

Design of Avionics Network Architecture under a Reliability Constraint

Thanakorn Khamvilai, Aqib A. Syed, John B. Mains, Eric Feron, Louis Sutter, Philippe Baufreton, François Neumann

Abstract—Reliability is an important aspect of operating safety-critical applications of networked systems, such as in avionics. In order to ensure sufficient reliability, one approach is to apply redundancy. However, redundancy may be costly if the degree of redundancy is too high. This paper proposes an optimization-based framework for guaranteeing the desired reliability of any graph-based networked system. It focuses on integrated modular avionics architecture, based on the use of minimally redundant components. This framework consists of two steps. The first is to compute the minimum number of components within the architecture, using a geometric program. The second is to determine the topology with the minimal number of connections between these components, using a signomial program. Finally, the method is illustrated on a small example network and a larger network of the A350 avionics architecture.

Index Terms—reliability, redundancy, optimization, signomial, networked system, integrated modular avionics

I. INTRODUCTION

An integrated modular avionics (IMA) architecture in a commercial aircraft usually consists of several end-system devices (ES), i.e., core processing input/output modules (CPI-OMs), common remote data concentrators (CRDCs), line replaceable units (LRUs), and Ethernet switches, connected in a network scheme called an avionics full duplex switched Ethernet (AFDX) network [1], [2]. In addition to the AFDX network, there are three additional architectures, i.e. ARINC 429, MIL-STD-1553B, and ARINC 629, used in the aircraft [3]. However, they are the federated architecture implemented in aircraft developed before the 2000s and unlikely to continue their service in modern aircraft. Additionally, their network topology is linear, which entails that the reliability may simply be analyzed using an equation for a series system. Thus, the AFDX network, which is applied to modern aircraft like A350, is the sole focus of this paper.

The AFDX network can be considered as a safety-critical networked system, in which its continuity of service is required in the presence of ES or component failures [4]. The capability

of the network to satisfy this requirement is usually quantified by its probability of maintaining its intended function, also named reliability. Network reliability may be evaluated by modeling the system as a probabilistic graph [5], where each vertex represents a single component or subsystem, and each edge represents communication, or the flow of information between vertices. Both vertices and edges are also associated with a component-level reliability, which can be obtained from their specification of failure rate or maintenance schedule. Then, the reliability of a system is interpreted as the probability that a vertex set of interest is properly connected through available edges.

To ensure that the evaluated system reliability is above a given level, more vertices and edges can be added to the network graph. However, having too much redundancy poses impracticalities due to several constraints, for example, the prices of components, the physical distance between two ES constrained by the size of the aircraft, the communication bandwidth of an Ethernet cable, and the aircraft maximum gross weight. Therefore, it is necessary to formulate an optimization problem which minimizes the cost of the network, while guaranteeing the desired reliability.

In this paper, a two-step framework for solving this problem is proposed. The first step is to leverage *Geometric Programming* (GP) [6] to determine the minimal number of redundant components that satisfy the required reliability. Hence, this step is referred to as a redundancy optimization. In the next step, by using *Signomial Programming* (SP), the minimum number of connections among these components can be computed, to form the network architecture, which maintains the same required reliability. This is effectively a topology optimization problem. Both methods are in the form of mixed-integer mathematical programs, which offer benefits of using readily available solvers, e.g., GPKIT [7] and Mosek [8], to determine their globally optimal solution, and providing a flexibility to introduce other relevant constraints, such as a common cause failure [9] or a bandwidth constraint [10]. The justification for the two-step formulation arises from the inherent traceability of the problem. Such a formulation mitigates issues with the unstructured nature of the one-step formulation and subsequent difficulties associated with solving the problem.

Several works on redundancy optimization are studied in [11]. Recent developments in meta-heuristic approaches are presented in [12], [13]; however, these methods cannot guarantee the solution to be globally optimal as opposed to exact methods as in the case of GP. The formulation of GP in the field of reliability has been introduced in [14]–[16]. However,

This effort has been funded in part by SAFRAN, KAUST, and by the National Science Foundation, Grants CNS 1544332 and 1446758.

T. Khamvilai and E. Feron are with King Abdullah University of Science and Technology, Thuwal, Saudi Arabia, {thanakorn.khamvilai,eric.feron}@atkaust.edu.sa

A. A. Syed is with Georgia Institute of Technology, Atlanta, GA, USA, asyed36[at]gatech.edu

J. B. Mains is with Yamaha Motor Corporation, Kennesaw, USA, bmains16277[at]gmail.com

L. Sutter is with Safran Electronics and Defense, Massy, France, sutter.louis[at]gmail.com

P. Baufreton, F. Neumann are with Safran Electronics and Defense, Massy, France, {philippe.baufreton, francois.neumann}@atksafrangroup.com

the formulation in [14] only yields an approximate solution. The formulation in [15] is only valid for a system connected in series. Lastly, although the formulation in [16] solves the previously mentioned issues, it does not consider the cost of establishing connections between redundant components.

For the topology optimization, there also exist meta-heuristic methods [17]–[19] for solving the problem. On the contrary, some exact methods, such as dynamic programming [20], and best-first search approach [21], can provide global optimality, however, they are not as robust to the introduction of new constraints. The proposed method of SP has been applied to many engineering applications, for instance, aircraft design [22], propulsion system sizing [23], circuit design [24], and wireless communication systems [25]. However, its application to topology optimization is seemingly novel.

The remainder of this paper is organized as follows. Section II introduces relevant mathematical background. Section III describes the formulation for the redundancy optimization problem as a mixed-integer GP. Section IV contains the formulation of a mixed-integer SP for a topology optimization problem, and proposes the algorithm and its implementation, to obtain a globally optimal solution. Section V presents an example inspired by the A350 IMA architecture. Section VI summarizes the contributions and discuss the future works.

II. MATHEMATICAL BACKGROUND

A. Network Model

The network model is an undirected graph, $\mathcal{G} = (V, E)$, where V represents a set of vertices, and $E \subseteq V \times V$ represents a set of edges. Let $M = |V|$ and $N = |E|$ be the number of vertices and the number of edges, respectively. Each vertex, $v_i \in V$ has a reliability, $r_i \in [0, 1]$, $i \in \{1, \dots, M\}$ defined as a probability of failure nonoccurrence. Similarly, each edge $e_j \in E$ has a reliability of $r_j \in [0, 1]$, $j \in \{M + 1, \dots, M + N\}$. Depending on the system behavior, the value of r_i and r_j can be either constant or evaluated at a specific point in time. In this paper, the network can remain *functional* in the presence of failures of its vertices and edges, should the vertices which provide different functionality to the network remain connected. For example, consider an A350 AFDX network comprised of two redundant networks, with seven switches each [3], [26] as shown in Figure 1. Assume that all functionalities provided by these components are equally important. All ES can be considered as vertices, and Ethernet cables as edges. If *PRIMIA*, *FMCI*, and the Ethernet cable between switch 1 and switch 2 fail, the network is still functional, as there exist other redundant ES which can provide the same functionality to the network. On the other hand, if *SCI* fails, the entire network also fails.

B. Reliability Evaluation

The probability that determines whether a network is *functional* defines its reliability. Assuming the failures of all vertices / edges are independent, the reliability equation of the network can be written

$$R(r, \mathcal{G}) = \sum_{\mathbf{y}} \left\{ \beta(\mathbf{y}) \prod_{i=1}^{M+N} [(1 - r_i)(1 - y_i) + r_i y_i] \right\} \quad (1)$$

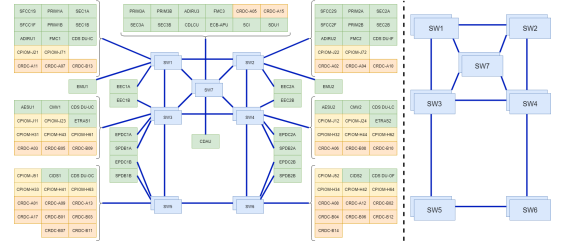


Fig. 1: A350 AFDX network (Left). Green boxes represent Line Replaceable Units (LRUs). Yellow boxes represent Core Processing Input/Output Module (CPIOMs). Orange boxes represent Common Remote Data Concentrator (CRDCs). Blue boxes represent Ethernet switches. Blue lines represent Ethernet cables. See [27] for abbreviation. A350 AFDX Network with only Ethernet switches (Right).

where $R(r, \mathcal{G}) \in [0, 1]$, and the binary vector $\mathbf{y} = [y_1, \dots, y_{M+N}]^T \in \{0, 1\}^{M+N}$ represents the state of vertices and edges where $y_i = 1$ means *active*. The binary Boolean function $\beta : \{0, 1\}^{M+N} \rightarrow \{0, 1\}$ represents the state of the network where

$$\beta(\mathbf{y}) = \begin{cases} 1 & \text{if the network is functional} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

To evaluate (1) with a linear time complexity, an approach that involves representing $\beta(\mathbf{y})$ as a Binary Decision Diagram (BDD) is used and described in the following section [28].

C. Binary Decision Diagram

A Binary Decision Diagram (BDD) is a Directed Acyclic Graph that consists of a hierarchy of nodes that have two branches, a low-branch and a high-branch, pointing to their lower-hierarchy nodes. The two lowest hierarchy nodes with no *outgoing branches* are called *terminal nodes*, labeled by either $\perp$ or $\top$. The highest hierarchy node with no *incoming branches* is called the *root node*. The BDD expands the solution of (2) in form of a binary tree. The details of constructing a BDD for the given network can be found in [29]–[31]. An example of a simple network and its BDD is given in Figure 2.

In order to distinguish between the network and the BDD, the words *vertices* and *edges* are used for the network, and the words *nodes* and *branches* are used for the BDD. The structure of the BDD is represented as a set of nodes $\mathcal{B} = \{b_1, \dots, b_B, b_\perp, b_\top\}$, where $B = |\mathcal{B}| - 2$ represents the size of the BDD and the ordering of $b_1, \dots, b_B$ is in the topological order, such that b_1 is the root node. The following definitions represent the relationship between network vertices/edges and BDD nodes/branches.

Definition II.1. Given a network $\mathcal{G}$ and its correspond BDD of a size B , the function $\text{ve} : \mathcal{B} \rightarrow \{V \cup E\}$ represents a mapping from the node to its correspond vertex or edge.

Definition II.2. Given a network $\mathcal{G}$ and its correspond BDD of a size B , the function $\text{lo}/\text{hi} : \mathbb{N} \rightarrow \mathbb{N}$ represents a mapping from the index of the node to the index of its lower-hierarchy node that corresponds to its low / high-branches.

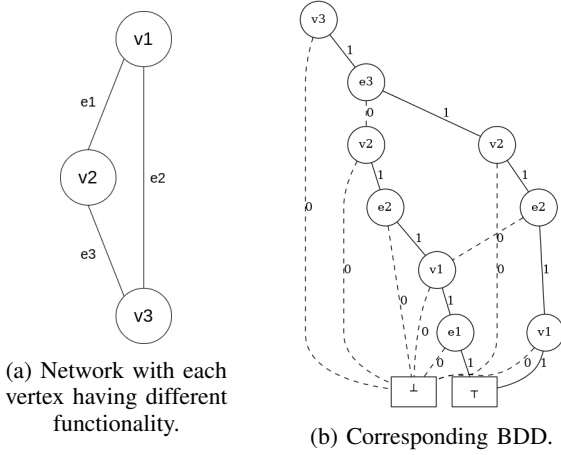


Fig. 2: An example network (2a), and its BDD (2b).

Defining II.1 and II.2, facilitates representing the Boolean function $\beta(\mathbf{y})$ as a recursive equation

$$\beta(\mathbf{y}|y_j) = (\neg y_j \wedge \beta(\mathbf{y}|y_{j0})) \vee (y_j \wedge \beta(\mathbf{y}|y_{j1})) \quad (3)$$

where $j = \text{ve}(\mathbf{b}_i)$, $j0 = \text{ve}(\mathbf{b}_{i0(i)})$, and $j1 = \text{ve}(\mathbf{b}_{i1(i)})$ for $i \in \{1, \dots, B\}$ and $j, j0, j1 \in \{1, \dots, M+N\}$. To determine whether the network is functional ($\beta(\mathbf{y}) = 0$, or $\beta(\mathbf{y}) = 1$) for a given vector $\mathbf{y}$, (3) may be simply evaluated from the root node b_1 down to the terminal node $b_\perp$ or $b_\top$. At a particular node i , if $y_j = 0$, the term $y_j \wedge \beta(\mathbf{y}|y_{j1})$ is always false and can be ignored. Similarly, for $y_j = 1$. The network is functioning if the node $b_\top$ is reached. To recursively evaluate the network reliability $R(r, \mathcal{G})$, substitution of (3) into (1) yields

$$R_i(r, \mathcal{G}) = (1 - r_j) \cdot R_{i0(i)}(r, \mathcal{G}) + r_j \cdot R_{i1(i)}(r, \mathcal{G}) \quad (4)$$

where $j = \text{ve}(\mathbf{b}_i)$ for $i \in \{1, \dots, B\}$ and $j \in \{1, \dots, M+N\}$. By using dynamic programming (DP), (4) can be computed, as described in Algorithm 1 with a complexity of $O(B)$.

Algorithm 1: DP for Reliability Evaluation

Input: network BDD, Reliability of vertices / edges

Output: Reliability of the Network

- 1 $R_\perp \leftarrow 0.0$, $R_\top \leftarrow 1.0$
 - 2 **For** $i = B, B-1, \dots, 1$
 - 3 $R_i \leftarrow (1 - r_{\text{ve}(i)}) \cdot R_{i0(i)} + r_{\text{ve}(i)} \cdot R_{i1(i)}$
 - 4 **Return** R_1
-

The following subsections introduce the mathematical programming techniques that will be used for formulating the optimization problems.

D. Exponential Cone Programming

Definition II.3. An exponential cone is defined as a convex subset of $\mathbb{R}^3$ of the form

$$K_{\text{exp}} = \left\{ (x_1, x_2, x_3) \mid x_2 \cdot \exp\left(\frac{x_3}{x_2}\right) \leq x_1 \right\} \cup \{(x_1, 0, x_3) \mid x_1 \geq 0, x_3 \leq 0\} \quad (5)$$

Definition II.4. An exponential cone program refers to a convex optimization problem that has at least one exponential conic constraint, i.e. $(x_1, x_2, x_3) \in K_{\text{exp}}$.

E. Geometric Programming

Definition II.5. A monomial function is a function, $f : \mathbb{R}_{++}^n \rightarrow \mathbb{R}$ defined by $f(\mathbf{x}) = c \prod_{i=1}^n x_i^{a_i}$ where $\mathbf{x} = [x_1, \dots, x_n]^T$, $c > 0$, and $a_i \in \mathbb{R}$.

Definition II.6. A posynomial function is a function, $f : \mathbb{R}_{++}^n \rightarrow \mathbb{R}$ defined by $f(\mathbf{x}) = \sum_i c_i \prod_{j=1}^n x_j^{a_{ij}}$ where $\mathbf{x} = [x_1, \dots, x_n]^T$, $c_i > 0$, and $a_{ij} \in \mathbb{R}$.

Definition II.7. A geometric program (GP) is an optimization problem of the form

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && f_0(\mathbf{x}) \\ & \text{subject to} && f_i(\mathbf{x}) \leq 1, \quad g_i(\mathbf{x}) = 1 \end{aligned} \quad (6)$$

where f_i are posynomial and g_i are monomial.

The problem (7) is generally not convex; however, it is possible to do a *log-transformation* to turn it into a convex program that guarantees a solution which can be found in polynomial time, i.e. a globally optimal solution [6].

F. Signomial Programming

Definition II.8. A signomial function is a function, $f : \mathbb{R}_{++}^n \rightarrow \mathbb{R}$ defined by $f(\mathbf{x}) = \sum_i c_i \prod_{j=1}^n x_j^{a_{ij}}$ where $\mathbf{x} = [x_1, \dots, x_n]^T$, $c_i \in \mathbb{R}$, and $a_{ij} \in \mathbb{R}$.

Definition II.9. A signomial program (SP) is an optimization problem of the form

$$\begin{aligned} & \underset{\mathbf{x}}{\text{minimize}} && f_0(\mathbf{x}) \\ & \text{subject to} && f_i(\mathbf{x}) \leq 1, \quad g_i(\mathbf{x}) = 1 \end{aligned} \quad (7)$$

where f_i are signomial and g_i are monomial.

In general, solving an SP involves approximating a signomial function as a monomial function, in which the SP becomes a GP, then iteratively solving the approximated problem, resulting in a locally optimal solution [6].

III. REDUNDANCY OPTIMIZATION

This section addresses an optimization problem for determining the minimum number of redundant components required to construct an initial network topology, under the constraint that its reliability be above a certain value.

A. Problem Description

Consider a system of I functionally-different subsystems. Each subsystem has J_i , $i \in \{1, \dots, I\}$ components. To mitigate the effects of a common cause failure, components in the same subsystem are assumed to be functionally-similar, meaning that they provide the same functionality, but differ in other features. The example are hardware from different manufacturers, or software compiled by different compilers or written by different teams. As illustrated in Figure 3, the connection of functionally-similar components can be assumed

to have a parallel network structure because the system can still provide the continuity of service if one or more, but not all, of these components fail. Otherwise, they are in a series network structure.

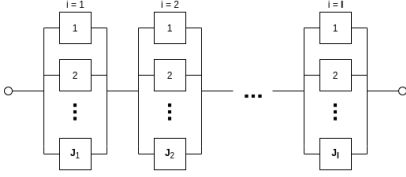


Fig. 3: A configuration of a series-parallel system.

Given the reliability of each component, r_{ij} where $j \in \{1, \dots, J_i\}$, the overall reliability of the series-parallel network may be calculated using a special case of an equation (1) [32].

B. Problem Formulation

Consider the same system in III-A, where each component is associated with a component weight, $w_{ij} > 0$ and a connection price $w_{ij,kl} > 0$ if the connection between the component ij and kl is required; otherwise, $w_{ij,kl} = 0$. Further, let $n \in \mathbb{N}^{I \times J}$ where $J = \max_i J_i$, be a decision matrix that determines the level of redundancy for each component in each subsystem (the number of components), and $\bar{r}$ be the desired reliability of the overall system. Then, the optimization problem for minimum redundancy can be cast as

$$\begin{aligned} & \text{minimize}_{n \in \mathbb{N}^{I \times J}} \sum_{i=1}^I \sum_{j=1}^{J_i} \left(w_{ij} n_{ij} + \frac{1}{2} \sum_{k=1}^I \sum_{l=1}^{J_k} w_{ij,kl} n_{ij} n_{kl} \right) \\ & \text{subject to } \prod_{i=1}^I \left[1 - \prod_{j=1}^{J_i} (1 - r_{ij})^{n_{ij}} \right] \geq \bar{r}. \end{aligned} \quad (8)$$

The objective function is to minimize the total cost of the system, where the factor $\frac{1}{2}$ accounts for the duplicate between the connection from the component ij to kl and from kl to ij . For the case where $ij = kl$, $w_{ij,kl}$ is set to zero, meaning that there is no connection to the same component. Although this problem is in the form of a mixed-integer nonlinear program, it can be reformulated into a form similar to a mixed-integer GP by exponentiating the objective function and performing appropriate change of variables, i.e., $p_{ij} = \log(1 - r_{ij})$, $n_{ij,kl} = n_{ij} n_{kl}$, $n_{(\cdot)} = \log[y_{(\cdot)}]$, and $\hat{y}_i \leq 1 - \prod_{j=1}^{J_i} y_{ij}^{p_{ij}}$.

$$\begin{aligned} & \text{minimize}_{\hat{y}_i, y_{(\cdot)}} \prod_{i=1}^I \prod_{j=1}^{J_i} \left(y_{ij}^{w_{ij}} \prod_{k=1}^I \prod_{l=1}^{J_k} y_{ij,kl}^{\frac{w_{ij,kl}}{2}} \right) \\ & \text{subject to } \bar{r} \left(\prod_{i=1}^I \hat{y}_i \right)^{-1} \leq 1, \quad \hat{y}_i + \prod_{j=1}^{J_i} y_{ij}^{p_{ij}} \leq 1 \\ & \log(y_{ij,kl}) = \log(y_{ij}) + \log(y_{kl}) \\ & \forall i, k \in \{1, \dots, I\}, \forall j, l \in \{1, \dots, J\} \end{aligned} \quad (9)$$

Because of the last constraint, this problem is not in the form of a GP; however, its integer relaxation can still be converted

into a convex optimization problem. By applying the log-transformation to the objective function and constraints, and letting $z_{(\cdot)} = \log[n_{(\cdot)}]$, $\hat{z}_i = \log(\hat{y}_i)$, this problem becomes

$$\begin{aligned} & \text{minimize}_{n_{(\cdot)}, z_{(\cdot)}, \hat{z}_i} \sum_{i=1}^I \sum_{j=1}^{J_i} \left(w_{ij} n_{ij} + \frac{1}{2} \sum_{k=1}^I \sum_{l=1}^{J_k} w_{ij,kl} n_{ij,kl} \right) \\ & \text{subject to } \log(\bar{r}) - \sum_{i=1}^I \hat{z}_i \leq 0 \\ & \log \left[\exp(\hat{z}_i) + \exp \left(\sum_{j=1}^{J_i} p_{ij} \cdot n_{ij} \right) \right] \leq 0 \\ & z_{ij} + z_{kl} - z_{ij,kl} = 0, \quad \exp[z_{(\cdot)}] \leq n_{(\cdot)} \\ & \forall i, k \in \{1, \dots, I\}, \forall j, l \in \{1, \dots, J\} \end{aligned} \quad (10)$$

The last inequality constraint is the convex relaxation of its equality constraint and is equivalent to $(n_{(\cdot)}, 1, z_{(\cdot)}) \in K_{exp}$. Despite, in theory, the convexified problem provides the lower bound in terms of the cost value to the original problem, solving (10) using a high performance solver, e.g. Mosek [8] together with a branch-and-bound method for integer variables, in practice, provides the solution closed to the case where the constraint is not relaxed as shown in Section V.

Once the level of redundancy of components is determined, these redundant components can be considered as vertices of the network, with fully-connected edges between subsystems.

C. Additional Constraints for Common Cause Failures

Since some of decision variables in the problem (10), i.e. n_{ij} , have the same physical interpretation as those in the problem (8), any additional mixed-integer constraints, whose relaxation is convex which depends on these decision variables can be included. For example, a boundary constraint for satisfying the allowable minimum and maximum number of components, or a modulo constraint for specifying the number of components to be even or odd [33]. Another useful constraint is one that handles common cause failure on redundant components.

The analysis result of common cause failures from [34] may be summarized as “for a given ratio of the failure rates between a single component and its redundant system, using one component may, on average, provide a longer continuity of service than the redundant one if the number of redundant components is lower than some value.” Mathematically, this is translated to

$$n_{ij} \begin{cases} = 1 & \text{if } n_{ij} < L_{ij} \\ \geq L_{ij} & \text{otherwise} \end{cases} \quad (11)$$

where L_{ij} is a lower bound for each component obtained from a method presented in [34]. By applying a Big- M method, the additional constraints are

$$-n_{ij} \leq -L_{ij} + M b_{ij} \quad (12)$$

$$n_{ij} \leq 1 + M(1 - b_{ij}) \quad (13)$$

where M is a sufficiently large positive number, and $b_{ij} \in \{0, 1\}$ is an extra decision variable indicating either (12) or

(13) is active. If $b_{ij} = 1$, the inequality constraint $n_{ij} \leq 1$ is active, which suggests that it is not beneficial to use a redundancy to improve the system reliability, and vice versa.

IV. TOPOLOGY OPTIMIZATION

In section III, it was shown that the network reliability can be improved using redundancy. However, it is advantageous in terms of both price and maintenance costs if the network is modified to contain fewer edges, whilst maintaining the desired reliability with the same number of vertices. Hence, this section addresses the problem of determining an optimal topology, in which the number of edges is minimized under a given reliability requirement.

A. Problem Formulation

Given a BDD of size B encoding an initial network topology graph, $\mathcal{G} = (V, E)$, and its connectivity, the problem can be formulated as an SP. Let $M = |V|$, $N = |E|$, $\bar{r}$ be the desired network reliability and $\mathbf{x} = [x_1, \dots, x_{M+N}]^T \in \{0, 1\}^{M+N}$ be a decision vector for determining whether a vertex $v_i \in V$, $i \in \{1, \dots, M\}$ or an edge $e_j \in E$, $j \in \{M+1, \dots, M+N\}$ should be kept ($x_i = 1$) or removed ($x_i = 0$). Additionally, denote the corresponding weights, $\mathbf{w} = [w_1, \dots, w_{M+N}]^T$, and reliability, $\mathbf{r} = [r_1, \dots, r_{M+N}]^T$, for each vertex and edge. The reliability equation (1) may then be expressed as

$$R(\mathbf{x}, r, \mathcal{G}) = \sum_{\mathbf{y}} \left\{ \beta(\mathbf{y}) \prod_{i=1}^{M+N} \left[(1 - r_i x_i)(1 - y_i) + r_i x_i y_i \right] \right\} \quad (14)$$

Furthermore, with the notions of BDD as described in II-C, equation (14) can be reformulated as a recursive equation

$$R_i(\mathbf{x}, r, \mathcal{G}) = (1 - r_j x_j) \cdot R_{lo(i)}(\mathbf{x}, r, \mathcal{G}) + r_j x_j \cdot R_{hi(i)}(\mathbf{x}, r, \mathcal{G}) \quad (15)$$

where $j = ve(b_i)$ for $i \in \{1, \dots, B\}$ and $j \in \{1, \dots, M+N\}$.

To evaluate (15), we may again leverage Algorithm 1 by multiplying r with x , which implies that the reliability of the removed vertex or edge is set to zero. The resulting equation is signomial in $\mathbf{x}$. Considering each R_i as an intermediate variable, the optimization problem may then be cast as an integer signomial program (SP).

$$\begin{aligned} \forall i \in \{1, \dots, B\}, j = ve(b_i), j \in \{1, \dots, M+N\} \\ \text{minimize } z = \mathbf{w}^T \mathbf{x} \\ \text{subject to } R_i \geq (1 - r_j x_j) R_{lo(i)} + r_j x_j R_{hi(i)} \\ R_1 \geq \bar{r}, \mathbf{x} \in \{0, 1\}^{M+N} \end{aligned} \quad (16)$$

To determine a globally optimal solution to this problem, $\mathbf{x}^*$, the branch-and-bound based approach can be used [35], [36]. However, this approach requires a valid upper bound, u , and lower bound, l , of the objective value for each iteration, which may be obtained via relaxations of the original problem (16).

B. Integer Relaxation for Upper Bound Computation

The upper bound of the objective value can be computed by relaxing the integer constraints, solving the following SP problem, and rounding the result to 0 or 1 [37].

$$\begin{aligned} \forall i \in \{1, \dots, B\}, j = ve(b_i), j \in \{1, \dots, M+N\} \\ \text{minimize } u = \mathbf{w}^T \mathbf{x} \\ \text{subject to } R_i \geq (1 - r_j x_j) R_{lo(i)} + r_j x_j R_{hi(i)} \\ R_1 \geq \bar{r}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{1} \\ \mathbf{x}_{\mu_k} = \nu_k \quad \forall k \in \{1, \dots, |\mu|\} \end{aligned} \quad (17)$$

where μ is a set of indices indicating which element of the decision vector $\mathbf{x}$ should be fixed during the branching process, and $\nu_k \in \{0, 1\}$.

C. Signomial Relaxation for Lower Bound Computation

To compute the lower bound, the convex relaxation of (16) must be approximated [37]. To do so, each signomial constraint is under-approximated by a polynomial function as follows. Each signomial constraint can be written as

$$\begin{aligned} f_i(\mathbf{x}, \mathbf{R}) = \frac{R_{lo(i)} - r_j x_j R_{lo(i)} + r_j x_j R_{hi(i)}}{R_i} \leq 1 \\ \implies \frac{R_{lo(i)} R_i^{-1} + r_j x_j R_{hi(i)} R_i^{-1}}{1 + r_j x_j R_{lo(i)} R_i^{-1}} \leq 1 \end{aligned} \quad (18)$$

Therefore, an under-approximation of $f_i(\mathbf{x}, \mathbf{R})$, may be found by computing an over-approximation of the denominator. The following proposition provides such an over-approximation in the form of a monomial function.

Proposition 1. *Given a function $f: [0, 1]^m \times (0, 1]^n \rightarrow \mathbb{R}_+$ of the form $f(\mathbf{x}) = 1 + c \prod_{i=1}^m x_i \prod_{j=m+1}^{m+n} x_j^{-1}$ where $c > 0$, the tightest monomial over-approximation, in the L_2 -norm sense, of $f(\mathbf{x})$ is given by*

$$\tilde{f}(\mathbf{x}) = (1 + c) \prod_{j=m+1}^{m+n} x_j^{-1}$$

Proof. See Appendix. ■

Hence, by applying Proposition 1 with $m = 2$, $n = 1$, and $c = r_j$, the relaxed problem becomes a GP of the form

$$\begin{aligned} \forall i \in \{1, \dots, B\}, j = ve(b_i), j \in \{1, \dots, M+N\} \\ \text{minimize } l = \mathbf{w}^T \mathbf{x} \\ \text{subject to } R_{lo(i)} + r_j x_j R_{hi(i)} \leq 1 + r_j \\ R_1 \geq \bar{r}, \mathbf{0} \leq \mathbf{x} \leq \mathbf{1} \\ \mathbf{x}_{\mu_k} = \nu_k \quad \forall k \in \{1, \dots, |\mu|\} \end{aligned} \quad (19)$$

which may then be transformed into a convex optimization problem and solved for the lower bound.

Algorithm 2: Mixed-Integer SP Algorithm for Reliability-Constrained Optimization Problem

Input: BDD representing the network, Reliability and weight of vertices / edges, Desired reliability

Output: $\mathbf{x}^*, z^*$

```

1  $z^* \leftarrow \mathbf{w}^T \mathbf{1} + 1$ 
2  $\mu \leftarrow \nu \leftarrow u \leftarrow l \leftarrow \text{None}$ 
3  $S \leftarrow U \leftarrow L \leftarrow \emptyset$ 
4 While  $z^* > \mathbf{w}^T \mathbf{1}$  or  $S \neq \emptyset$ 
5   If  $S \neq \emptyset$ 
6      $\mu, \nu \leftarrow S_1, S \leftarrow S \setminus S_1$ 
7      $u \leftarrow U_1, U \leftarrow U \setminus U_1$ 
8      $l \leftarrow L_1, L \leftarrow L \setminus L_1$ 
9   If  $l$  is None or  $l \leq \max_u U$ 
10     $U \leftarrow U \oplus \{u\}, L \leftarrow L \oplus \{l\}$ 
11     $\mathbf{x}_u, u \leftarrow \text{solve (17)}$ 
12     $\mathbf{x}_l, l \leftarrow \text{solve (19)}$ 
13    If  $\mathbf{x}_u \in \{0, 1\}^{M+N}$  and  $u < z^*$ 
14       $\mathbf{x}^* \leftarrow \mathbf{x}_u, z^* \leftarrow u$ 
15    Else if  $0 \leq \mathbf{x}_u \leq 1$  or  $0 \leq \mathbf{x}_l \leq 1$ 
16      If  $l \leq \max_u U$ 
17         $s \leftarrow \arg \max_{\mathbf{x}} \{\mathbf{x} \mid 0 < \mathbf{x} < 1\}$ 
18         $\mu \leftarrow \mu \oplus s$ 
19         $S \leftarrow S \oplus \{(\mu, \nu \oplus 0), (\mu, \nu \oplus 1)\}$ 
20         $U \leftarrow U \oplus \{u, u\}, L \leftarrow L \oplus \{l, l\}$ 
21 Return  $\mathbf{x}^*, z^*$ 

```

D. Algorithm for Reliability-Constrained Optimization

The Algorithm 2 is based on a standard branch-and-bound method [38] with the upper bound and lower bound being a solution from integer-relaxed problem (17) and a convex relaxation obtained from the transformation of (19), respectively (which becomes convex upon making a change of variables). The heuristic used in this algorithm is to choose the element of $\mathbf{x}$ closest to one as a branching variable, since such variables are more likely to be a vertex / an edge that cannot be removed from the network. Hence, the branch where this variable is fixed to zero is likely to be infeasible and immediately fathomed.

Theorem 1. *Given the problem formulation of reliability-constrained optimization for networked systems as (16), the branch-and-bound based algorithm 2 provides a globally optimal solution through relaxation problems (17) and (19).*

Proof. See Appendix. ■

V. NUMERICAL EXAMPLES

This section provides applications of the proposed methods on a small network of mobile robots with a hardware implementation, and on a larger network of A350 IMA architecture.

A. Small network of mobile robots

Consider two types (pink, and green) of mobile robots collaborating as a network of two components, as shown in Figure 4a. Each robot has a different functionality. Assume

that the first robot has a reliability of 0.8, whereas the second has a reliability of 0.7. Additionally, let the reliability of the wireless communication between a pair of robots be 0.9. Suppose further that the desired reliability of the network is 0.8. It is obvious that this requirement is not satisfied by the two-robot configuration. Therefore, the method proposed in section III is used to improve the network reliability. To this end, the method proposed in section IV is used to optimize the topology.

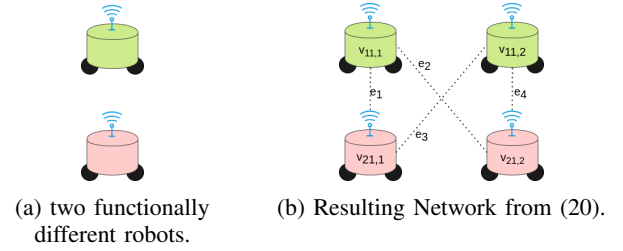


Fig. 4: Network of Mobile Robots.

1) *Redundancy Optimization:* Following the description in section III, this problem has two functionally-different subsystems, each with a single component. Taking all the weights to be unity, the optimization problem (10) can be expressed as

$$\begin{aligned}
 & \underset{\substack{\hat{z}_1, \hat{z}_2, \\ n_{11}, n_{21}, n_{11,21}, \\ z_{11}, z_{21}, z_{11,21}}}{\text{minimize}} && n_{11} + n_{21} + n_{11,21} \\
 & \text{subject to} && \log(0.8) - \hat{z}_1 - \hat{z}_2 \leq 0 \\
 & && \log[\exp(\hat{z}_1) + \exp(\log(0.2)n_{11})] \leq 0 \\
 & && \log[\exp(\hat{z}_2) + \exp(\log(0.3)n_{21})] \leq 0 \\
 & && z_{11} + z_{21} - z_{11,21} = 0 \\
 & && \exp(z_{11}) \leq n_{11}, \exp(z_{21}) \leq n_{21} \\
 & && \exp(z_{11,21}) \leq n_{11,21}
 \end{aligned} \tag{20}$$

where n_{11} and n_{12} indicate the number of robots for each type. The results of (20) are provided in Table I and Figure 4b. Note that there are only fully-connected communications between functionally-different subsystems. Following the correspond BDD in Figure 5, the reliability of the network is 0.8559, which is higher than the desired reliability.

TABLE I: Optimization Result of (20)

$\hat{z}_1$	$\hat{z}_2$	n_{11}	n_{21}	$n_{11,21}$	z_{11}	z_{21}	$z_{11,21}$
-0.0408	-0.0943	2	2	4	0.693	0.693	1.386

2) *Topology Optimization:* To further optimize the network in Figure 4b, its corresponding BDD (Figure 5) is used for formulating a problem. With all weights set to 1, the

optimization problem (16) can be written as

$$\begin{aligned}
& \underset{\mathbf{x} \in \{0,1\}^4, \mathbf{R} \in \mathbb{R}^{16}}{\text{minimize}} && x_1 + x_2 + x_3 + x_4 \\
& \text{subject to} && R_1 \geq 0.8 \\
& && R_1 \geq 0.2R_4 + 0.8R_2, \quad R_2 \geq (1 - 0.9x_1)R_3 + 0.9x_1R_6 \\
& && R_3 \geq 0.3R_8 + 0.7R_5, \quad R_4 \geq 0.3R_9 + 0.7R_7 \\
& && R_5 \geq (1 - 0.9x_2)R_7 + 0.9x_2R_{11}, \quad R_6 \geq 0.3R_8 + 0.7 \\
& && R_7 \geq 0.3R_{13} + 0.7R_{10}, \quad R_8 \geq (1 - 0.9x_2)R_9 + 0.9x_2R_{16} \\
& && R_9 \geq 0.7R_{12}, \quad R_{10} \geq (1 - 0.9x_3)R_{12} + 0.9x_3R_{15} \\
& && R_{11} \geq 0.3R_{13} + 0.7, \quad R_{12} \geq 0.8R_{14} \\
& && R_{13} \geq 0.9x_3R_{15}, \quad R_{14} \geq 0.9x_4 \\
& && R_{15} \geq 0.8, \quad R_{16} \geq 0.7
\end{aligned} \tag{21}$$

The optimization result of (21) is shown in Table II and Figure 6. Note that because the number of vertices has been determined by (20), there is no need to consider vertices as decision variables. The result shows that $e1$ can be removed, while the reliability of this final configuration (0.8062) still remains higher than the desired reliability.

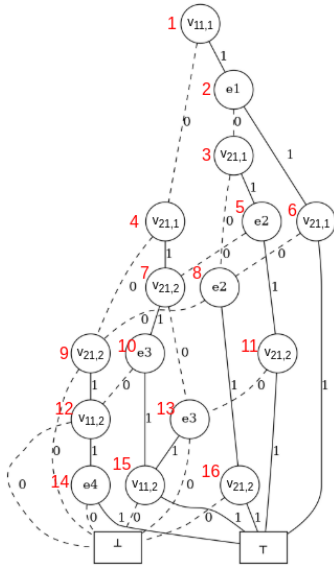


Fig. 5: BDD corresponds to the network in Figure 4b.

TABLE II: Optimization Result of (21).

variable	x_1	x_2	x_3	x_4	R_1
value	0	1	1	1	0.8062
variable	R_2	R_3	R_4	R_5	R_6
value	0.8351	0.8351	0.6905	0.9014	0.9041
variable	R_7	R_8	R_9	R_{10}	R_{11}
value	0.7704	0.6804	0.504	0.792	0.916
variable	R_{12}	R_{13}	R_{14}	R_{15}	R_{16}
value	0.72	0.72	0.9	0.8	0.7

3) *Hardware Implementation:* An experiment has been conducted on a swarm robotics platform [39] to illustrate the obtained result. The objective of this experiment is to simulate the failure of the networked system over time. This means that

when failure occurs on one or more components such that the entire system fails, data will be recorded, and the system will be repaired, such that it may continue operating. Then, from the recorded data, the overall system reliability can be shown to converge to the reliability obtained from (21), i.e., 0.8062.

In this setup, the robot network is moving along a rectangular trajectory, where each corner is marked by G1-G4 as shown in Figure 6. To evaluate the reliability, at each time step the component-level reliability are evaluated to determine whether the vertices (robot) or edges (communication between robots) are available. If not, those vertices or edges are removed from the network for a brief fixed interval, immobilizing the disconnected robot. After the interval has passed, the faulty vertices and edges are recovered. Then, the overall reliability is re-evaluated for the next time step. This experiment procedure is summarized in Algorithm 3.

Algorithm 3: Procedure for Estimating System Reliability in Example V-A

Input: The number of time steps T , The number of robots M , The number of communication pairs N , Reliability of each robot r_{robot} , and each communication pair r_{comm}

Output: Estimated System Reliability R

```

1  $R \leftarrow 0, n_f \leftarrow 0$ 
2 For  $t = 1 : T$ 
3    $\mathbf{a} \leftarrow \mathbf{0}_{M \times 1}, \mathbf{b} \leftarrow \mathbf{0}_{N \times 1}$ 
4   For  $i = 1 : M + N$ 
5      $x \leftarrow \mathcal{U}(0, 1)$ 
6     If  $i \leq M$  and  $r_{\text{robot}} < x$ 
7        $\mathbf{a}_i \leftarrow 1$ 
8        $\mathbf{b}_{\text{communication pair connected to robot } i} \leftarrow 1$ 
9     If  $i > M$  and  $r_{\text{comm}} < x$  and  $\mathbf{b}_{i-M} = 0$ 
10       $\mathbf{b}_{i-M} \leftarrow 1$ 
11   Check the connection among robots
12   If one or more robots is disconnected
13      $n_f \leftarrow n_f + 1$ 
14    $R \leftarrow 1 - \frac{n_f}{t}$ 
15   Recover all of faulty components
16 Return  $R$ 

```

where n_f is the number of times that that system has been failed, $\mathbf{a}$ and $\mathbf{b}$ are Boolean vectors indicating which components are failed at a given time step. Line 4 – 10 are used to randomly inject faults to the system based on the value of its components' reliability. Line 11 check whether all robots are still able to communicate despite some injected faults. One method to do this is to construct the graph from the remaining components and compute eigenvalues of the graph Laplacian [40]. The remaining lines are to keep track of the number of failures and to calculate the reliability of each time step. Then the state of the system is recovered for the next step.

In this example, the experiment is repeated for 300 time steps. Since the desired reliability is 0.8, it is expected that out of these 300 evaluations, approximately 60 of them should end up in the system failures. The number of failures is counted and shown in 6 as well as the plot of the estimated system

reliability in Figure 7, which shows that the desired reliability of 0.8 can be achieved.

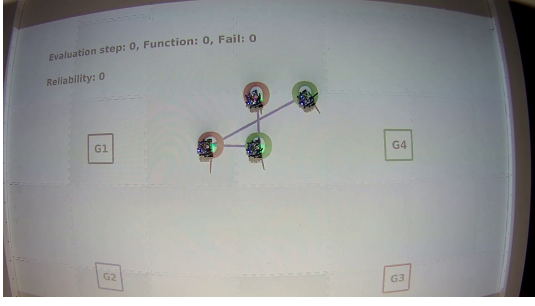


Fig. 6: Experiment illustrating the assurance of the reliability. Link to the video of the experiment: dribbble.com/s/ti01o4z7v2nt62r/RobotExp.mpeg?dl=0

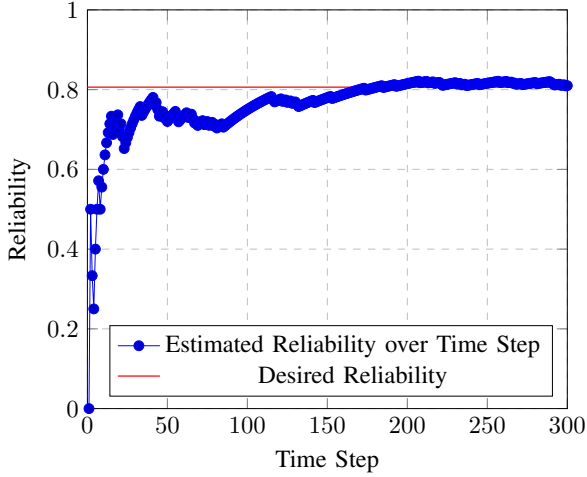


Fig. 7: Evaluation of system reliability at each time step.

B. A350 IMA Architecture

A large network inspired by the A350 Integrated Modular Avionics, shown in Figure 1 is now considered. However, instead of assuming the reliability of each ES is known, as in the previous example, the procedure adopted from ARP4761 [41] is applied to determine the acceptable component-level reliability for the original IMA architecture. Then, it is shown that from starting with this component-level reliability, the optimal architecture obtained from the proposed methodology is the same as the original one.

1) *Determination of component-level reliability:* Given a hazardous system severity (failure rate $\leq 10^{-7}$ per hour), the fault tree analysis (FTA) [42] can be used to determine each ES reliability, as shown in Figure 8. The level of redundancy for each ES is shown in Table III. At the bottom of the fault tree, the failure may come from either the ES or the cables connecting these ES. The reliability of an individual ES can be computed using equations for series-parallel system structure [43], while the reliability of the cable can be computed by constructing a BDD and solving the higher-order polynomial equation obtained from recursive substitution of terms on the

right-hand side of (4) [44]. It should be noted that although the switches are connected in a network structure, a single switch failure blocks the communication of other ES that connect to the faulty switch. Said failure leads to the failure of the entire network. Thus, this behavior of failure is equivalent to the series system. Assuming the average flight hour of an A350 is 7 hours, the desired reliability to satisfy the severity requirement is $e^{-7 \times 10^{-7}} \approx 1 - 7 \times 10^{-7}$. The computational result of component-level reliability is shown in Table III.

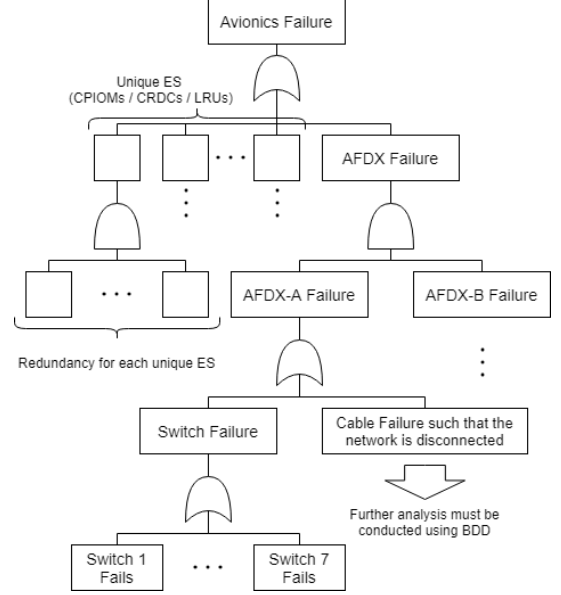


Fig. 8: FTA inspired by Airbus A350 AFDX network.

TABLE III: Component-Level Redundancy and Reliability.

End-Systems	Redundancy based on Fig. 1	Reliability
CRDC-A15, CDAU, SDU DU-{IC,UC,OC,IF,LC,OF} CDLCU, ECB-APU, SCI	1	0.999999985
Other CRDCs, AESU CPIOM-{J1, J5, J7}, CMV CMV, ETRAS, EMU	2	0.999877961
ADIRU, FMC	3	0.997539632
CPIOM-{H3, H4, H6, J2} SFCC, EEC, EPDC, SPDB	4	0.988952855
PRIM, SEC	6	0.950397903
Switches (per ADFX)	7	0.999991282
Ethernet Cables	N/A	0.999994915

2) *Redundancy Optimization:* The problem has been formulated as in the form of (10) with all weights set to 1. Connections are only allowed between CPIOMs/CDRCs/LRUs and switches. There are 233 variables (excluding slacks), 93 of which are integer. The problem is solved in 40.08 milliseconds on an Intel i7-9700F computer, with 16 GB of memory running Ubuntu 20.04. The result is provided in Table IV, whose contents are similar to those in Table III.

3) *Topology Optimization:* The problem has been formulated as in the form of (16), with a fully-connected network

TABLE IV: Redundancy Optimization Result.

End-Systems	n_{ij}	End-Systems	n_{ij}
CRDC-A15, CDAU, SDU DU-{IC,UC,OC,IF,LC,OF} CDLCU, ECB-APU, SCI	1	Other CRDCs, AESU CPIOM-{J1, J5, J7}, CMV CMV, ETRAS, EMU	2
ADIRU, FMC	3	CPIOM-{H3, H4, H6, J2} SFCC, EEC, EPDC, SPDB	4
PRIM, SEC	6	Switches (per ADFX)	7

between CPIOMs/CDRCs/LRUs and switches, and among switches. There are 1526 binary variables to determine the connections that can be removed without affecting network reliability. To avoid non-unique solutions, each weight is set to the scaled distance between ES and switches, according to Figure 1. An additional equality constraint forcing the homogeneity between two redundant AFDX network is imposed.

The problem is solved on the same machine as the redundancy optimization problem. It takes 549.35 seconds to obtain the results shown in Table V, and Figure 9. The number 1 in Table V indicates that there is a connection between two ES, and 0 otherwise. In Figure 9, for the ease of visibility, only the connections among Ethernet switches are shown.

VI. CONCLUSION

A two-step framework for designing a network architecture under reliability constraints is proposed. Both steps are based on a mixed-integer program, in which the tools solving for their globally optimal solution are also presented. Numerical calculations show that the pertinence of the proposed method for optimization of real-world network applications on an A350 IMA. Furthermore, the assurance of the desired reliability is illustrated by a hardware implementation on a swarm robotics environment, further demonstrating the effectiveness of the proposed formulation.

For the future works, the actual component reliability data should be gathered. A more sophisticated representation of reliability function should be considered. Potential sources of a common cause failure must be examined.

ACKNOWLEDGMENT

The authors would like to thank Dr. Pascal Traverse at Airbus for his useful discussion about avionics architecture.

APPENDIX

Proof of Proposition 1. Consider a monomial function of the general form $\tilde{f}(\mathbf{x}, \alpha) = \alpha_0 \prod_{k=1}^{m+n} x_k^{\alpha_k}$. Finding the tightest monomial over-approximation of f is equivalent to minimizing a function, $e(\mathbf{x}, \alpha) = \tilde{f}(\mathbf{x}, \alpha) - f(\mathbf{x})$, i.e.,

TABLE V: Network Topology Result. The value 1 indicates the connection between two components, 0 otherwise.

Switches End-Systems	1	2	3	4	5	6	7
PRIM-{3,4}, SEC-{3,4} CRDC-{A2,A4,A10}, DU-IC CPIOM-{J22,J72}, ADIRU2, FMC2 SFCC-{3,4}, EMU2, EEC3	1	0	0	0	0	0	0
PRIM-{1,2}, SEC-{1,2} CRDC-{A7,A11,B13}, DU-IF CPIOM-{J21,J71}, ADIRU1, FMC1 SFCC-{1,2}, EMU1, EEC1	0	1	0	0	0	0	0
AESU1, CMV1, DU-UC CPIOM-{J11,J23,H31,H43,H61} CRDC-{A3,B5,B9}, ETRAS1 EEC2, EPDC1, SPDB1	0	0	1	0	0	0	0
AESU2, CMV2, DU-LC CPIOM-{J12,J24,H32,H44,H62} CRDC-{A6,B8,B10}, ETRAS2 EEC4, EPDC3, SPDB4	0	0	0	1	0	0	0
CIDS1, DU-OC, EPDC2, SPDB2 CPIOM-{J51,H33,H41,H63} CRDC-A-{1,9,13,17}, B-{1,3,7,11}	0	0	0	0	1	0	0
CIDS2, DU-OF, EPDC4, SPDB4 CPIOM-{J52,H34,H42,H64} CRDC-A-{8,12}, B-{2,4,6,12,14}	0	0	0	0	0	1	0
PRIM-{5,6}, SEC-{5,6}, ADIRU3 CDLCU, FMC3, ECB-APU, SCI CRDC-{A05,A15}, SDU, CDAU	0	0	0	0	0	0	1
Switch 1	-	1	1	0	0	0	1
Switch 2	1	-	0	1	0	0	1
Switch 3	1	0	-	1	1	0	1
Switch 4	0	1	1	-	0	1	1
Switch 5	0	0	1	0	-	1	0
Switch 6	0	0	0	1	1	-	0
Switch 7	1	1	1	1	0	0	-

$$\begin{aligned}
& \underset{\alpha \in \mathbb{R}^{m+n+1}}{\text{minimize}} \quad \frac{1}{2} \left\| \alpha_0 \prod_{k=1}^{m+n} x_k^{\alpha_k} - 1 - c \prod_{i=1}^m x_i \prod_{j=m+1}^{m+n} x_j^{-1} \right\|_2^2 \\
& \text{subject to} \quad \alpha_0 \prod_{k=1}^{m+n} x_k^{\alpha_k} \geq 1 + c \prod_{i=1}^m x_i \prod_{j=m+1}^{m+n} x_j^{-1} \quad (22) \\
& \quad \forall x_i \in [0, 1], \quad i \in \{1, \dots, m\} \\
& \quad \forall x_j \in (0, 1], \quad j \in \{m+1, \dots, m+n\}
\end{aligned}$$

To solve this problem, the feasible domain of each α_i must be analyzed. First, $\alpha_0 > 0$ by the definition of monomial function. Next, for the constraint to be valid at every $x_i = 0$, $\tilde{f}(\mathbf{x})$ must be greater or equal to 1, which implies $\alpha_i^* = 0$, and $\alpha_j \leq 0$. If every x is set to 1 except one single x_l where $l \in \{m+1, \dots, m+n\}$, then $-1 < \alpha_l \leq 0$ violates the constraint, and $\alpha_l \geq -1$ increases the value of the objective function as α_l increases. Therefore, $\alpha_l^* = -1$ is optimal in this case. Applying this for every $l \in \{m+1, \dots, m+n\}$ implies that every $\alpha_j = -1$. Consequently, when every x is set to 1, any $\alpha_0 < 1+c$ violates the constraint, and $\alpha_0 \geq 1+c$

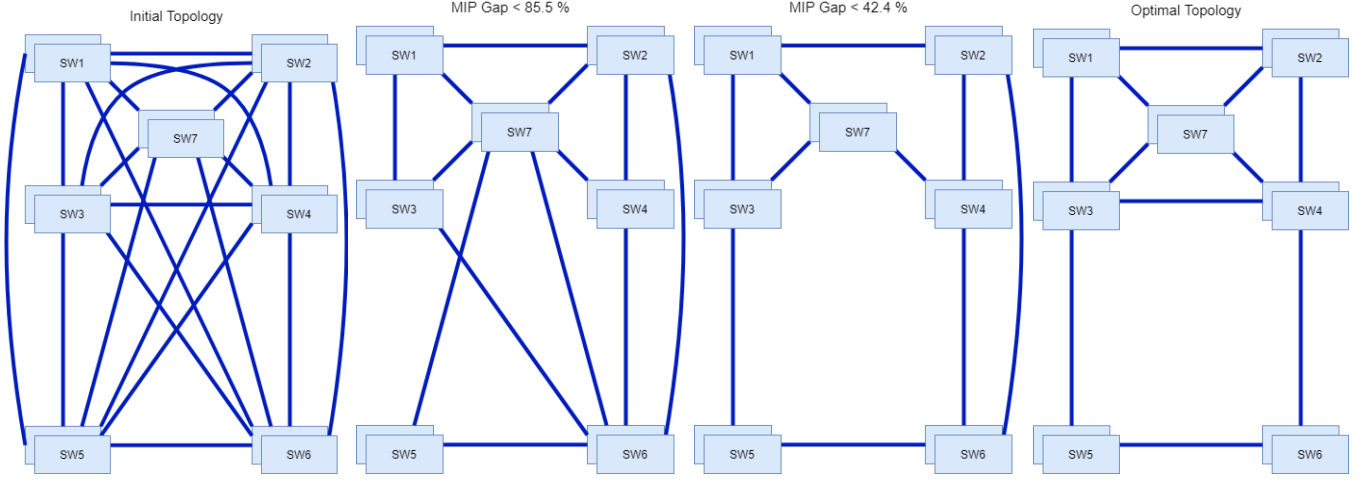


Fig. 9: Sequence of Integer-Feasible Solutions for AFDX Network Topology Optimization. The optimal topology is the same as the one shown in Figure 1.

increases the value of the objective function as α_0 increases. Therefore, $\alpha_0^* = 1 + c$ is optimal.

To check that $\alpha_0^* = 1 + c$, $\alpha_i^* = 0$, and $\alpha_j^* = -1$ are indeed optimal, the KKT necessary conditions are used [45].

The equation for Lagrangian can be written as

$$L = \frac{1}{2} \left\| \alpha_0 \prod_{k=1}^{m+n} x_k^{\alpha_k} - 1 - c \prod_{i=1}^m x_i \prod_{j=m+1}^{m+n} x_j^{-1} \right\|_2^2 + \lambda \left(1 + c \prod_{i=1}^m x_i \prod_{j=m+1}^{m+n} x_j^{-1} - \alpha_0 \prod_{k=1}^{m+n} x_k^{\alpha_k} \right) \quad (23)$$

It can be seen that it is not possible for the constraint to be active for all value of $\mathbf{x}$ in the domain; therefore, the Lagrange multipliers must be zero ($\lambda = 0$) to satisfy the complementary slackness condition. The partial derivatives of Lagrangian w.r.t. α_0 , α_i , and α_j are

$$\left. \frac{\partial L(\mathbf{x}, \alpha)}{\partial \alpha_0} \right|_{\alpha^*} = e(\mathbf{x}, \alpha^*) \cdot \prod_{k=m+1}^{m+n} x_k^{-1} \quad (24)$$

$$\left. \frac{\partial L(\mathbf{x}, \alpha)}{\partial \alpha_{i/j}} \right|_{\alpha^*} = e(\mathbf{x}, \alpha^*) \cdot (1 + c) \prod_{k=m+1}^{m+n} x_k^{-1} \cdot \log(x_{i/j}) \quad (25)$$

where

$$e(\mathbf{x}, \alpha^*) = (1 + c) \prod_{k=m+1}^{m+n} x_k^{-1} - 1 - c \prod_{i=1}^m x_i \prod_{j=m+1}^{m+n} x_j^{-1}$$

Since $e(\mathbf{x}, \alpha^*) = 0$ at $\mathbf{x} = \mathbf{1}$, KKT necessary conditions are satisfied. Note that, based on the previous analysis, if α is picked to be optimal at other point, it will violate the constraint at $\mathbf{x} = \mathbf{1}$. For the second-order sufficient condition, it can be seen that the only non-zero second-order partial derivative of Lagrangian evaluated at α^* and $\mathbf{x} = \mathbf{1}$ is

$$\left. \frac{\partial^2 L(\mathbf{x}, \alpha)}{\partial \alpha_0^2} \right|_{\alpha^*, \mathbf{x}=\mathbf{1}} = \prod_{k=m+1}^{m+n} x_k^{-2} \Big|_{\alpha^*, \mathbf{x}=\mathbf{1}} = 1 \quad (26)$$

Hence, the Hessian is positive semi-definite, which implies optimality. ■

Proof of Theorem 1. Because the branch-and-bound method depends on the convergence of upper and lower bounds towards each other as the size of the optimization domain shrinks, its global optimality can be achieved if both bounds obtained from relaxation problems (17) and (19) are valid. The validity of the upper bound can be derived by leveraging that fact that the problem (17) is an integer-relaxed SP and its solution is at least locally optimal. Thus, if this solution is not globally optimal itself, there always exists another solution that cannot be worse than this one. Hence, it is a valid upper bound.

For a lower bound, it is necessary to show that nonconvex constraints are under-approximated and the relaxed problem (19) can be solved to a globally optimal solution. From (18), since every element of $\mathbf{x}$, and $\mathbf{R}$ are in the domain of $[0, 1]$.

$$\begin{aligned} f_i(\mathbf{x}, \mathbf{R}) &= \frac{R_{lo(i)} R_i^{-1} + p_j x_j R_{hi(i)} R_i^{-1}}{1 + p_j x_j R_{lo(i)} R_i^{-1}} \\ &= \frac{R_{lo(i)} + p_j x_j R_{hi(i)}}{R_i + p_j x_j R_{lo(i)}} \geq \frac{R_{lo(i)} + p_j x_j R_{hi(i)}}{1 + p_j} \end{aligned} \quad (27)$$

Therefore, the constraints used in (19) under-approximate the original ones in (16). Furthermore, these functions are posynomial, which implies that the problem (19) can be solved to a globally optimal solution. ■

REFERENCES

- [1] J.-B. Itier, "A380 integrated modular avionics," in *Proceedings of the ARTIST2 meeting on integrated modular avionics*, vol. 1, no. 2, 2007, pp. 72–75.
- [2] B. Annighöfer and E. Kleemann, "Large-scale model-based avionics architecture optimization methods and case study," *IEEE Transactions on Aerospace and Electronic Systems*, 2019.
- [3] I. Moir, A. Seabridge, and M. Jukes, *Civil avionics systems*. John Wiley & Sons, 2013.
- [4] R. P. Collinson, *Introduction to avionics systems*. Springer Science & Business Media, 2013.

- [5] B. Bollobás, *Modern graph theory*. Springer Science & Business Media, 2013, vol. 184.
- [6] S. Boyd, S.-J. Kim, L. Vandenbergh, and A. Hassibi, "A tutorial on geometric programming," *Optimization and engineering*, vol. 8, no. 1, p. 67, 2007.
- [7] E. Burnell, N. B. Damen, and W. Hoburg, "GPkit: A human-centered approach to convex optimization in engineering design," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, 2020.
- [8] "MOSEK software (2020)," URL <http://www.mosek.com>, 2020.
- [9] J. H. Saleh and J.-F. Castet, *Spacecraft reliability and multi-state failures: a statistical approach*. John Wiley & Sons, 2011.
- [10] B. Annighoefer, C. Reif, and F. Thieleck, "Network topology optimization for distributed integrated modular avionics," in *2014 IEEE/AIAA 33rd Digital Avionics Systems Conference (DASC)*. IEEE, 2014, pp. 4A1–1.
- [11] R. Soltani, "Reliability optimization of binary state non-repairable systems: A state of the art survey," *International Journal of Industrial Engineering Computations*, vol. 5, no. 3, pp. 339–364, 2014.
- [12] A. Peiravi, M. Karbasian, M. A. Ardakan, and D. W. Coit, "Reliability optimization of series-parallel systems with k-mixed redundancy strategy," *Reliability Engineering & System Safety*, vol. 183, pp. 17–28, 2019.
- [13] S. Gupta, K. Deep, and A. Assad, "Reliability–redundancy allocation using random walk gray wolf optimizer," in *Soft Computing for Problem Solving*. Springer, 2020, pp. 941–959.
- [14] K. B. Misra and J. Sharma, "A new geometric programming formulation for a reliability problem," *International Journal of Control*, vol. 18, no. 3, pp. 497–503, 1973. [Online]. Available: <https://doi.org/10.1080/00207177308932529>
- [15] K. Govil, "Geometric programming method for optimal reliability allocation for a series system subject to cost constraint," *Microelectronics Reliability*, vol. 23, no. 5, pp. 783 – 784, 1983. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/0026271483909940>
- [16] G. Mahapatra and T. Roy, "Optimal redundancy allocation in series-parallel system using generalized fuzzy number," *Tamsui Oxford Journal of Information and Mathematical Sciences*, 2011.
- [17] D. P. Kroese, K.-P. Hui, and S. Nariyai, "Network reliability optimization via the cross-entropy method," *IEEE Transactions on Reliability*, vol. 56, no. 2, pp. 275–287, 2007.
- [18] W.-C. Yeh, Y.-C. Lin, Y. Y. Chung, and M. Chih, "A particle swarm optimization approach based on Monte Carlo simulation for solving the complex network reliability problem," *IEEE Transactions on Reliability*, vol. 59, no. 1, pp. 212–221, 2010.
- [19] J. Zhao, S. Si, and Z. Cai, "A multi-objective reliability optimization for reconfigurable systems considering components degradation," *Reliability Engineering & System Safety*, vol. 183, pp. 104–115, 2019.
- [20] B. Elshqeirat, S. Soh, S. Rai, and M. Lazarescu, "Topology design with minimal cost subject to network reliability constraint," *IEEE Transactions on Reliability*, vol. 64, no. 1, pp. 118–131, 2014.
- [21] M. Nishino, T. Inoue, N. Yasuda, S.-I. Minato, and M. Nagata, "Optimizing network reliability via best-first search over decision diagrams," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 1817–1825.
- [22] B. Ozturk and A. Saab, "Optimal aircraft design decisions under uncertainty via robust signomial programming," in *AIAA Aviation 2019 Forum*, 2019, p. 3351.
- [23] A. P. Dowdle, D. K. Hall, and J. H. Lang, "Electric propulsion architecture assessment via signomial programming," in *2018 AIAA/IEEE Electric Aircraft Technologies Symposium (EATS)*. IEEE, 2018.
- [24] M. J. Bellotti and M. Vucic, "Sparse FIR filter design based on signomial programming," *Elektronika ir Elektrotehnika*, 2020.
- [25] S. Fu, H. Wen, and B. Wu, "Power-fractionizing mechanism: Achieving joint user scheduling and power allocation via geometric programming," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 3, pp. 2025–2034, 2017.
- [26] Airbus, "A350 technical training manual maintenance course - T1+T2 - RR trent XWB integrated modular avionics and avionics data communication network," Oct 2013.
- [27] Airbus, "Lexinet - Airbus Reference Language Abbreviation," URL http://www.smartcockpit.com/docs/airbus_abbreviations_4.pdf, 2017.
- [28] G. Hardy, C. Lucet, and N. Limnios, "K-terminal network reliability measures with binary decision diagrams," *IEEE Transactions on Reliability*, vol. 56, no. 3, pp. 506–515, 2007.
- [29] M. O. Ball, "Computational complexity of network reliability analysis: An overview," *IEEE Transactions on Reliability*, 1986.
- [30] D. E. Knuth, *The art of computer programming, volume 4A: combinatorial algorithms, part 1*. Pearson Education India, 2011.
- [31] S.-Y. Kuo, F.-M. Yeh, and H.-Y. Lin, "Efficient and exact reliability evaluation for networks with imperfect vertices," *IEEE Transactions on Reliability*, vol. 56, no. 2, pp. 288–300, 2007.
- [32] S. Chauhan and S. Malik, "Reliability evaluation of series-parallel and parallel-series systems for arbitrary values of the parameters," *International Journal of Statistics and Reliability Engineering*, vol. 3, no. 1, pp. 10–19, 2016.
- [33] F. S. Hillier, *Introduction to operations research*. Tata McGraw-Hill Education, 2012.
- [34] V. Hoepfer, J. H. Saleh, and K. B. Marais, "On the value of redundancy subject to common-cause failures: toward the resolution of an on-going debate," *Reliability Engineering & System Safety*, 2009.
- [35] I. P. Androulakis, *MINLP: branch and bound global optimization algorithm*. Boston, MA: Springer US, 2009, pp. 2132–2138.
- [36] D. Li, X. Sun, and K. McKinnon, "An exact solution method for reliability optimization in complex systems," *Annals of Operations Research*, vol. 133, no. 1–4, pp. 129–148, 2005.
- [37] S. Boyd and J. Mattingley, "Branch and bound methods," *Notes for EE364b, Stanford University*, pp. 2006–07, 2007.
- [38] E. L. Lawler and D. E. Wood, "Branch-and-bound methods: A survey," *Operations research*, vol. 14, no. 4, pp. 699–719, 1966.
- [39] S. Wilson, P. Glotfelter, L. Wang, S. Mayya, G. Notomista, M. Mote, and M. Egerstedt, "The robotarium: Globally impactful opportunities, challenges, and lessons learned in remote-access, distributed control of multirobot systems," *IEEE Control Systems Magazine*, 2020.
- [40] M. Mesbahi and M. Egerstedt, *Graph theoretic methods in multiagent networks*. Princeton University Press, 2010.
- [41] SAE International Committee, "ARP4761 guidelines and methods for conducting the safety assessment process on civil airborne system and equipment," *Society of Automotive Engineers: Warrendale, WA, USA*, pp. 22–28, 1996.
- [42] C. A. Ericson and C. Li, "Fault tree analysis," in *System Safety Conference, Orlando, Florida*, vol. 1, 1999, pp. 1–9.
- [43] A. Yalaoui, E. Châtelet, and C. Chu, "Series-parallel systems design: Reliability allocation," *Journal of decision systems*, 2005.
- [44] Y. Ikeda, R. Kawahara, and H. Saito, "Generating a network reliability formula by using binary decision diagrams," *IEICE Communications Express*, vol. 4, no. 9, pp. 299–303, 2015.
- [45] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.



Thanakorn Khamvilai received his B.S. degree in Aerospace Engineering from Kasetsart University, Bangkok, Thailand in 2016. He is currently a visiting student at King Abdullah University of Science and Technology and a Ph.D. candidate the Georgia Institute of Technology. His interests include flight dynamics, avionics, control systems, and optimization.



Aqib A. Syed received his B.S. degree in Aerospace Engineering from the Georgia Institute of Technology in 2019. He is currently an M.S. student in the Aerospace Engineering department at the same institution, and will graduate with this degree in 2021. His interests include controls, optimization, and robotics.



John B. Mains received his B.S. and M.S. degree in Electrical and Computer Engineering from Georgia Institute of Technology in 2016 and 2020, respectively. His interests include controls, optimization, and estimation.



Eric Feron is currently a Professor of Electrical and Computer Engineering at King Abdullah University of Science and Technology. He trained at Ecole Polytechnique (France), Ecole Normale Supérieure (France), and Stanford University (US). He was a professor at MIT for 12 years before joining Georgia Tech in 2005 and KAUST in 2019. He is interested in the applications of Control Systems, Computer Science, and Optimization to develop advanced aerospace embedded systems. His work includes departure planning for congested airports, aerobatic

drone flight, model-based systems engineering and code generation, and innovative robotic mechanisms



Louis Sutter received the diplôme d'ingénieur from Institut Supérieur de l'Aéronautique et de l'Espace (ISAE-SUPAERO), France, and the MS degree from the Georgia Institute of Technology, USA, in 2019. He was a control engineer at Safran Electronics and Defense and currently working at Dassault Aviation, Paris, France.



Philippe Baufreton is an senior expert engineer in the safety area at Safran Electronics and Defense in Massy, France. He is currently working on prospective research in the area of embedded software.



François Neumann is currently working as director of Research and Technology program at Safran Electronics and Defense in Massy, France.