatt C. D. and Vecchi M. P. Optimization by simulated annealing. *Science* **220**, 671–680 (1983).
32. Vanlaarhoven P. J. M. and Aarts E. H. L. *Simulated Annealing: Theory and Applications*. Reidel, Dordrecht (1987).
33. Kirkpatrick S. Optimization by simulated annealing: quantitative studies. *J. Stat. Phys.* **34**, 975–986 (1984).
34. Pirlot M. General local search heuristics in combinatorial optimization: a survey. *Belgium J. O.R., Stat. Comput. Sci.* **32**, 8–67 (1992).
35. Coan B. A., Leland W. E., Vecchi M. P. and Wu L. T. Using distributed topology update and preplannned configurations to achieve trunk network survivability. *IEEE Trans. Reliab.* **40**, 404–416 (1991).

AUTHORS' BIOGRAPHIES

Samuel Pierre received a B.Eng. degree in 1981 from École Polytechnique de Montréal, a B.Sc. and an M.Sc. in Mathematics and Computer Science from UQAM in 1984 and 1985 respectively, an M.Sc. in Economics from Université de Montréal in 1987, and a Ph.D. in Electrical Engineering from École Polytechnique de Montréal in 1991. He is currently Professor of Computer Science at Université du Québec. His research areas include computer networks, distributed systems, artificial intelligence and CAD/CAM systems. He is the author of 3 books, co-author of two books and two book chapters, as well as more than 30 journal and proceedings papers. In 1989, he received the best paper award of the *Ninth International Workshop in Expert Systems & Their Applications*, held in France.

Michel-Ange Hyppolite received his diploma in Civil Engineering from Faculté des Sciences of Université d'état d'Haïti in 1979, his B.Sc. and M.Sc. degrees in Mathematics and Computer Science from Université du Québec à Montréal (UQAM) in 1991 and 1993 respectively. His research interests include computer communication networks and software engineering.

Jean-Marie Bourjolly is Associate Professor at Concordia University in Montréal. He holds an M.Sc. in Mathematics (Université Paul Sabatier, Toulouse), an M.Sc. in Computer Science (Université de Montréal) and a Ph.D. in Combinatorics and Optimization (University of Waterloo). His research interests include classical Operations Research models such as the maximum clique and stable set problems as well as applications of optimization methods to problems arising in telecommunications and production planning.

Oumar Dioume graduated in engineering mathematics in 1974 from École Polytechnique Fédérale de Lausanne (Switzerland). He received in 1978 a Ph.D. in Mathematical Economics and Econometrics from Université de Paris-I (France), in 1992 a Ph.D. in Engineering Mathematics from École Polytechnique de Montréal (Canada). He is currently living in Montréal where he is Senior Consultant at DTI Télécom. His interests are in telecommunication networks and operations research.