RELIABILITY MAXIMIZATION FOR NETWORKS WITH PATH-LENGTH CONSTRAINTS

Pablo Burgos
pabbur@hotmail.com
Instituto de Computación, Facultad de Ingeniería, Universidad de la República
J. Herrera y Reissig 565, Montevideo, URUGUAY

Héctor Cancela
cancela@fing.edu.uy
Depto. de Investigación Operativa, Instituto de Computación, Facultad de Ingeniería, Universidad de la República
J. Herrera y Reissig 565, Montevideo, URUGUAY

Alfredo Godoy
godoyalfredo@hotmail.com
Instituto de Computación, Facultad de Ingeniería, Universidad de la República
J. Herrera y Reissig 565, Montevideo, URUGUAY

Louis Petingi
petingi@postbox.csi.cuny.edu
Computer Science Department, College of Staten Island, City University of New York
2800 Victory Boulevard, Staten Island, N.Y. 10314, USA

Abstract:

The topology of a communication network with end-to-end maximum delay constraints may be modeled by an undirected graph G=(V,E), where the node-set V represent the sites of the network, and the edge-set E corresponds to the network communication links. If links are subject to random, independent failures, we say that the network is in operational state if there exist paths of bounded length D between each pair of terminal nodes K. The probability of operation of the graph is called the Diameter-constrained Reliability R(G,K,D), a measure that generalizes the classical network reliability.

There are different variants on the problem of designing a topology of a network with high reliability. One of them is to characterize topologies with the highest reliability, among all competing topologies with the same number of nodes and edges, when paths of length D should connect a given number of terminal nodes |K|, and where links are assigned a unique reliability and cost. In this paper we present the solution for this optimization problem when |K| =2 and D=2.

Moreover, in this paper we present the more general case in which different link reliabilities and costs are allowed, and the problem is to characterize topologies with the highest reliability, subject to a total budget (an upper bound on the sum of the costs of links included in the network). This problem can be tackled using heuristic methods like Genetic Algorithms and GRASP. Experiments with those two methods show a better performance of Genetic Algorithms when complex topologies are under consideration.

Key words: graphs, communication networks, diameter-constrained network reliability, network synthesis, optimization, genetic algorithms, GRASP, metaheuristics, heuristics.

1. Introduction

Let us consider a communication network with transmission delays at each a link (or node), and where any two sites s and t can communicate if the total transmission time between them is at most D times the individual delay (i.e, if there is a path connecting s and t of at most D links). We will represent the network by an undirected, connected graph G = (V, E, K) consisting of a set of nodes V, a set of links E and a set of terminals K (a fixed subset of V, representing the sites which must communicate). We will suppose that nodes do not fail, but each link *e* is assigned an independent probability of failure q_e (called link unreliability). If we consider all nodes and only the operational links, we have a random subgraph of G. We say that the network is in operational state if for each pair of terminal nodes, we can find a path between them having at most length D in this random subgraph. The probability of this random event, which we will denote by R(G,K,D), is called the *diameter-constrained network reliability* measure (Petingi and Rodriguez, 2001), an extension of the classical reliability measure (Colbourn, 1987).

As real networks are subject to failures, the diameter-constrained reliability can be useful in different contexts. For example, this measure gives an indicator of the suitability of an existing network topology to support good quality voice over IP applications between a pair of terminals. In the case of a videoconference, we take K to be the set of the participating nodes, and the diameter-constrained reliability gives the probability that we can find short enough paths between all of them. Another potential case of interest are a number of protocols which, in order to avoid congestion by looping data, assign a timeout date or a maximum number of hops to each data packet, or to control information. In this case, the diameter-constrained unreliability (the complement to one of the reliability) gives the probability that, due to failed links, there are some nodes of the network that are not reachable by using these protocols.

When building a new network (or modifying the topology of an existing one), one of the possible objectives is the maximization of the reliability as a measure of the quality of service. We will discuss here different network design problems, which arise when trying to optimize this measure. Section 2 presents a formulation for the diameter-constrained reliability in terms of the paths of the network. Section 3 formulates one variant of the design problems, called the network synthesis problem. Section 4 discusses a more general network design formulation, and presents two heuristic methods, Genetic Algorithms and GRASP, which can be applied to find an approximate solution. Section 5 presents some experimental results. Finally, Section 6 presents conclusions and directions for future work.

2. A path-based formulation for diameter constrained reliability

In this section, we develop a formulation for the diameter-constrained reliability based on the paths of the network. We will use the following notations:

- G=(V,E,K): an undirected network topology
- V={1,...,n}: the node-set of G
- E={e_1,...,e_m}: the link-set of G
- K: the terminal set of G
- m,n,k: the number of [links, nodes, terminals] of G
- x_e: state of the link e; we take state 1 when the link e is up (operational), and state 0 when it is down (failed). Sometimes we will denote a link by its extremities, for example e=(s,t); in this case x_{st} will be an alternative notation for the state of link e
- r_e=Pr(x_e=1): operating probability of link e (also called the elementary reliability of e)
- q_e=Pr(x_e=0)=1-r_e: failure probability of link e (also called the unreliability of e).

Suppose we have a network G=(V,E,K), and we want to compute its diameter-constrained reliability R(G,K,D), for a given value of D. For any pair of nodes s, t belonging to K, we define $P_{st}(d)$ as the set of paths between s and t, of length at most d (with length measured as the number of links of the path). We will denote by E(p) the event in which all the links in a path p are operational. The probability of event E(p) is the probability of finding all the links composing the path p in operational state, and by the hypothesis of independence between the states of the links, it can be computed as the product of the links reliabilities: $\Pr(E(p)) = \prod_{e \in p} r_e$.

The diameter-constrained reliability measure can then be expressed in function of these events:

$$R(G,K,D) = \Pr\left(\bigcap_{s,\, t \in K} \left(\bigcup_{p \in Pst(D)} E(p) \right) \right).$$

In order to illustrate this formula, we will take the small network shown in Figure 1 (which is known in the literature as the bridge network). We will suppose that we want to establish a connection of maximum path length 2 between s and t (i.e, the terminal set is K={s,t}, and D=2).

In this example, there are only two paths of length at most 2 between s and t, which will be included in the set $P_{st}(D) = \{(e_1,e_2), (e_4,e_5)\}$. The paths (e1,e3,e5) and (e4,e3,e2), which connect s and t, are not considered, because they exceed the length bound. For this network, the paths are independent, and then we can easily compute the diameter-constrained reliability:

$$R(G,\{s,t\},2) = \Pr\Big(E((e_1,e_2)) \bigcup E((e_4,e_5))\Big) = r_{e_1} r_{e_2} + \left(1 - r_{e_1} r_{e_2}\right) r_{e_4} r_{e5}.$$

When we take an arbitrary network, the events corresponding to the paths are neither independent from each other, nor disjoint, so this formula is not always suitable for direct computation of the reliability. Nevertheless, one important consequence is that for any link $e \in E$, if e it is not used in any path between terminals of length at most D, then it is irrelevant for computing R(G,K,D). In this case, it is possible to reduce the graph, eliminating e, and obtaining a new network G-e which will have the same reliability as the original one: R(G,K,D)=R(G-e,K,D). In the case of the bridge network, link e_3 does not belong to any of the relevant paths, and can be deleted; in Figure 1 we see the resulting topology.

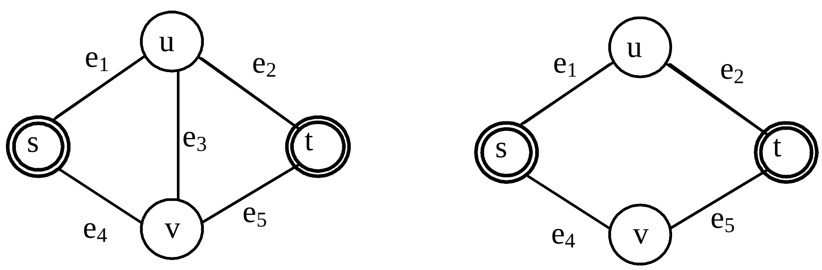

Figure 1: the bridge network before and after eliminating irrelevant link e_3

3. Network synthesis problem

In some contexts, it is useful to consider that all links have unitary costs, and have the same reliability r. If we know the terminals K, and we have a total budged (expressed as an upper bound in the number of nodes and links that can be employed), an interesting problem is to find the topology that maximizes network reliability. This is the *problem of characterizing graphs with maximum reliability among all graphs with n nodes and m links,* sometimes known as the network synthesis problem (Satyanarayana et al, 1992).

This is a difficult problem in general, but can be treated when K = {s,t} and D=2. We observe that in this case, the only relevant paths are those belonging to $P_{st}(2) = \{(s,t)\} \cup \{(s,i,t),\ i \in V \backslash \{s,t\},\ (s,i) \in E, (i,t) \in E\}$, which are disjoint, and they can be represented graphically as parallel paths. As the path (s,t) has larger reliability and smaller cost than the other paths, it will always be included in the optimal topology. The optimal topology will have a direct (s,t) link, and additional h=min(floor((m-1)/2),n-2) paths of length 2 connecting s and t (these paths contain h additional nodes). Figure 2 represents such a network. The reliability of this series-parallel structure can be easily evaluated, giving the following formula (see Cancela and Petingi (1991)):

$$R(G,\{s,t\},2) = r + (1-r) \sum_{i \in V \backslash \{s,t\}} r^2 \prod_{j<i,\, j \in V \backslash \{s,t\}} \left(1-r^2\right) = r + (1-r) r^2 \sum_{j=0}^{h-1} \left(1-r^2\right)^j = r + (1-r) r^2 \frac{1-\left(1-r^2\right)^h}{1-\left(1-r^2\right)} = 1 - (1-r)\left(1-r^2\right)^h .$$

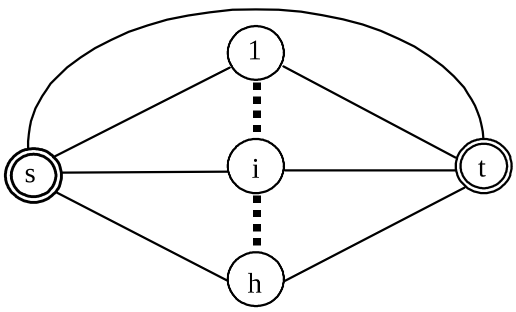

Figure 2: Solution of (n, m) synthesis problem with K={s,t}, D=2,
where h=min(floor((m-1)/2,n-2).

4. Metaheuristics for network design problem

We consider now the design of a network topology taking into account both reliability and budget constraints, where links may have different costs and reliabilities. In general, given a fixed set of nodes V, two terminals s,t, and a set of feasible links E, where any possible link $e=(x,y)$ in E has associated cost c_e and reliability r_e, we can look at (at least) two different design problems, consisting in choosing a subset E* of E in order to either maximize the reliability R(G,{s,t},D), given a total budget B, or minimize cost, given a reliability constraint.

We can formalize the first of these network design problems as follows:

$$\begin{cases} \max_{E^* \subseteq E} R\left(G\left(V,E^*\right),\{s,t\},D\right) \\ \quad s.t. \sum_{e \in E^*} c_e \leq B \end{cases} .$$

This kind of combinatorial optimization problem, is in general, difficult to solve exactly. An alternative instead is the application of metaheuristics, which can give good results in reasonable execution times.

We considered different metaheuristics: Genetic Algorithms, Ant Systems, GRASP, Neural Nets, Simulated Annealing, and Tabu Search. Among these, Genetic Algorithms and GRASP were chosen for implementation, on the basis of criteria like clarity, existing documentation, flexibility, and ease of implementation of the method. In addition, the fact that one is population based, and the other based on Local Search, make them representatives of wider classes of heuristics.

For the Genetic Algorithm, the implementation follows the lines of this simple pseudo-code:

```
Procedure GeneticAlgorithm (OriginalNetwork *I*, GSolucionGenetic *O*);
Input: Original network G
Output: Most reliable network found G*
Begin
    Generation=1
    P=GenerateInitialPopulation(G)
    While Generation<MaxNumberofGenerations do
        Mutate(P,percentageMutation, percentageStrongMutation);
        ReproduceandSelect(P, maxPopulationSize, eliteSize);
        Generation = Generation+1
    End while
    G* =BestIndividual(P)
    Return G*
End
```

Each individual in the population represents a network topology; the corresponding chromosome contains the list of the links that make up the topology (the list of nodes are the end-points of the links).

The fitness of the individual is the Diameter-constrained Reliability of the corresponding network, and as the Diameter-constrained reliability is a generalization of the classical reliability, and the latest was shown to be NP-hard (Ball 1979, Ball 1986), this measure is estimated by a Monte Carlo method, within a given precision.

We define two mutation mechanisms. One is called “strong mutation”, and corresponds to taking the complementary graph with respect to the set of feasible links. If G’=(V,E’) is the selected individual, and E is the set of feasible links, the mutated individual G’’=(V, E/E’), is then augmented with new paths made up from remaining and already employed links. This procedure allows the increment the reliability, and it is repeated until the total budget B has been used. The “weak mutation” corresponds to randomly eliminating some of the paths of G’, and augmenting the resulting network with feasible paths up to budget B.

The selection takes a certain number of the best individuals in the current population for creating the new one; and fills up the rest of the new population by new individuals created by reproduction.

The reproduction mechanism takes two parents G’ and G’’, chosen at random, and selects a list of paths of length smaller than D from s to t in both individuals, such that the cost of each list is a randomly fixed percentage of total budget B. A new individual G’’’ is then generated, which has in its chromosome all links of both list of paths, and is eventually augmented with more paths up to budget B. The best among the two parents and the new individual are then included in the future population.

The GRASP procedure is shown in the following pseudocode:

```
PROCEDURE GRASP();
Input: Original network G
Output: Most reliable network found G*
Begin
        Iteration=1
        While Iteration<MaxNumberofIterations do
                        S=BuildRandomGreedySolution(G)
                        S*=GRASPLocalSearch(S)
                        If  S* more reliable than G*
                                G*=S*
                        End if
        End while
        Return G*
End
```

As in the case of the GA, the solutions (network topologies) are represented by the list of links of the network topology, including an auxiliary list of paths of length less than D (this list is not exhaustive, and its main purpose is to simplify the local search procedure). At each iteration, a topology is built by a randomized greedy procedure, which iteratively selects links according to their reliabilities, while trying to build a path from s to t, using a random portion of budget B.

The Local Search procedure implements an iterative X-change operation between pairs of paths of the network built in the previous step. This operation, commonly used in heuristics for the Traveling Salesman Problem and similar problems, consists in randomly selecting two disjoint paths, selecting a link (x,y) of the first path and a link (x’,y’) of the second one, deleting those links and adding links (x,y’) and (x’,y), which effectively reconnect the paths. The variant, here implemented, does not delete links (x,y) and (x’,y’), because they may be present at other feasible paths. These operations are accepted when they increment the reliability and the procedure is iterated until no further improvement is possible.

5. Experimental results

To study the performance of GA and GRASP we made two sets of experiments. In the first one, the algorithms were run on problems corresponding to the network synthesis problem discussed in Section 3. This allowed us to check the solutions found by these methods and comparing them with the optimal topologies that were analytically derived by the synthesis problem. Six instances were chosen by randomly sampling the number of nodes and links; the parameters (n,m) were (10, 17), (25, 47), (55, 107), (68,133), (80, 157), (91, 189). Both algorithms found the optimal solution for the six cases checked, validating in this way the correctness of the their design and implementation.

In the second experiment, we compared the performances of GA and GRASP on a set of randomly generated scenarios. To cover a wide number of different cases, the scenarios were generated following different scenario profiles. The scenario profiles are defined on the basis of the problem definition parameters: number of nodes and links, bound on the path length, total budget. For each of these parameter, we identify some predefined intervals

- Number of nodes and links: three intervals were identified for the number of nodes, small graphs, from 10 to 40 nodes; medium graphs, from 40 to 70 nodes, and large graphs, from 70 to 100 nodes. The number of feasible links is twice the number of nodes.

- Bound on the path length between the terminals s and t: three intervals were defined, relative to the number of nodes n of the scenario; these intervals are (2,n/3), (n/3,2n/3), (2n/3,n-1), corresponding to short, medium and long paths allowed.
- Total budget: this is the upper bound on the total cost allowed to build the network. The three intervals defined correspond to tight, medium and large budget.

We have three criteria, each having three levels; this adds up (in fact, multiplies) to twenty seven different scenario profiles. For each scenario profile, five different instances were generated, resulting in a total of 135 test cases. The following table shows the codes used to identify each scenario profile (we will not need to identify individual test cases, as we will only present average results).

		Path length bound/ Total Budget								
		Lo/Lo	Lo/Med	Lo/Hi	Med/Lo	Med/Med	Med/Hi	Hi/Lo	Hi/Med	Hi/Hi
Number of nodes	Small	S1	S2	S3	S4	S5	S6	S7	S8	S9
	Medium	M1	M2	M3	M4	M5	M6	M7	M8	M9
	High	H1	H2	H3	H4	H5	H6	H7	H8	H9

The experiments were run over the 135 test cases. We averaged the results over the five instances in each scenario profile. Figure 3 presents the average reliability obtained by GRASP and GA; while Figure 4 shows the mean execution times required. The number of iterations for the GRASP method was equal to the product between the number of generations and the population size for the GA method, so that both algorithms explore the same number of potential solutions.

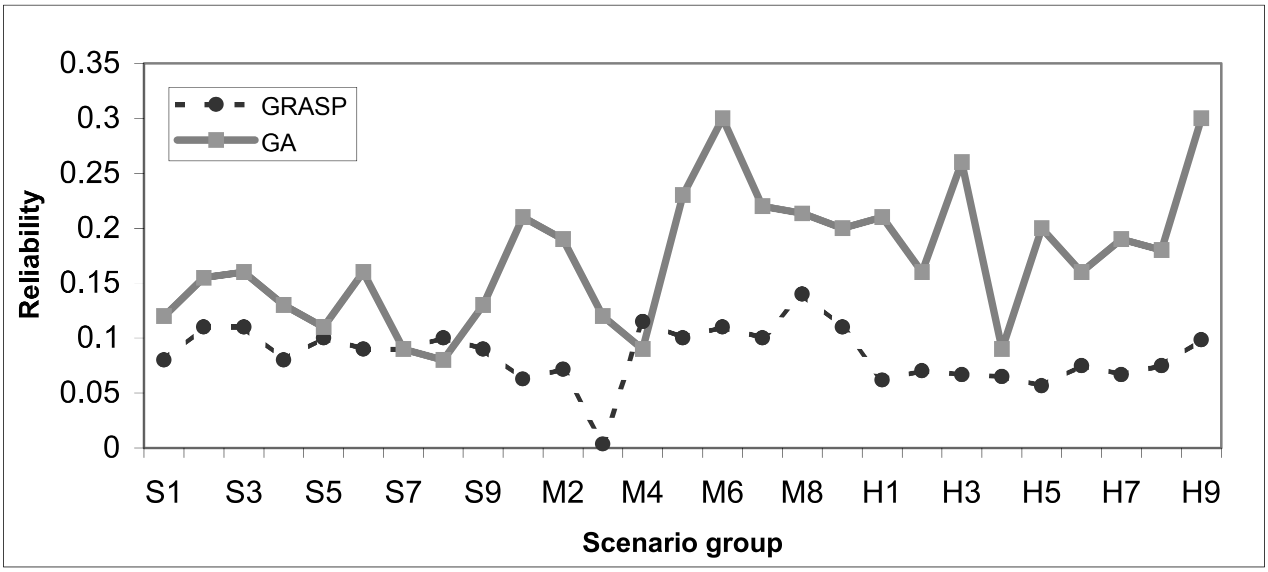

Figure 3: Average of reliabilities of solutions found for each scenario profile group.

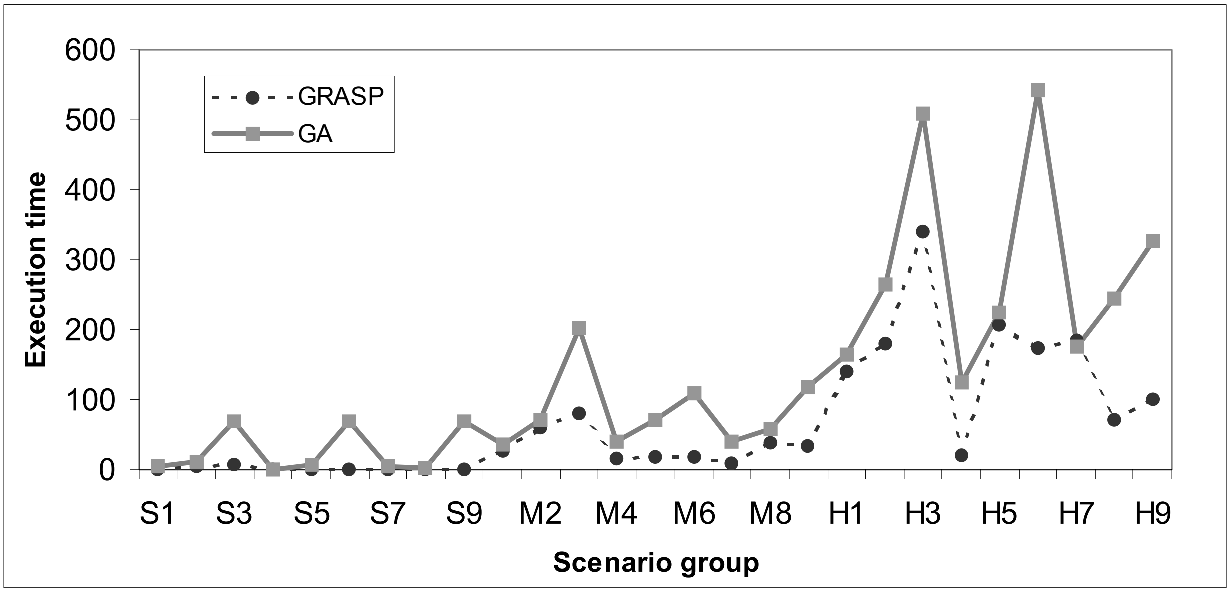

Figure 4: Execution times for each scenario profile group.

As it can be seen, in almost all the cases GA provides substantially better results than GRASP; the difference gets larger for the scenarios with high number of nodes and links (larger networks). Execution times of GA are also somewhat larger, but with a less pronounced difference. We can find a cyclic pattern of period 3 appearing; first a small execution time, then a somewhat larger one, and then an even larger one, followed by a smaller one and the pattern restarts. If we look at how the scenarios were numbered, this corresponds to execution times increasing for scenarios with larger budgets, being all other factors the same. This effect is not totally unexpected: as the solutions will include more links and paths when budgets are larger, and both the GA and GRASP mechanisms were defined in terms of finding, adding, or exchanging paths, they will then require more computing time at each single generation or iteration.

6. Conclusions

This paper discusses different methods for designing the topology of a network in order to maximize a certain reliability measure, namely the diameter-constrained network reliability R(G,K,D). This reliability measure is particularly useful for real-time services, such as voice over IP or videoconference applications, because it allows to model performance objectives that restrict the maximum length of a path in the network.

We have presented in first place the network synthesis problem, where links are identical in cost and reliability, and there is a bound in the number of links and nodes that can be included in the network.. The optimal topology and its reliability value is given for the particular case where D=2 and K={s,t}.

For the more general network design problem, where links have different costs and reliabilities, we have looked at two different metaheuristics for obtaining approximate solutions: Genetic Algorithms, and GRASP. Both were implemented, and experiments on a set of random scenarios show that Genetic Algorithms provide consistently better results. Both methods employed standard Monte Carlo method for evaluating the reliability of the network topologies; this is somewhat time consuming, and also leads to precision problems specially when the network reliabilities are very high (close to 1). An alternative would be the to employ Monte Carlo with variance reduction techniques such as those studied in other reliability problems by Cancela and El Khadiri (1998) and Cancela and Urquhart (2002).

References

Ball, M. (1979). Computing network reliability. Operations Research, 27:823-836.

Ball, M. (1986). Computational complexity of network reliability analysis: An overview. IEEE Transactions on Reliability, R-35(3):230-239.

Cancela, H. and El Khadiri M. (1998) Series-parallel reductions in Monte Carlo network reliability evaluation. IEEE Transaction on Reliability, 47(2): 159-164.

Cancela, H. and Petingi, L. (2001). Network reliability evaluation for delay constrained services. ITU/ITC 17 Developing Countries Workshop, International Teletraffic Congress, ITC 17, Salvador, Bahia, 2-6 diciembre 2001.

Cancela, H. and Urquhart, M. E. (2002). RVR simulation techniques for residual connectedness network reliability evaluation. IEEE Transactions on Computers 51(4): 439-443.

Colbourn, C. (1987). The Combinatorics of Network Reliability. Oxford University Press, New York.

Petingi L., Rodriguez J. (2001). Reliability of Networks with Delay Contraints. Submitted to Congressus Numerantium.

Provan, J. and Ball, M. (1983). The complexity of counting cuts and of computing the probability that a graph is connected. SIAM Journal on Computing, 12:777-788.

Rubino, G. (1996). Network reliability evaluation, chapter 11 of State-of-the art in performance modeling and simulation. Bagchi, K. and Walrand, J., editors. Gordon and Breach Books.

Satyanarayana A., Schoppmann L., Suffel C. (1992). A reliability-improving graph transformation with application network reliability. Networks, 22: 209-16.