Optimal Link Assignments for All-Terminal Network Reliability

Constantine Stivaros
Computer Science Dept.
Fairleigh Dickinson University
Madison, NJ 07940

Klaus Sutner
Computer Science Dept.
Stevens Institute of Technology
Hoboken, NJ 07030

April 16, 2003

Abstract

Probabilistic networks whose links are subject to failure have been extensively studied in the literature. Their reliability is regarded as the network's ability to continue operation after failures have occurred. We consider the *all-terminal reliability* measure, defined as the probability that the network remains connected after failures. Previous work on this model assumes uniform failures probabilities for all links. In this paper, we discuss the design problem of assigning given distinct probabilities to the links of a network so as to maximize the reliability of the system. We determine optimal link assignments for some sparse networks such as multi-rings, and show that the problem is **NP**-hard in general.

1 Introduction

One of the most important functions of a communications network is to carry messages as reliably as possible. Such a network is regarded "reliable" if a message entered into the net at one terminal can actually be routed to the required destination. This corresponds to the requirement that the components of the network (nodes or links) are connected and, thus, the reliability of communication becomes dependent on the network's underlying structure. Given that components may be subject to failure in a random and unpredictable manner, the network's reliability is then the probability that the network remains connected after failures have occurred. The aim of reliability studies is two-fold: compute the reliability measure of an existing network (*analysis aspect*) and/or design a probabilistic network that will tolerate failures in the best possible way, i.e., maximize certain reliability parameters (*synthesis aspect*).

Numerous network reliability models have emerged in the literature. Most of them focus on the vulnerability of the links rather than nodes; the latter is a relatively new subject and many results can be found in [11, 4, 10], for both synthesis and analysis problems.

The most common link-failure models, as motivated by practical applications, are the *K-terminal* and *All-terminal* reliability models. These models are concerned with the connectedness of either a subset K of the nodes, or all nodes, when links fail independently while nodes are assumed perfectly reliable. Excellent surveys on these models can be found in [3, 8, 9] where optimal networks are reviewed and computation methods are described.

In all models, a network is represented by an undirected simple graph whose points are the nodes of the network and whose edges are the connecting links. The design of an "optimal network" amounts to constructing a graph that maximizes certain reliability parameters. All existing models so far have considered only uniform probabilities for the failure of edges.

In this paper, we extend the scope of optimal design for the all-terminal reliability measure. We consider arbitrary probabilities for the operation of the links. We will think of theses probabilities as being supplied by the manufacturer of the communication medium. Given the network topology, we are faced with the problem of associating the edges of the graph with actual links; we refer to this problem as the *Optimal Link Assignment* problem. As we will show, optimal assignments are difficult to compute even for sparse classes of networks, although their reliabilities can be computed in polynomial time.

The assignment problem for edge-failures originated in recent work of these authors on the node-failure case, see [12, 13]. The reliability measure used in these references is the *residual node connectedness reliability*, the probability that the operating nodes of the network remain connected. The *Optimal Node Assignment* problem is directly analogous to the Optimal Link Assignment problem. In [12] optimal assignments are computed for classes such as Bus/Path, Ring/Cycle, Threshold and a dense class of networks. It was shown in [12] that the node assignment problem is **NP**-Complete for an arbitrary graph and a given probability vector.

In the following sections, we present and define the optimal link assignment problem for all-terminal reliability; we develop combinatorial tools for analysis, and exhibit solutions for sparse classes of networks. In the last section, we show that the computational complexity of the problem is hard.

2 The Network Model

We consider networks having perfectly reliable nodes but links prone to statistically independent failures; furthermore, assume that the operating probabilities of the links are given in a sorted vector $\vec{P}$. As it is customary, we refer to a network by its underlying undirected graph $G = \langle V, E \rangle$. Throughout this paper we assume $|V| = n$ and $|E| = e$.

A *state* of G is the subgraph induced by a set Θ of operating edges. If Θ induces a

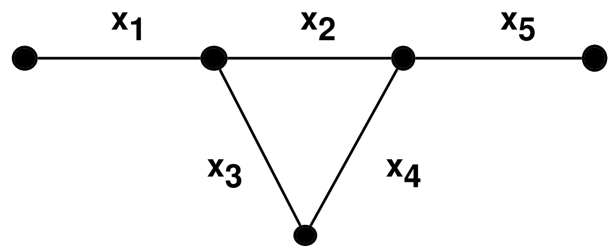

Figure 1: A small unicyclic graph used in the sample calculation below.

connected subgraph, it is called an *operating state*; we denote the set of all operating states of G by $\Omega(G)$. The *all-terminal reliability* $R(G)$ of G is the probability that G is connected; in other words, the probability that the current state of G is an operating state. Thus, computation of this measure involves the evaluation of the probabilities of all operating states of G.

In the uniform failure case, where all links operate with the same probability p, this reliability measure is referred to as the *reliability polynomial* of G:

$$R(G,p) = \sum_{i=n-1}^{e} s_i \, p^i (1-p)^{e-i}$$

where s_i is the number of operating states of size i.

For the general case, suppose we are given a sorted vector of probability values $\vec{P} = (p_1, \ldots, p_e)$ of length e.

Definition 2.1 *An* assignment *for G and $\vec{P}$ is a bijection $\gamma : E \to [e]$.*

Here $[e]$ denotes the set of integers $\{1, \ldots, e\}$. We think of an assignment γ as associating edge x with a link of operating probability $p_{\gamma(x)}$. Then the probability of a state Θ of G and $\vec{P}$ under assignment γ is:

$$Pr(\Theta, \vec{P}, \gamma) = \prod_{x \in \Theta} p_{\gamma(x)} \prod_{x \notin \Theta} (1 - p_{\gamma(x)})$$

The all-terminal reliability of G and $\vec{P}$ under assignment γ becomes:

$$R(G, \vec{P}, \gamma) = \sum_{\Theta \in \Omega(G)} Pr(\Theta, \vec{P}, \gamma).$$

For example, consider the unicyclic graph G from figure 1.

This graph has only four operating states, Θ_0 when no edge has failed, Θ_2 when edge x_2 has failed and similarly Θ_3, Θ_4, for the failure of edges x_3, x_4 respectively.

Consider the ordered probability vector $\vec{P} = (0.88, 0.91, 0.91, 0.95, 0.98)$ and the assignment γ given by

$$x_1 \mapsto p_4,\ x_2 \mapsto p_2,\ x_3 \mapsto p_5,\ x_4 \mapsto p_1,\ x_5 \mapsto p_3.$$

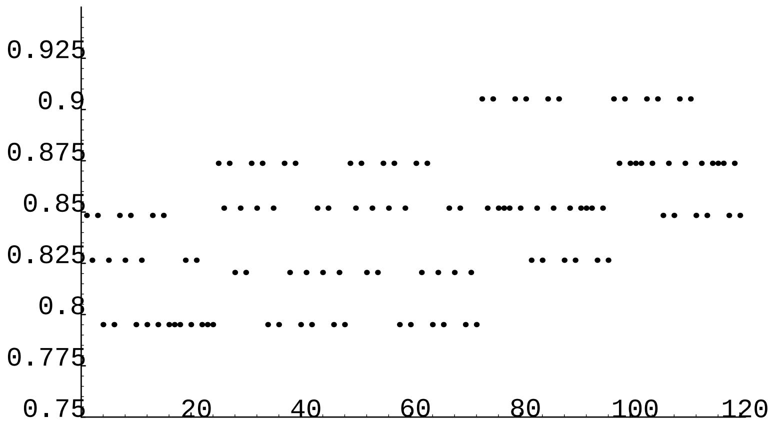

Figure 2: All 120 possible network reliabilities for the unicyclic graph G from figure 1.

The reliability of whole network under assignment γ is

$$R(G,\vec{P},\gamma) = \Theta_0 + \Theta_2 + \Theta_3 + \Theta_4 = p_1p_2p_3p_4p_5(1 + \frac{1-p_2}{p_2} + \frac{1-p_5}{p_5} + \frac{1-p_1}{p_1}) = 0.85.$$

Among all assignments for G, an *optimal assignment* γ will maximize the all-terminal reliability measure for G:

Definition 2.2 *Given G and an ordered probability vector $\vec{P}$, an* optimal link assignment (OLA) γ *is an assignment such that*

$$R(G,\vec{P},\gamma) \geq R(G,\vec{P},\gamma'),$$

for all link assignments γ'.

In the sample graph above, among the $120 = 5!$ many possible assignments for G, an optimal assignment γ is given by:

$$x_1 \mapsto p_4,\ x_2 \mapsto p_1,\ x_3 \mapsto p_2,\ x_4 \mapsto p_3,\ x_5 \mapsto p_5.$$

As we will show below, all other optimal assignments can be obtained be permuting the values for the end-edges x_1, x_5, and for the cycle edges x_2, x_3, x_4. Hence, there are 12 optimal assignments for G. The reliabilities produced by all possible assignments are shown in figure 2. We will actually show that γ is uniformly optimal in the sense that the same assignment is optimal for all ordered probability vectors $\vec{P}$.

3 Reliability preserving reductions

In this section we describe a few combinatorial tools that are useful in computing the reliability of a network with a given assignment. To lighten notation, we write p_x instead of $p_{\gamma(x)}$.

a. The Factoring Theorem

The factoring theorem evaluates the reliability of a network G by considering the operating status of an edge. For this, we need to define the *deletion* and *contraction* of an edge x of G. If x is an edge from u to v, then $G - x$ is the subgraph obtained from G by deleting the edge x; note that deletion of x does not imply deletion of u, v. In G, contracting an edge x involves deleting x and merging its end-vertices u, v into a "super-vertex" that is given the adjacency of both u, v in G. The graph obtained from G by contracting x is denoted as G/x.

A state of the network is a collection of edges and therefore its operating probability is a "compound" event. It consists of more elementary events, the survival or failure of each individual edge. Thus, the operating status of an edge x partitions the states of G into two sets and the all-terminal reliability of G can be written as:

$$R(G) = p_x R(G|\ x \text{ is functional }) + (1 - p_x) R(G|\ x \text{ is not functional })$$

The above equation is known as the *pivotal decomposition* [1, 2]. Perhaps the earliest to this equation was made by Moskowitz in [6]; it is used there to derive the following simple but significant topological transformation for graphs:

$$R(G) = p_x R(G - x) + (1 - p_x) R(G/x) \qquad (1)$$

This is often called the *factoring theorem* and has been instrumental in constructing recursive algorithms for the computation of $R(G)$, see [8].

The next three reduction rules can be verified using the the factoring theorem.

b. Parallel Reduction

Let $x = \{u, v\}$ and $y = \{u, v\}$ be two parallel edges in a graph G. A direct connection between u, v exists if at least one of the two edges operates; thus, a parallel reduction replaces x and y with a single edge $z = \{u, v\}$ such that $p_z = p_x + p_y - p_x p_y$ ($= 1 - q_x q_y$, where $q = 1 - p$). If G' is the graph obtained from G, then $R(G) = R(G')$.

c. Series Reduction

Let $x = \{u, v\}$ and $y = \{v, w\}$ two edges of G adjacent at a degree–2 vertex v. Then a series reduction replaces x and y by a single edge $z = \{u, w\}$ such that $p_z = \frac{p_x p_y}{p_x + p_y - p_x p_y}$. If G' is the graph obtained from G, then $R(G) = (1 - q_x q_y) R(G')$.

Note that this reduction–also referred to as degree two reduction in [7]–is different from the standard series reduction, since in this model there are no vertex failures.

d. End-Reduction

Let $x = \{u, v\}$ be an edge of G, and u an end-point. Clearly, any operating state of G must include x. Thus, an end-reduction deletes vertex u and edge x producing a new graph G'. Then $R(G) = p_x R(G')$.

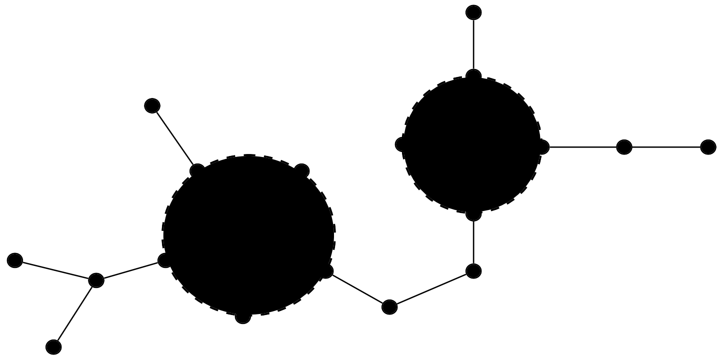

Figure 3: A multi-ring network with two cycles and 30 operating states. The essential edges are drawn solid.

4 Optimal Link Assignments

We now introduce a class of sparse graphs that includes unicyclic graphs. A connected graph G is a *multi-ring network* if all edges of G lie on at most one cycle. Thus, we require cycles to be edge-disjoint but not necessarily vertex-disjoint. One can determine in linear time whether a graph G is a multi-ring network, using the following recognition algorithm.

Step 1 Decompose the graph into its biconnected components $B_1, \ldots, B_l$, $B_i \subseteq E$.

Step 2 For each of these components, test if either $|B_i| = 1$ or if the subgraph induced by B_i is a simple cycle (i.e., all points in the subgraph induced by B_i have degree 2).

Step 3 Return "Yes" if all biconnected components pass the test, and "No" otherwise.

Note that the algorithm can easily be modified to produce a list of all the edge-disjoint cycles in the graph, if the graph is indeed a multi-ring. This will be useful later in the computation of the reliability of a graph with a given assignment. The decomposition into biconnected components as well as the test can clearly be carried out in linear time. Figure 3 shows a typical multi-ring network with two cycles, a *by-cycle*.

Given an multi-ring network G, a probability vector $\vec{P}$ and an assignment γ we can compute the reliability $R(G, \vec{P}, \gamma)$ as follows. Let $C_1, \ldots, C_m$ be an enumeration of the cycles of G where we think of each C_i as a set of edges. We attach a dummy "edge" $\perp$ to each cycle and define

$$\mathcal{C} = (C_1 \cup \{\perp\}) \times (C_2 \cup \{\perp\}) \times \ldots \times (C_m \cup \{\perp\}).$$

Fix some assignment γ and assume for the sake of simplicity that edge x is assigned probability p_x. We may safely assume $p_x > 0$ for all edges x. Also, define $r_x := (1 - p_x)/p_x$ and for the dummy edge $\perp$ set $r_\perp := 1$. We can now parameterize the operating states as follows. For $(x_1, \ldots, x_m) \in \mathcal{C}$ set

$$\Theta(x_1, \ldots, x_m) := E - \{\, x_i \mid x_i \neq \perp,\ i = 1, \ldots, m \,\}.$$

Then Θ is an operating state iff $\Theta = \Theta(x_1, \ldots, x_m)$ for some $(x_1, \ldots, x_m) \in \mathcal{C}$. Note that the number of operating states is therefore $\prod_i |C_i| + 1$, which is not polynomially bounded in the size of the graph. Thus, brute force computation of the reliability of a multi-ring network would not yield a polynomial time algorithm.

However, we can compute the reliability of G under γ in linear time as follows. We use P_E as an abbreviation for $\prod_{x \in E} p_x$.

$$\begin{aligned} R(G, \vec{P}, \gamma) &= \sum_{\Theta \in \Omega(G)} P(\Theta) \\ &= \sum_{(x_1, \ldots, x_m) \in \mathcal{C}} P(\Theta(x_1, \ldots, x_m)) \\ &= P_E \sum_{(x_1, \ldots, x_m) \in \mathcal{C}} \prod_{i=1,\ldots,m} r_{x_i} \\ &= P_E \prod_{i=1,\ldots,m} \sum_{x \in C_i \cup \{\perp\}} r_x \end{aligned}$$

To see that the last two lines are equal, note that one can expand the product of sums as

$$P_E \sum_{\substack{f:[m] \to E \cup \{\perp\} \\ f(i) \in C_i \cup \{\perp\}}} \prod_{i=1,\ldots,m} r_{f(i)},$$

which can be rewritten as the sum of products in line 3 above. Hence, we have essentially established the following theorem.

Theorem 4.1 *Given a multi-ring network G, a probability vector $\vec{P}$ and an edge assignment γ, we can compute the reliability of G under γ in a linear number of arithmetic operations, and in polynomial time overall.*

Proof. From the previous remarks, we can compute the edge-disjoint cycles of G in linear time. Hence, we can compute the sums $\sum_{x \in C \cup \{\perp\}} r_x$ in a linear number of arithmetic operations. Likewise, we can determine $R(G, \vec{P}, \gamma)$ by computing a product of these sums in a linear number of arithmetic operations. It is easy to check that the number of bits needed to represent the numbers that occur during the computation is $O(n(\log n + \alpha))$ where α is the maximum number of bits in the input data. Hence the overall computation is still polynomial time. □

In order to determine an optimal link assignment, first note that certain edges such as bridges or end-edges are part of each operating state. Thus, it is intuitively clear that these edges should be assigned the highest available probabilities. To make this precise, define the set I of *essential* edges for G to consist of those edges that appear in every operating state of G. Equivalently, the set I could be defined as the intersection of all spanning trees of G. For example, for the bi-cycle in figure 3, the essential edges are drawn solid, the others are dashed.

Definition 4.1 *A set of edges $E' \subseteq E$ is* assignment-invariant *if for any two assignments γ, γ' for G and $\vec{P}$ that differ at most on E', $R(G, \vec{P}, \gamma) = R(G, \vec{P}, \gamma')$.*

The next proposition is obvious from the definitions.

Proposition 4.1 *I is an assignment-invariant set.*

The next claim establishes the fact that essential edges are more important for the reliability of the network:

Proposition 4.2 *If γ is an optimal link assignment for G, x is an essential edge and y fails to be essential, then $p_{\gamma(x)} \geq p_{\gamma(y)}$.*

Proof. Let γ be an optimal link assignment for G and obtain a new assignment γ' switching the values of x and y. Since $R(G, \gamma) \geq R(G, \gamma')$, we can write:

$$\sum_{\Theta \in \Omega(G)} \prod_{u \in \Theta} p_{\gamma(u)} \prod_{v \notin \Theta} (1 - p_{\gamma(v)}) \geq \sum_{\Theta \in \Omega(G)} \prod_{u \in \Theta} p_{\gamma'(u)} \prod_{v \notin \Theta} (1 - p_{\gamma'(v)})$$

Note that in the above expression the only Θ terms that do not cancel out, are those for which $x \in \Theta$ and $y \notin \Theta$. Therefore we have

$$\begin{aligned} p_{\gamma(x)}(1 - p_{\gamma(y)})(\ldots) &\geq p_{\gamma'(x)}(1 - p_{\gamma'(y)})(\ldots), \\ p_{\gamma(x)}(1 - p_{\gamma(y)}) &\geq p_{\gamma(y)}(1 - p_{\gamma(x)}), \\ p_{\gamma(x)} &\geq p_{\gamma(y)}, \end{aligned}$$

as required. □

Using the last two propositions, we can immediately determine optimal link assignments in unicycles. Note that for a unicycle G, the edges are partitioned into I and $C = E - I$, the essential edges and the cycle edges.

Proposition 4.3 *An optimal link assignment for a unicycle G assigns the $|I|$-many highest probabilities to the essential edges and the remaining probabilities to the cycle edges, in any order.*

Proof. It is clear that the non-cycle edges of G are its essential set I. From propositions (4.1) and (4.2), any optimal assignment must attach higher probabilities to the essential edges I than to the cycle edges C. We claim that both I and C are assignment-invariant, so that any permutation of values within I and C does not affect the reliability of the network. But by theorem 4.1, or, equivalently, by applying the reduction rules of section 3, the reliability of the network is

$$P_E \sum_{x \in C \cup \{\perp\}} r_x,$$

and this term is clearly invariant with respect to the permutations under consideration. □

5 Complexity Considerations

From the previous section, we know that the reliability associated with an assignment for a multi-ring network is always computable in polynomial time. Moreover, if there is at most one cycle in the network, then we can actually compute an optimal assignment, essentially by sorting the probabilities. By contrast, we will now show that already for two cycles it becomes **NP**-hard to determine optimal assignments.

To see this, let us write the two cycles as $A, B \subseteq E$. Using the same notation as above, we have from the theorem:

$$\begin{aligned} R(G, \vec{P}, \gamma) &= R_E \sum_{a \in A \cup \{\perp\}} r_a \sum_{b \in B \cup \{\perp\}} r_b \\ &= R_E \Big(1 + \sum_{x \in A \cup B} r_x + (\sum_{a \in A} r_a)(\sum_{b \in B} r_b) \Big) \end{aligned}$$

Hence, in order to maximize reliability, we have to maximize the term

$$(\sum_{a \in A} r_a)(\sum_{b \in B} r_b).$$

This is easy only in a few special cases. For example, if $|A| = |B|$ and $r_{2i} = r_{2i-1}$, we can assign all the odd-index probability values to cycle A and the others to B. Then

$$\sum_{a \in A} r_a = \sum_{b \in B} r_b = 1/2 \sum_{x \in A \cup B} r_x$$

and this assignment is optimal by a simple calculation. Indeed, an assignment that satisfies the last condition will always be optimal, regardless of whether A and B have the same cardinality. This is the key to the following hardness result. As is customary, our optimization problem has to be rephrased slightly as a decision problem to fit into the standard complexity hierarchies. In our case, consider the following problem.

Problem: Bi-Cycle Optimal Link Assignment Problem (BCOLA)

Instance: A bi-cycle G, a sorted probability vector $\vec{P}$, and a bound c, $0 \leq c \leq 1$.

Question: Is there a link assignment γ such that $R(G, \vec{P}, \gamma) \geq c$?

Theorem 5.1 *Finding optimal link assignments in a multi-ring network is* **NP**-*complete in general, even if the network is required to be a bi-cycle.*

Proof. As we have seen in the last section, the problem is in **NP**: we may guess the proper assignment γ and then verify in polynomial time that in fact $R(G, \vec{P}, \gamma) \geq c$.

To establish hardness, we will show that the Partition Problem is polynomial time reducible to optimal link assignments in bi-cycles; see [5] for a discussion of Partition and a reduction from 3-dimensional Matching that establishes hardness. It is easy to see that the proof shows that Partition remains hard even in the following modified form.

Problem: Constrained Partition Problem

Instance: A list of positive integers $s_1, \ldots, s_t$, a number t' such that $1 \leq t' \leq t$.

Question: Is there a partition of the index set $[t]$ into A and B such that

$$\sum_{i \in A} s_i = \sum_{i \in B} s_i.$$

Assume we are given an instance of Constrained Partition of the form $s_1, \ldots, s_t$ and t'. Set $S := \sum s_i$ and $T := \prod s_i$. Consider a graph consisting of two edge-disjoint cycles A and B of length t' and $t - t'$, respectively, and a probability vector $\vec{P} = p_1, \ldots, p_t$ given by $p_i = 1/(1 + s_i)$. Then G together with $\vec{P}$ and any bound c is a yes-instance of BCOLA iff

$$(\sum_{a \in A} s_a)(\sum_{b \in B} s_b) \geq c/T - S - 1.$$

Now choose the bound c to be $c = T(S^2/4 + S + 1)$. Then we have a yes-instance iff

$$\sum_{a \in A} s_a = \sum_{b \in B} s_b = S/2.$$

But that means that $s_1, \ldots, s_{2t}$ is a yes-instance of Partition. The opposite direction is entirely similar. Moreover, the BCOLA instance G, $\vec{P}$ and c can be constructed from the Partition instance in polynomial time. Hence, BCOLA is **NP**-hard and we are done. □

References

[1] R. E. Barlow, F. Proschan, "*Statistical Theory of Reliability and Life Testing*", Holt, Rinehart, Winston, 1975.

[2] Z. W. Birnbaum, J. D. Esary, S. C. Saunders, "*Multi–component systems, structures and their reliability*", TECHNOMETRICS, vol. 3, pp. 55-77, 1961.

[3] C. Colbourn, "*The combinatorics of network reliability*", Oxford University Press, 1987.

[4] C. Colbourn, A. Satyanarayana, C. Suffel, K. Sutner, "*Computing the Residual Connectedness Reliability for Restricted Networks*", DISCRETE APPLIED MATHEMATICS, vol. 44, pp. 221-232, 1993.

[5] M. Garey, D. Johnson "*Computers and Intractability*", W.H. Freeman and Company, 1979.

[6] F. Moskowitz, "*The analysis of redundancy networks*", AIEE TRANS. COMMUN. ELECTRON., vol. 39, pp. 627-632, 1958.

[7] T. Politof, A. Satyanarayana, L. Tung, "*An $O(n \log n)$ algorithm to compute the all-terminal reliability of $(K_5, K_{2,2,2})$ free networks*", IEEE TRANSACTIONS ON RELIABILITY, vol. 41, No. 4, pp. 512-517, December 1992.

[8] A. Satyanarayana, M. K. Chang, "*Network Reliability and the Factoring Theorem*", NETWORKS 13, pp. 107-120, John Wiley & Son, 1983.

[9] A. Satyanarayana, R. Wood, "*A linear time algorithm for computing the K-terminal reliability in series-parallel networks*", SIAM JOURNAL ON COMPUTING 14, pp. 818-832, 1985.

[10] C. Stivaros, "*The Complete Multipartite Networks are Not Always Reliable*", CONGRESSUS NUMERANTIUM, vol. 88, pp. 193-206, November 1992, Utilitas Mathematica Publishing.

[11] C. Stivaros, C. Suffel, "*Uniformly Optimal Networks in the Residual Node Connectedness Network Reliability Model*", CONGRESSUS NUMERANTIUM, vol. 81, pp. 51-64, December 1991, Utilitas Mathematica Publishing.

[12] C. Stivaros, K. Sutner, "*Computing Optimal Assignments for Residual Network Reliability*", DISCRETE APPLIED MATHEMATICS (under review), North Holland Publ.

[13] C. Stivaros, K. Sutner, "*The Intractability of the Reliable Assignment in Split Networks*", NETWORKS (under review), John Wiley & Son.