Simulated Annealing for Communication Network Reliability Improvement

Héctor Cancela *María E. Urquhart*

cancela@fing.edu.uy urquhart@fing.edu.uy

Investigación Operativa [*]

Instituto de Computación, PeDeCiBa Informática

Facultad Ingeniería, Montevideo, Uruguay

July 12, 1995

Abstract — A communication network is composed by a set of centers which transmit and receive data, and a set of links which transport this data. To evaluate the capacity of a communication network architecture to resist to the possible failures of some of its components, several reliability metrics are currently used.

When designing such a network, one of the problems is the definition of its topology to guarantee a maximum of reliability. In this work, we tackle the following version of this problem: given an user-defined network topology, find a more reliable alternative network. The proposed methodology is based on the Simulated Annealing meta-heuristic, of recent use in combinatorial optimization.

Key words — Network reliability, combinatorial optimization, simulated annealing.

Resumen — Una red de comunicaciones es un sistema compuesto por un conjunto de centros que reciben y envían información, y un conjunto de conexiones que transportan esa información. La capacidad de una red de comunicaciones para resistir a posibles fallos de algunos de sus componentes se evalúa a través de distintas métricas de confiabilidad.

Uno de los problemas que se plantean durante el diseño de una red, es el de elegir su topología precisamente para asegurar al máximo las comunicaciones. En este trabajo, abordamos una versión de este problema, en la que el objetivo es encontrar una topología más confiable a partir de una red dada por el usuario. La metodología empleada corresponde a la metaheurística "simulated annealing", de reciente utilización en el campo de la optimización combinatoria.

Palabras clave — Confiabilidad de redes, optimización combinatoria, "simulated annealing".

[*] This work is a result of BID/CONICYT Project 153

1 Introduction

During the design of a communication network, one of the problems is the definition of its topology in such a way as to guarantee a maximum of reliability in the communications. Such a network can be modeled by a graph $G = (\mathcal{V}, \mathcal{E})$, with node set $\mathcal{V}$ and edge set $\mathcal{E}$. Its reliability is defined as the probability of successful communication between the nodes, and can be computed from its components' reliabilities and the topology of the network, either by exact, NP-hard methods, or by Monte Carlo estimation. If there is no information on the behavior of the components, vulnerability measures based solely on the topology can be used [Fis86].

In this work, we have considered the following problem: given an initial network G, find an improved (more reliable) topology in the feasible set $\mathcal{F}$. The combinatorics of the problem are such that preclude trying to find the best topology (global optimum in $\mathcal{F}$). Therefore, techniques that give locally optimal results must be applied. In a previous work [CRU92] we explored the use of a simple local search heuristic (based on 2-opt), with results that showed the need for a more powerful technique. In this work, we consider the application of simulated annealing/simulated quenching techniques. These meta-heuristics have known recent success in different optimization domains, in particular in combinatorial optimization [Egl90] [Con92].

This paper is organized as follows. In Section 2 we give the problem definition. In Section 3 we present the basic elements of the simulated annealing meta-heuristic. Section 4 deals with the application of simulated annealing to the network optimization problem. In Section 5 we present some computational results, based on a model of a subset of Montevideo's optical fiber telephonic network. Finally, some conclusions are given in Section 6.

2 Problem definition

Consider a communication network G where nodes are perfect and links fail randomly and independently. We consider G to be an undirected graph, connected and without loops. When the link failure probabilities are known, the success of communication between nodes in some fixed subset $\mathcal{K}$ of the node-set is a random event. The probability $R_{\mathcal{K}}$ of this event is usually called the *$\mathcal{K}$-terminal reliability*, equivalently the parameter $Q_{\mathcal{K}} = 1 - R_{\mathcal{K}}$ is called the *$\mathcal{K}$-terminal unreliability.* The problem of its evaluation has received considerable attention from the research community (see [LS86], [Col87] and [Rub94] for many references). One of the reasons is that in the general case, this problem is in the #P-complete class, a family of *NP*-hard problems not known to be in *NP* [Bal80] [Bal86] [Pro86]. Roughly speaking, this means that it is most unlikely that a general algorithm to compute exactly $R_{\mathcal{K}}$ exists, whose running time grows polynomially with the problem size. Practically, the computational time required to solve a medium size general model (say, with many dozens of links) is prohibitive on a workstation. A Monte Carlo approach is an alternative allowing the evaluation of larger networks, specially when used in conjunction with variance reduction techniques (see for example [Fis86], [ER92], [EMR90], [EGL91], [CK94]).

Depending on the choice of the set $\mathcal{K}$, we have different reliability metrics. The most used ones are source-terminal reliability R_{st} where s and t are two fixed nodes of V, and all-terminal reliability R_V. This work uses the reliability metrics R_V, but the ideas presented are independent of the choice of $\mathcal{K}$.

When there is no information available on the network components' reliability, purely topological measures can be used to quantify the fragility of a network. These measures (called vulnerability indexes) generally are simple functions of easily computed network parameters having clear relations with the intuitive notion of fragility of a topology [SA81].

In this context, we consider the following problem: given a feasible set of network topologies $\mathcal{F}$, and given an initial network $G \in \mathcal{F}$, find the optimal topology (in the sense of maximal reliability or minimal vulnerability). We will consider $\mathcal{F}$ as a subset of the set of connected networks with the same node set as G. If there's no further restriction upon $\mathcal{F}$, the optimal topology is trivially the complete graph. A more interesting case is when the cost of a network is defined as the sum of the individual costs of all its edges, which are defined to take into account the special characteristics of the problem (geographical, or other). We may then impose an arbitrary (user defined) upper bound on the considered networks' cost. Then $\mathcal{F} = \{g | COST(g) \leq BOUND\}$, where $COST()$ is the cost function and $BOUND$ is the cost upper bound. In this case, the optimization problem is rather hard, because of the combinatorial explosion in size of the search state space. An additional difficulty is that the objective function (network reliability) is computationally hard to evaluate. In practice, we must then restrict ourselves to consideration of the following modified problem: given a feasible set of network topologies $\mathcal{F}$, and given an initial network $G \in \mathcal{F}$, find an improved (more reliable or less vulnerable) topology respect to the original one. This is then the problem addressed in Section 4.

3 Simulated Annealing

Simulated annealing (SA) is a method for obtaining good solutions to difficult optimization problems which has received much attention over the last few years. In particular, it has been applied with success in the domain of combinatorial optimization [Egl90] [Con92]. The recent interest began with the work of Kirkpatrick et al [KGV83] and Cerny [Cer82], which generalized previous results by Metropolis et al. [MRR$^+$53].

Suppose that the solution space $\mathcal{S}$ is the finite set of all solutions and the energy function f is a real valued function defined on members of $\mathcal{S}$. The problem is to find a solution or state, $s \in \mathcal{S}$, which minimizes f over $\mathcal{S}$. Simulated annealing works by generating a random neighbor of current state s, and accepting (or rejecting) it following an acceptance rule. The acceptance rule always accepts a move if it results in a reduced value of the objective function f; but if the considered neighborhood move results in an increase δ in f, it may still be accepted, with probability $e^{-\delta/T}$ (Metropolis acceptance rule). Here T is a control parameter (called temperature), which controls the probability

```
Select an initial state $i \in \mathcal{S}$;
Select an initial temperature $T > 0$;
Set temperature change counter $t := 0$;
Repeat
    Set repetition counter $n := 0$;
    Repeat
        Generate state $j$, a random neighbor of $i$;
        Calculate $\delta := f(j) - f(i)$;
        If $\delta < 0$ then
            $i := j$;
            If $f(i) < f(i^*)$ then $i^* := i$;
        else if $random(0,1) < e^{-\delta/T}$ then $i := j$;
        n := n+1;
    until $n := N(t)$;
    $t := t + 1$;
    $T := T(t)$;
Until stopping criterion true;
Return best found solution $i^*$
```

Figure 1: Pseudo-code for Simulated Annealing

of uphill moves: when T is high, most moves will be accepted, but as it approaches 0, most uphill moves will be rejected. In classical simulated annealing, the algorithm is started with a relatively high value of T, and proceeds by attempting a certain number of moves at each temperature, while the temperature parameter is gradually dropped. The simulated annealing heuristics is illustrated in pseudo-code in Figure 1. The general scheme of this meta-heuristic is very simple, but its application to a given problem is only possible after a number of problem-specific choices.

We may identify several elements to be defined when applying SA to a particular problem [Egl90]: **the energy function** f, usually taken as the objective function for the original problem; **the neighborhood structure**, with a generation probability function; **the acceptance rule**, usually taken to be the Metropolis rule, as illustrated in Figure 1; **the cooling schedule**, i.e: the functions $N(t)$, the number of iterations at step t, and $T(t)$, the temperature; **the stopping criterion**, which may be a final temperature, a number of steps, a number of steps without changes in the state, etc.

There are a number of results concerning the convergence of SA to an optimal solution in quite general conditions (see between others [AF87], [LM86]). But for the most part, they depend on using a cooling schedule which leads to an exponential number of iterations. In most practical applications, a cooling schedule is used which doesn't guarantee the convergence to the global optimum. The resulting algorithms are sometimes labeled Simulated Quenching [Ing93].

4 Simulated Annealing for reliability improving

In this section we discuss the tailoring of SA to our problem. In choosing among different alternatives, we take as first criterion the simplicity of concept and implementation, in order to obtain a basic algorithm which may be improved in the future in the directions suggested by the results of this work.

Solution space and energy function

The most obvious choices for the solution space $\mathcal{S}$ and the energy function f are the set $\mathcal{F}$ of all feasible networks (as defined in Section 2) and the unreliability Q_V (or a vulnerability index Vul), respectively. A better approach is to relax the cost restriction and incorporate it in the objective function [CS94].

We propose that the solution space be defined as the set of all networks having the same node set $\mathcal{V}$ as the original network G, and that the cost network restriction be incorporated in the energy function f. Then this function will be the sum of two terms, one related to the network's reliability and the other a penalty term related to the cost. As these two quantities have different measure units, they must be normalized in some way. We use, for any given network g,

$$f(g) = \frac{Q_V(g)}{Q_V(G)} + \alpha(T) \max\left(\frac{COST(g)}{BOUND} - 1, 0\right)$$

where G is the initial network topology, $BOUND$ is the (user-defined) upper bound on the cost of a network, and $\alpha(T) = K/T$ is decreasing on T (then, it's increasing on t). The penalty term is 0 for all feasible networks (those with cost not greater than $BOUND$).

For infeasible networks, the penalty is decreasing with T; this means that in the start of the SA algorithm, a network can be accepted even if it doesn't satisfy the cost constraint, but as T decreases, the penalty term will grow, leading the algorithm to only accept feasible solutions. The constant K must be fixed following some criterion. In our implementation, we choose to take it equal to the starting temperature T_0: this means that at the start of the search, reliability and cost have same relative weight, and so a given percent increase of cost can be offset by an equal decrease in unreliability.

When the elementary link reliabilities are not known, the measure $Q_V()$ is not defined, and a vulnerability index must be used instead. In this work we consider the vulnerability index Vul defined in [CRU92] as

$$Vul(g) = \frac{D + CE(c) + CV(c')}{t + c.c'},$$

constructed with the following parameters:

- D : diameter of graph g, i.e. the lenght of the longest shortest path between a pair of nodes of $\mathcal{V}$;
- t : girth of graph g, i.e. the lenght of the shortest cycle of g;
- c : edge connectivity, i.e. the minimum number of edges whose removal disconnects the graph;
- c' : connectivity (also called node connectivity), i.e. the minimum number of nodes whose removal disconnects the graph;
- $CE(m)$: the maximum over all pairs of nodes i, j of the number of i, j cutsets of size m (an i, j cutset is a set of edges whose removal disconnects i and j),
$$CE(m) = \max_{i,j \in V} |\{i, j \text{ cutsets of size } m\}|;$$
- $CV(m)$: the maximum over all pairs of nodes i, j of the number of i, j node-cutsets of size m (an i, j node-cutset is a set of nodes whose removal disconnects i and j),
$$CV(m) = \max_{i,j \in V} |\{i, j \text{ node} - \text{cutsets of size } m\}|.$$

The energy function is then

$$f(g) = \frac{Vul(g)}{Vul(G)} + \alpha(T) \max\left(\frac{COST(g)}{BOUND} - 1, 0\right),$$

and has the same properties as in the previous case (when the elementary link reliabilities are known).

Neighborhood structure

To choose the neighborhood structure of our problem we take into account the following properties: to have access to the entire state space (defined as the set of all networks with same node set as G), and to have an efficient implementation of the random generation function. We define the neighborhood of a graph g as being composed of all graphs with an added or deleted link with respect to g. The random selection of a neighbor is done in the following way: two nodes n_1, n_2 are taken at random; if the link (n_1, n_2) was present in g, it is deleted, else it is incorporated. With this neighborhood definition, the maximum distance between any two networks (in terms of moves needed to pass from one to the other) is $n(n-1)/2$ (this is the distance between any network and its complementary, where to pass from one to the other we must delete all existing links and add all non-existing ones). The convergence speed, as discussed in [Haj88], is inversely proportional to this maximum distance. Another property of this neighborhood structure is that the size of $N(g)$ is also $n(n-1)/2$ for all g (this is the number of ways to choose two different nodes).

Acceptance rule

The standard Metropolis acceptance rule[1] was selected. This is the choice made in most SA schemes, even when there are theoretical and empirical results [AF87] [Egl90] showing that a wide gamma of other rules give comparable results. As reported in [Egl90], there's some literature on the use of an approximate value of δ, when its exact evaluation is computationally hard. For our problem, we use a Monte Carlo estimate of the unreliability parameter $Q_V()$ to compute δ, instead of its exact value.

Cooling schedule

For the cooling schedule, several choices are possible. To assure the convergence to the global optimum, a logarithmic schedule ($T(t) = T_0/\log(t)$) is necessary, but this leads to computational requirements too high for practical use. An exponential schedule ($T(t) = T_0 e^{(c-1)t}$) is most often used, but given the highly combinatorial nature of the network reliability optimization problem, we have chosen an heuristic proposed by Lundy and Mees [LM86], and used with good results in [Con92]. This schedule performs only one iteration for each temperature level ($N(t) = 1, \forall t$) and uses a temperature function $T(t+1) = T(t)/(1+BT(t))$. This function decreases more slowly than the exponential schedule, but without guaranteeing global convergence. The actual speed of descent in temperature is proportional to the cooling constant B; [Con90] suggests taking $B << T_0$.

Stopping criterion

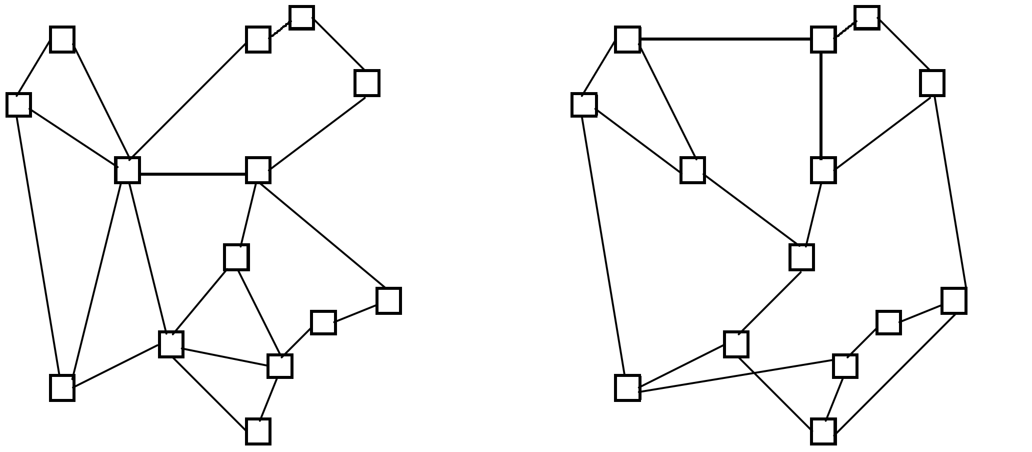
Following the simplicity rule stated at the beginning of the section, the stopping criterion selected is a total number of iterations: the search ends when $t = MAX\_REPL$. In our implementation, this is equivalent to setting a final temperature level, as we have that $T_f = T_0/(1 + MAX\_REPL \times B \times T_0)$.

The above choices define a SA algorithm suited for the network reliability optimization problem. But there are still some free parameters: the initial temperature, T_0; the cooling constant B; the total number of replications MAX_REPL; the initial topology G and the cost upper bound, $BOUND$. All these must be fixed in a problem by problem basis, and will be discussed in the case study treated in the following section.

5 Computational Results

In this section we present the results of applying the previously defined SA algorithm to a model of a subset of Montevideo's optical fiber telephonic network. The topology shown in Figure 2 (14 nodes, 21 links) is the one in operation in 1992, and was designed by the national telecommunications company following heuristic rules.

[1] $Pr(\text{acceptance}) = e^{-\delta/kT}$, with δ the change in the energy function



a) Original topology

b) Alternative (by long runs SA)

Figure 2: Montevideo's network

If we take the costs of the links to be their geographical length (in an arbitrary lenght unit), this network has an overall cost of 1186. Its reliability parameter, as computed by a Monte Carlo method, is $R_V \simeq 0.896$ when the elementary reliability of the links is set to 0.9. The value of the vulnerability index for this topology is $Vul = 2.429$.

The objective of the experimentation was to validate the approach defined in the previous section, and to find which range of parameters was appropriate for the network under consideration. We considered two main scenarios for the network cost upper bound as related to the cost of the existing network: either no cost increase, or an increase of 10% (the upper bounds for the network cost being 1186 and 1304 respectively). Both with the reliability and the vulnerability measures were considered.

The SA algorithm was implemented using the C++ network classes and methods available in the HEIDI tool [CRU92]. The evaluation of the Q_V component of the objective function was done by a standard Monte Carlo method (number of replicacions 10^3). The reliability estimates for the best found topologies was done with an increased number of replications, namely 10^5.

For the vulnerability measure, the experiments couldn't be finished, as the time per iteration of the SA algorithm was exceedingly large (due to the cost of computing the index Vul). After 20 hours of work, less than 500 iterations were completed, and no global improvement had been found.

Three series of experiments were done with the reliability measure R_V. In the first two, the initial topology was set to the original network (Figure 2), while varying the lenght of the SA runs. The first series consisted of short runs of 5×10^3 replications each, while the second corresponds to longer runs (5×10^4 replications each). For the shorter runs the cooling parameter B was set to 0.005, and for the longer ones $B = 0.001$. The third series corresponds to short runs of a different initial topology, namely the complete graph K_{14}. In all cases, the initial temperature, T_0, was set to 0.5 as this was considered a high enough value to allow exploration of the whole state space (at this temperature, a move resulting in network cost increment of 50% is accepted with probability $e^{-1} \simeq 0.368$).

In order to compare the results, the SA algorithm was run five times for each considered parameter combination. The results presented in tables 1, 2 and 3 show the reliability and the cost of the best topology found in each run, as well as the mean value over all runs. For quick reference, in each table the reliability and cost of the best topology over the five runs is tabulated in the last column.

	Run						
	1	2	3	4	5	Mean	Best
R_V	.9357	.9088	.9348	.9075	.9234	.9220	.9357
$Cost$	1161	1150	1169	1139	1140	1152	1161
Parameters: $MAX\_REPL = 5 \times 10^3$, $BOUND = 1186$.							

	Run						
	1	2	3	4	5	Mean	Best
R_V	.9549	.9602	.9304	.9454	.9470	.9476	.9602
$Cost$	1303	1281	1287	1275	1295	1288	1281
Parameters: $MAX\_REPL = 5 \times 10^3$, $BOUND = 1304$.							

Table 1: Results of SA short runs.

	Run						
	1	2	3	4	5	Mean	Best
R_V	.9383	.9358	.9589	.9533	.9250	.9423	.9589
$Cost$	1178	1183	1186	1172	1161	1176	1186
Parameters: $MAX\_REPL = 5 \times 10^4$, $BOUND = 1186$.							

	Run						
	1	2	3	4	5	Mean	Best
R_V	.9639	.9527	.9619	.9563	.9717	.9613	.9717
$Cost$	1300	1281	1294	1301	1304	1296	1304
Parameters: $MAX\_REPL = 5 \times 10^4$, $BOUND = 1304$.							

Table 2: Results of SA long runs.

Table 1 shows the results for the short runs when the starting solution is the original graph given in Figure 2. In all runs, the SA algorithm found an improved topology over the original one. The amount of the reliability improvement varies from one run to

	Run						
	1	2	3	4	5	Mean	Best
R_V	.9134	.9135	.9032	.9125	.9041	.9093	.9135
$Cost$	1169	1168	1174	1182	1173	1173	1168
Parameters: $MAX\_REPL = 5 \times 10^3$, $BOUND = 1186$.							
	Run						
	1	2	3	4	5	Mean	Best
R_V	.9364	.9417	.9507	.9489	.9406	.9436	.9507
$Cost$	1275	1291	1303	1304	1301	1295	1303
Parameters: $MAX\_REPL = 5 \times 10^3$, $BOUND = 1304$.							

Table 3: Results of SA short runs with initial topology the complete graph.

another; when $BOUND = 1186$ the improvement is in the range $1\% - 4\%$, and when $BOUND = 1304$ it is in $3\% - 7\%$.

Table 2 shows the results for the longer runs. These results, compared with those of Table 1, show a more stable behaviour of the algorithm. This is probably due to the increased number of iterations and the slower cooling schedule. The average reliability improvement is of 5% and 7% respectively for both cost upper bounds, and the best solutions over the five iterations are also considerably better than the ones obtained with the shorter runs. The best network topology obtained for $BOUND = 1186$ is also shown in Figure 2.

Finally, Table 3 shows the results when the starting solution is the complete graph K_{14}, for short runs ($MAX\_REPL = 5000$, $B = 0.005$). This corresponds to starting the search from a "highly energetic" state, which might be advantageous to more thoroughly search the solution space. The results are rather worse those obtained with the original starting solution (results in Table 1). Then, we can conclude that the starting solution has a non negligeable influence in the search.

6 Conclusions

In this paper we have presented the network reliability optimization problem, and we have proposed a Simulated Annealing algorithm which can be applied to this problem. The performance of the algorithm was tested on a particular case (a model of a subset of Montevideo's telephonic network) for different parameter values. The results obtained show the potential of the SA heuristic to find improved alternatives to a given network topology, when using a Monte Carlo estimate of the reliability measure R_V. Further work is needed to determine the appropriate parameter values for different topologies. On the other hand, when using the vulnerability index Vul the SA took too much time to be useful. Future work must include the developement of other, less computationally expensive vulnerability indexes, suitable for use with this methodology.

Some other modifications of the SA heuristic which should be considered are the inclusion of a local descent algorithm at the end of the SA search, and the study of the behaviour of the search with other cooling schedules (in particular, with a temperature automatically fixed at a suitable level); also, the Monte Carlo evaluation of the reliability measure could be modified to incorporate variance reduction techniques and to take into account the local nature of the proposed neighborhood moves.

References

[AF87] S. Anily and A. Federgruen. Simulated annealing methods with general acceptance probabilities. *J. Appl. Prob.*, 24, 1987.

[Bal80] M.O. Ball. The complexity of network reliability computations. *Networks*, 10:153–165, 1980.

[Bal86] M.O. Ball. Computational complexity of network reliability analysis: An overview. *IEEE Trans. Reliab.*, R-35(3), 1986.

[Cer82] V. Cerny. A thermodynamical approach to the travelling salesman problem: An efficient simulation algorithm. Report, Comenius University, Bratislava, Czechoslovakia, 1982.

[CK94] H. Cancela and M. El Khadiri. A recursive variance reduction method for estimating communication network reliability. Research Report 860, IRISA, Rennes, France, Campus de Beaulieu 35042 France, September 1994.

[Col87] C.J. Colbourn. *The Combinatorics of Network Reliability.* Oxford University Press, New York, 1987.

[Con90] D. Connolly. An improved annealing scheme for the QAP. *European Journal of Operational Research*, 46, 1990.

[Con92] D. Connolly. General purpose simulated annealing. *European Journal of Operational Research*, 43, 1992.

[CRU92] H. Cancela, G. Rubino, and M. E. Urquhart. Optimización en el diseño de redes de comunicación. In *Proceedings of the XIII CILAMCE*, Porto Alegre, Brasil, November 1992.

[CS94] W.H. Chen and B. Srivastava. Simulated annealing procedures for forming machine cells in group technology. *European Journal of Operational Research*, 75, 1994.

[Egl90] R.W. Eglese. Simulated annealing : A tool for OR. *European Journal of Operational Research*, 46, 1990.

[EGL91] T. Elperin, I. Gertsbakh, and M. Lomonosov. Estimation of network reliability using graph evolution models. *IEEE Trans. Reliab.*, 40(5), 1991.

[EMR90] M. El Khadiri, R. Marie, and G. Rubino. Estimation of network reliability on a parallel machine by means of a Monte Carlo technique. Technical Report 545, IRISA, Campus de Beaulieu, 35042 Rennes, France, 1990.

[ER92] M. El Khadiri and G. Rubino. Reliability evaluation of communication networks. In *SAFECOMP'92, International Conference on Safety, Security and Reliability of Computers*, Zurich, Suisse, October 1992.

[Fis86] G.S. Fishman. A comparison of four Monte-Carlo methods for estimating the probability of s-t conectedness. *IEEE Trans. Reliab.*, R-35(2), 1986.

[Haj88] B. Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13, 1988.

[Ing93] L. Ingber. Simulated annealing: Practice versus theory. *J. Mathl. Comput. Modelling*, 18(11), 1993.

[KGV83] S. Kirkpatrick, C.D. Gelatt, Jr., and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598), 1983.

[LM86] M. Lundy and A. Mees. Convergence of an annealing algorithm. *Mathematical Programming*, 34, 1986.

[LS86] M.O. Locks and A. Satyarayana (editors). Network reliability – the state of the art. *IEEE Trans. Reliab.*, R-35(3), 1986.

[MRR$^+$53] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equation of state calculation by fast computing machines. *J. Chem. Pys.*, 21(6), 1953.

[Pro86] J.S. Provan. The complexity of reliability computations in planar and acyclic graphs. *SIAM J. Comput.*, 15(3), Aug. 1986.

[Rub94] G. Rubino. Tutorial: Efficient evaluation of network reliability. In *7th. Conference on Modeling Tools and Techniques*, Vienna, Austria, 1994.

[SA81] I.M. Soi and K.K. Aggarwal. Reliability indices for topological design of computer communication networks. *IEEE Trans on Reliab.*, R-30(5), 1981.