Optimal Multiculture Network Design for Maximizing Resilience in the Face of Multiple Correlated Failures

Yasmany Prieto 1, Nicolás Boettcher 1,2, Silvia Elena Restrepo 3 and Jorge E. Pezoa 1,*

1 Departamento de Ingeniería Eléctrica, Universidad de Concepción, Concepción 4070386, Chile; yprieto@udec.cl (Y.P.); nboettcher@udec.cl (N.B.)
2 Escuela de Informática y Telecomunicaciones, Universidad Diego Portales, Santiago 8370190, Chile
3 Departamento de Medio Ambiente y Energía, Universidad Católica de la Santísima Concepción, Concepción 4090541, Chile; srestrepo@ucsc.cl

* Correspondence: jpezoa@udec.cl; Tel.: +56-41-220-3333

Received: 18 March 2019; Accepted: 24 May 2019; Published: 31 May 2019

Abstract: Current data networks are highly homogeneous because of management, economic, and interoperability reasons. This technological homogeneity introduces shared risks, where correlated failures may entirely disrupt the network operation and impair multiple nodes. In this paper, we tackle the problem of improving the resilience of homogeneous networks, which are affected by correlated node failures, through optimal multiculture network design. Correlated failures regarded here are modeled by Shared Risk Node Groups (SRNGs) events. We propose three sequential optimization problems for maximizing the network resilience by selecting as different node technologies, which do not share risks, and placing such nodes in a given topology. Results show that in the 75% of real-world network topologies analyzed here, our optimal multiculture design yields networks whose probability that a pair of nodes, chosen at random, are connected is 1, i.e., its ATTR metric is 1. To do so, our method efficiently trades off the network heterogeneity, the number of nodes per technology, and their clustered location in the network. In the remaining 25% of the topologies, whose average node degree was less than 2, such probability was at least 0.7867. This means that both multiculture design and topology connectivity are necessary to achieve network resilience.

Keywords: correlated failures; multiculture topology; network diversity; network vulnerability; network robustness; resilience; software risks

1. Introduction

Under normal circumstances, data networking systems are designed to provide connectivity to all its nodes while, simultaneously, managing limited resources such as bandwidth, buffers, and the number of simultaneous connections. In the presence of failures or attacks, the design problem becomes very challenging because it must jointly provide some level of connectivity to the operating nodes, using protection schemes, manage the available resources, and offer restoration schemes. Thus, the purpose of the resilient design is to ensure both that a large portion of a communication network remains connected after a failure occurs and recovers promptly. In the literature, this is referred to as the reliable path provisioning problem, and such issue evidences the fundamental trade-off between providing reliable paths and efficiently utilizing the network resources. Lastly, correlated failures affecting network nodes have raised the attention of researchers because they impact multiple nodes, thereby their consequence on both users and network operators is severe [1–5]. Correlated failures may be triggered from natural phenomena, such as earthquakes and hurricanes, or may be
induced intentionally by men, as in the case of weapons of massive destruction, electromagnetic pulses, or cyber attacks.

Data networking systems are highly homogeneous because the trend in networking has been that all the technologies, at each layer of the architecture, must converge to a single one. The main effect of such a tendency is that most of the nodes are purchased from a single vendor and, consequently, they turn out to be identical or very similar devices. Data networks designed in this fashion are termed as monoculture networks. From joint management, economic, and interoperability point of view, monocultures are appealing. However, operating monoculture networks may introduce severe problems to their survivability. For instance, the lack of diversity in monoculture networks introduces shared risks to attacks/failures such as exploits (0-day vulnerabilities). Thus, a single attack/failure may affect multiple nodes and entirely disrupt the network operation.

Nowadays, network operators have at hand new, flexible, compatible, and more importantly diverse technologies for managing networked systems. For instance, Software-defined Networking (SDN) allows network operators to manage and control networking devices from multiple vendors [6]. Besides, Network Function Virtualization (NFV) techniques implement Network Functions (NFs) exploiting software virtualization, and such functions are executed on commodity hardware from multiple vendors [7]. Practical examples of how these two technologies have enabled multivendor implementations are: The CloudNFV platform for cloud computing, [8], the SDN-based Packet Transport Network operated by China Mobile [9], the Optical SDN designed by China Telecom [9], and the NFV-based Service Orchestration implemented by Anuta Networks, [10]. Remarkably, the multivendor interoperability trend has been stated as a requirement in 5G implementations and applications such as Machine-to-Machine (M2M) and Internet of Things (IoT) [11].

Since the tide is turning now towards multivendor environments, we raise the following question: May we exploit the available node technology diversity to improve the resilience of an entire network, which faces multiple correlated node failures, by introducing in the design process multiculture networks? This seems to be a valid question because the diversity in biological systems is indeed a valuable commodity for survivability, and researchers have shown that provisioning an adequate number of different species may be one reason for preserving biodiversity [12].

In this paper, we tackle the problem of improving network resilience, in the presence of correlated failures, by carrying out an optimal multiculture network design. To do so, we divided this complicated problem into simpler, sequential optimization problems. First, we pose a constrained optimization problem for selecting as many different technologies as possible, which do not share common risks while are capable of communicating the network nodes. We termed this problem as “The optimal selection of the technology set.” Second, we propose another constrained optimization problem for selecting the number of nodes to be used from each technology, in order to fulfill a network design requirement with a Capital Expenditure (CAPEX) restriction. We termed this problem as “The fair technology distribution problem.” Lastly, we state yet another constrained optimization problem for placing the nodes within the network topology so that its resilience is maximized. We termed this last problem as “The reliable node placing problem.” The correlated failures regarded in this paper are modeled using Shared Risk Node Groups (SRNGs). These failures represent here cyber attacks, which aim to take advantage of specific vulnerabilities shared in all the network nodes belonging to the same SRNG event. We comment that, in the eight real-world network topologies considered here, our multiculture design enhances network resilience as compared to a monoculture design. In fact, results show that the proposed multiculture network design, which regards individual and shared node risks, can cope with multiple node failures induced by correlated SRNG events. In doing so, it efficiently trades off the number of technologies to be used (i.e., the degree of heterogeneity in the network), the number of nodes to specify per technology, and their location in the network topology. We note that, when our algorithms select larger sets of technologies, an attacker should make more efforts to compromise the network integrity and the impact caused by a particular attack on the network infrastructure is reduced. Additionally, our results also show that the node placing
algorithm attempts to cluster nodes belonging to the same technology, and remarkably, it locates the most vulnerable technologies at sites where the impact of multiple node failures on the entire network connectivity is reduced. Lastly, we comment that the location method proposed here exhibits better results, concerning the Average Two-Terminal Reliability (ATTR) metric than our earlier method [13].

The rest of this paper is organized as follows. In Section 2, we present and summarize the related work in the area. In Section 3, we introduce the principles of multiculture network design that we have employed in our work for increasing resilience in the face of correlated attacks. In Section 4, we explain our methodology, mathematically define the problem of redesigning a network topology, introduce our correlated failure model, and formulate the three optimization problems used to carry out the multiculture network design. Next, we introduce the resilience metrics used here and present the search algorithms we developed for solving the above-mentioned optimization problems. In Section 5, we present and compare the numerical results achieved by our algorithms. Lastly, in Section 6, we draw the conclusions of the paper.

2. Related Work

Diversity has been accepted as a method that plays a decisive role in network resilience [14–16]. Furthermore, it has been exploited as a mechanism to avoid fate sharing during correlated cyber-attacks, large-scale natural disasters, “buggy” software updates, etc. Sterbenz et al. provide in [14,15] a framework for resilience in communication networks. They formally define the terms resilience, reliability, survivability, and disruption tolerance in communication networks, and present diversity as an important requirement to deal with attacks of intelligent adversaries. A diverse system ensures that, under a correlated attack, all its parts are unlikely to share the same fate. Consequently, they can maintain a partial system operation. In [16] a systematic approach is conceived to build a resilient network, taking actions in a control loop to respond to attacks and recover to normal operation. In such a framework, redundancy and diversity are exploited as defensive mechanisms to maintain reliability in the presence of software faults.

Unlike in communication networks, diversity is a well-established concept for increasing reliability and robustness in software engineering. Pioneering research works such as [17–21] have developed concepts like N-version programming and data or instruction set randomization to introduce diversity. Regarding N-version programming, the term “natural diversity” was coined in [21] and described the existence of different pieces of either software or Operating Systems (OSs) with the same functionality, which appear spontaneously in the market and are supposed not exhibit common vulnerabilities. Examples of natural diversity can be found in web browsers, firewalls, virtual machines, routers, etc. In [22,23], the research focus was to disclose which applications and OSs that are available in the market, offer mutually exclusive software risks. Remarkably, more than 50% of the analyzed OSs had, at most, one non-application common vulnerability that can be remotely exploited, while up to 98% of the analyzed software could be exchanged by one with the same functionality, yet with different not-common vulnerabilities. Also, from vulnerabilities disclosure websites, such as [24–28], it can be observed that from the extensive list of data routers available in the market, only a few of them share the same vulnerability risks.

Some research works have been carried out to increase network robustness exploiting mutually exclusive software risks, [13,29–31]. In these works, nodes belong to mutually exclusive risk classes and when a failure occurs at a node, all the nodes in the same class fail simultaneously. Thus, researchers develop algorithms aiming to increase the network connectivity based on node diversity and the way such nodes can connect to each other. In [29] a grid network is created and the number of network devices is divided evenly among the classes. Next, every node in each class is linked to nodes belonging to a different class. Thus, the graph partitioning algorithm creates a topology that maintains a connected network when a single class of nodes breaks down. Caballero et al. introduced two methods for increasing the network resilience against software defects, bugs, and vulnerabilities affecting network routers [30]. The methods aim to locate network nodes using graph clustering and
graph partitioning. Routers were clustered according to their risk classes, and the network connectivity
was maximized when any class was removed from the graph. We note that, since in both works node
classes were considered to be homogeneous, the optimal class balance turned out to be a uniform
distribution of nodes per class. This is a major difference to our work. In [31] the Diversity Assignment
Problem is proposed: find the optimum assignment, from classes to network nodes, that maximizes
the average connectivity among end-nodes. In a case study, they showed that by employing three
classes of nodes, which fail independently and with different probabilities, a more resilient network
can be obtained as compared to a monoculture topology, which is composed solely by nodes from the
most reliable class. The optimum assignment is reached when for every class there is at least one path,
consisting of nodes of the same class, set up among every pair of end-nodes. Prieto et al. proposed
in [13] to decompose the network diversification problem in two subproblems: One for identifying the
number of nodes per class should be used to minimize the entire network vulnerability, and another
to determine the location of those different node classes to maximize the average connectivity of the
entire network. The work presented here is a substantial extension of the ideas exposed in [13]. The key
differences between the works are: (i) Here the network design starts one step earlier by selecting the
network technologies; (ii) classes are mapped here to technologies, bringing our earlier theoretical
work to a more applied setting; (iii) the vulnerability index introduced in [13] is now defined through
the number of risks exhibited by a technology; (iv) the devices’ placement method is improved here
by formulating an optimization problem that aims to locate the more resilient nodes at the most
vulnerable network zones; and (v) novel methods for solving the proposed optimization problems are
presented here.

Other approaches have been also employed for improving resilience and survivability. Zhang et al.
presented a new model for network survivability based upon network heterogeneity [32]. They introduced the concept of diversity space, where different network elements, like OSs,
communication media, service models, network protocols, routing mechanisms, etc., were mapped
onto different dimensions. Equipped with this definition for a diversity space, the authors used
the distance between network elements as a metric for stating that, the larger the distance between
network elements, the higher the diversity and smaller the network vulnerability. In [33], Caesar and
Rexford proposed a bug tolerant router. This router contains several virtual implementations running
in parallel inside the same hardware. The idea is that software diversification makes unlikely that
every implementation fails at the same time. Finally, the output of the virtual instances enters a voting
process that selects the router output.

Another two different applications of network diversity for increasing network survivability
and reliability were introduced in the contexts of cyber-security [34–37] and virus contention [38–41].
In [34,35] attack graphs and attack paths are defined as the ways an attacker can get access to a
network asset. Security metrics were designed to characterize how difficult it is for the attacker to
exploit the security mechanism of each node between him and the asset. Thus, node diversity imposes
independent efforts from the attacker to get access to each of them. In [36] Zhang et al. proposed
both the least attacking effort and the average attacking effort metrics to compute a distance between
an attacker and an asset. These metrics were based upon the number of hops and the number of
different types of nodes separating the attacker and the asset. Consequently, the more diverse the
types of nodes the more resilient the network in the face of 0-day attacks. In [37], the authors aimed
to allocate heterogeneous security mechanisms at the network nodes, thereby making difficult the
access of an attacker to a target asset of interest. Their main research idea relied on allocating nodes in
such a way that neighbors should not share the same vulnerabilities. This idea produced a decrease
in the severity of cyber-attacks by reducing the repetition of a single vulnerability in every attack
path. To avoid malware spreading the theory of perfect coloring, which aims to prevent that two
neighboring nodes share the same color, was used. In [38], the authors established a relationship
between the average degree and the number of necessary classes required to avoid the emergence of a
giant component in random networks. In [39], three random-distributed techniques were developed
to sub-optimally solve the NP-hard perfect coloring problem in non-exponential time. Huang et al. proposed the graph multicoloring problem to minimize the number of shared software executed on neighboring nodes [40]. Should malware compromises software in one node, this would stay contained in the subgraph containing the node and the neighbors with the common vulnerability. Temizkan et al. considered shared vulnerabilities between software variants and proposed a software allocation mechanism, based on combinatorial optimization and linear programming, that was applied on scale-free networks prone to be infected by viruses [41]. As a direct result, such methodology increased the network resilience against virus and worms attacks.

3. Rationale

Before presenting the rationale of our work, we formally define the terms monoculture and multiculture technology in a communication network.

**Definition 1.** A data communication network is defined as a monoculture if the technology used to implement the networking nodes is homogeneous. More precisely, the network technology is a monoculture if all its communication nodes belong to the same vendor and the implementations of their OS and protocol stack are the same.

**Definition 2.** A data communication network is defined as a multiculture if the technology used to implement the networking nodes is heterogeneous. More precisely, the network technology is a multiculture if all its communication nodes either do not belong to the same vendor, execute different OSs, or employ different protocol stack implementations.

We note that Definitions 1 and 2 transpire from the diversity space introduced in [32] to represent the functional capabilities of network nodes and architectures.

Multiculture network design can offer to network architects clear advantages as compared to monoculture networks. Figure 1 depicts three networks, with different technologies and different node locations in the topology, showing how a proper multiculture design can improve network resilience. The first case, depicted in Figure 1a, is a monoculture network, where only one kind of technology is employed by all the nodes. The problem here is that one type of exploit is enough to attack all the nodes and, consequently, induce multiple failures in the network. The second case, depicted in Figure 1b, exhibits some degree of diversity. In fact, three technologies are deployed on the network, and are represented by different colors. It can be observed that should a failure or an attack occur on the orange nodes, the rest of the network would remain disconnected because of the improper location of such nodes in the topology. The third case, depicted in Figure 1c, shows that a multiculture network design, where several technologies coexist without shared common risk, in conjunction with a proper location of these diverse technologies, effectively reduces the post-failure lack of connectivity as compared to both a monoculture network and a multiculture network incorrectly designed. In this work, we take into consideration these issues and aim to design multiculture networks that are minimally affected by failures or attacks to a single technology.

![Figure 1](image-url)
The optimal multiculture network design involves the selection of as many different technologies as possible, which do not share common risks, to be properly placed in an existing network topology. This problem represents a huge challenge in terms of modeling, the definition of optimization functions and their associated interdependent constraints, and also the emergence of huge search spaces where feasible solutions must be found. The approach in this work is to simplify this complicated problem by breaking it down into simpler sequential optimization problems.

4. Methodology

In this section, we describe the materials and methods used in our research. First, we present the mathematical models used to represent a communication network and its correlated failures. Next, we formulate three sequential optimization problems, which correspond to the core of the multiculture network design method. The first problem introduces diversity in the selection of node technologies by finding the maximum number of different technologies that do not share common risks. The second problem optimally selects the number of nodes of each technology that must be specified to maximize the network resilience. The third problem optimally specifies the location of each technology on each node, on a given network topology, in order to minimize the impact of a correlated failure on the entire network connectivity. Since the above-mentioned problems are NP-hard, we also introduce the search algorithms we developed for solving the optimization problems. The materials used in this paper correspond to both the eight real-world topologies that are commonly used in the literature to assess networking methods and the reliability metrics used to evaluate the results of the proposed methods. In summary, we state that the research questions guiding our work are: (i) What is the optimal number of technologies that can be employed to increase the diversity in a given network? (ii) What is the optimal number of nodes of each technology that are required to maximize the resilience of the entire network? and (iii) What is the optimal node location of each technology for maximizing the network resilience?

4.1. Problem Statement, System Model, and Correlated Failures Model

As mentioned earlier, the goal of this paper is to devise an optimal multiculture network capable of improving the network resilience when correlated failures impair simultaneously several nodes. The communication network topology is mathematically represented by the undirected graph $G = (V, E)$, where $V = \{1, 2, \ldots , n\}$ is the set of communication nodes and $E = \{(u, v) : \text{nodes } u \text{ and } v \text{ are connected}\}$ is the set of communication links between nodes. Suppose that a network architect must carry out a technological update of network nodes. To do so, a multiculture set of nodes may be used to replace the existing ones, thereby different technologies, i.e., vendors, OSs, and protocol stack implementations, can be introduced in the topology. We denote here by $K = \{1, 2, \ldots , N\}$ the set of $N$ different networking node technologies available to the architect. We assume also that the $N$ available technologies implement a joint protocol stack with a set of $L$ different communication protocols, where $Y_{il} = 1$ (correspondingly, $Y_{il} = 0$) denotes that a node, equipped with technology $i$, is able to (correspondingly, incapable of) communicate to other nodes through protocol $l$. Next, let us assume that the set $K' = \{1, 2, \ldots , \kappa\}$ represents the technologies to be implemented in the network devices during the upgrade. (This set is defined by solving the problem in Section 4.2.1).

We are interested here in modeling correlated failures triggered by breakdowns or attacks which diminish the connectivity of the infrastructure by impairing several nodes at the same time and for a long period. Thus, we assume here both that each regarded technology is prone to fail or to be attacked without recovery and that they share common risks.

**Definition 3.** A SRNG in a data communication network is a set of nodes that may be affected by a common failure to the infrastructure under the condition that they share a common failure risk.
Suppose now that there exists a set \( \mathcal{A} = \{A_1, A_2, \ldots, A_M\} \) of \( M \) different SRNG events that may induce correlated failures to the network. Furthermore, assume that each event \( A_i \in \mathcal{A} \) has a probability of occurrence of \( p_i \). Consequently, when the SRNG event \( A_i \) happens, the set of nodes \( V \) can be partitioned into two sets: \( V_{A_i} \) and \( V^c_{A_i} \), where the former set denotes the collection of all networking nodes sharing the common risk associated to the SRNG event \( A_i \), and the latter set denotes all those nodes unaffected by the event.

**Definition 4.** A Probabilistic Shared Risk Node Group (PSRNG) in a data communication network is a set of nodes that fail with a positive failure probability, in the event of a SRNG failure. More precisely, the failure probability of the \( i \)th node, conditional on the SRNG failure event \( A_i \), is denoted as \( p_{i,r} \) and satisfies: \( p_{i,r} > 0 \) for all \( i \in V_A \) and \( p_{i,r} = 0 \) otherwise.

**Definition 5.** We say that the nodes \( i \) and \( j \) belonging to a data communication network are correlated if \( p_{i,r} \) and \( p_{j,r} \) are both positive for the \( A_r \) PSRNG. Moreover, upon the occurrence of the \( A_r \) SRNG event, these probabilities are identical and mutually exclusive for all the pairs of nodes in \( V_{A_r} \), that is: \( p_{i,r} = p_{j,r} = p_r \) for all \( i, j \).

Following [2], we assume that only one PSRNG event may occur at a time. This means that the \( M \) shared risks defining the PSRNG events are mutually exclusive; therefore: \( \sum_{r=1}^{M} p_r = 1 \). This otherwise arbitrary definition has been effectively used in the networking community and makes sense in the context of the class of failures considered in this paper [2–4]. We note also that from Definitions 1 to 5, both monoculture and multiculture data network technologies can be affected by more than one SRNG event.

**Definition 6.** The resilience of a data communication network is defined as its ability to provide and maintain an acceptable level of node connectivity in the face of correlated failures triggered by the above specified PSRNGs.

In this paper, we will assess the resilience of a communication network after the occurrence of a PSRNG event by means of two metrics, which are mathematically defined in Section 4.4. One metric quantifies whether the network topology remains connected or is partitioned after an event, while the other metric quantifies how well-connected remains a network after the occurrence of a PSRNG event. With this ideas at hand, we can now introduce quantitative definitions for the resilience of a data communication network, which complement Definition 6.

**Definition 7.** A data communication network is defined as resilient to correlated failures triggered by the above specified PSRNGs, if its All Terminal Reliability (ATR) metric is equal to one. In addition, the average degree of resilience of a data communication network, in the face of correlated failures triggered by the above specified PSRNGs, is given by the ATTR metric.

### 4.2. Sequential Optimization Problems

The three sequential optimization problems mentioned at the beginning of Section 4 are specified next in full detail. For clarity in the presentation, Algorithm 1 is presented at the end of the section to summarize the workflow for solving these problems.

**Algorithm 1 Optimal Multiculture Network Design**

**Require:** \( G(V, E), X, Y, Q_1, Q_2, \ldots, Q_N, B \)

**Ensure:** \( T^*(V) \)

\[
M = \dim(X, 1); \quad K = \dim(X, 2);
\]

\[
K' \leftarrow \text{Optimal Selection of Technology Set } (X, Y), \text{ where } K' \leq K
\]

Compute \( a_k = \sum_{r=1}^{M} X_{kr} p_r \)

\( (n_1, n_2, \ldots, n_{K'}) \leftarrow \text{Fair Technology Distribution Problem } ((a_1, a_2, \ldots, a_{K'}), |V|, K', B) \)

\( T^*(V) \leftarrow \text{Reliable Node Placing } (G, (n_1, n_2, \ldots, n_{K'}), (a_1, a_2, \ldots, a_{K'})) \)
4.2.1. Optimal Selection of Technology Set

The goal of the proposed multiculture network design is to provide diversity in the communication network nodes. In this work, we aim to specify as many different compatible node technologies as possible, from a given pool of technologies, under the constraint that the selected technologies must communicate and are mutually exclusive in their risks, that is they do not belong to the same PSRNG. Consequently, an attack on some vulnerability would not damage more than one kind of technology. In this scenario, and relying on a database with both the node technologies available in the market and the information about their risks and communication protocols, it is possible to formulate the following optimization problem: Picking the maximum number of technologies that do not present common risks and are able to communicate between them. We have called this problem the optimal selection of technology set.

We depict, in Figure 2, an example of the problem showing the input data (in two tables) and the solution. Columns, at the left table, list four different types of SRNG events (r1 to r4 represent, respectively, the shared risks number 1 to 4), while at the right table list the different communication protocols that each node is equipped with (p1 to p3 represent, respectively, the communication protocols 1 to 3). Rows list seven different node technologies. In Figure 2, the optimal technology set was computed using exhaustive search and has been marked with a red box. Note that in this optimal solution the number of selected technologies is maximal, technologies do not exhibit shared risks, and they are capable of communicating, at least, by one available protocol. We note that, we have followed the seminal work [42] and used a risk matrix representation to characterize the software risks and their correlations with other network nodes for failure correlation analysis.

\[
\mathbf{\text{r}^* = \arg\max_{\theta \in \mathcal{T}} \sum_{i=1}^{N} T_{\theta_i}}
\]

Figure 2. Left: Matrix representation of the SRNG events or risks for each technology implemented at the networking nodes. Right: Communication protocol matrix for different networking node technologies.

The first problem tackled in this paper is the optimal selection of a technology set, that is, optimally specifying how many and which technologies are needed to maximize the diversity of the entire network. This problem can be mathematically posed as:
subject to:

\[ \sum_{i=1}^{N} X_{ir}T_i \leq 1, \text{ for each } r, \text{ (single SRNG constraints) } \quad (2) \]

\[ T_iT_j \leq Y_iY_j^T, \forall i,j, \text{ (common protocol constraints) } \quad (3) \]

\[ T_i \in \{0,1\}, \forall i, \text{ (4) } \]

where \( i, j, \) and \( r \) represent, respectively, the \( i \)th and \( j \)th technology, as well as the \( r \)th SRNG event in the network, \( T_i \) is a binary variable indicating the presence, \( T_i = 1 \) or absence, \( T_i = 0 \), of technology \( i \) in the solution, \( X_{ir} = 1 \) represents that the shared risk \( r \) affects the technology \( i \) and \( X_{ir} = 0 \) represents otherwise. \( Y_i = (Y_{i1} \ Y_{i2} \cdots \ Y_{iL}) \) is a row vector containing all the communication protocols offered by technology \( i \). In addition, \( T \) is the search space, which corresponds to the collection of every possible combination of technologies that can be selected out of the \( N \) available classes, while \( \mathbf{o}^* = (T_1 \ T_2 \ \cdots \ T_N) \) is an element of \( T \) specifying the optimal selection. For the sake of notation, we introduce also the risk vector associated to the \( i \)th technology as the row vector \( X_i = (X_{i1} \ X_{i2} \cdots \ X_{IM}) \), the risk matrix \( X = (X_1 \ X_2 \cdots X_N)^T \), and the communication protocol matrix \( Y = (Y_1 \ Y_2 \cdots Y_N)^T \). We recall that the parameters \( M, N, \) and \( L \) are the total number of SRNG events, technologies, and protocols respectively.

The first set of \( M \) constraints ensures that if a particular technology belongs to the optimal solution, it is the only one exposed to the SRNG event \( r \). We note that these constraints do make the problem unfeasible but they yield a monoculture network. In addition, the second set of, at most \( \binom{N}{2} \), constraints ensures that technologies are not orthogonal in terms of communication protocols, that is, there must exist at least one shared communication protocol between technologies \( i \) and \( j \), should they appear in the optimal solution. (Consequently, such constraint was formulated in terms of the dot product between the communication protocol vectors associated to every pair candidate technologies.) Finally, we mention that the failure probability associated to the \( r \)th SRNG event can be computed in practice from the risk matrix as:

\[ P_r \triangleq \sum_{k=1}^{N} X_{kr} \sum_{s=1}^{M} X_{ks}, \quad (5) \]

which means that such probability is given by the frequency of occurrence of a SRNG event among the available technologies in the market.

4.2.2. Fair Technology Distribution Problem

The solution to the problem of Section 4.2.1 provides the \( \kappa \) technologies, out of the \( N \) available, that will be used in the design. Most of the time, the number of these available technologies is less than the number of nodes in the network, meaning that several nodes will be using the same technology. The number of required devices, per selected technology, to minimize the vulnerability of the entire network is calculated by fairly balancing the total number of SRNG events among the network devices. This design step depends on the number of nodes in the analyzed topology and, in practice, is constrained by a fixed CAPEX. The problem is mathematically stated as:

\[ n^* = \arg\min_{n \in \mathbb{N}} \sum_{k=1}^{\kappa} \left( \frac{a_{k}n_k}{n} - \frac{\bar{a}_Cn}{\kappa} \right)^2, \quad (6) \]
subject to:

\[ \sum_{k=1}^{\kappa} n_k Q_k \leq B, \quad \text{(the CAPEX constraint)}, \quad (7) \]

\[ \sum_{k=1}^{\kappa} n_k = n, \quad \text{(the number of nodes constraint)}, \quad (8) \]

where \( n = (n_1, \ldots, n_{\kappa}, \ldots, n_{\kappa}) \) is a vector of \( \kappa \) elements that specifies the number of nodes of each technology, \( N = V^\kappa \) represents the search space in the optimization problem and corresponds to the collection of every possible combination of number of nodes per technologies that can be selected, \( Q_k \) is the cost, in some predefined currency, of each node belonging to the \( k \)th technology, \( B \) is the total CAPEX available to purchase network nodes, and \( \alpha_c = n^{-1} \sum_{k=1}^{\kappa} a_k n_k. \) (We refer to [13] for the details on how to obtain the value of \( \alpha_c \)).

The term \( a_k \) is a key parameter termed as the risk index associated to the \( k \)th technology. We introduced this parameter for the first time in [13], and in this paper, we redefine it formally in a more practical manner using the formula:

\[ a_k \triangleq \sum_{r=1}^{M} X_{kr} P_r, \quad (9) \]

which means that the risk index of each technology is given by the failure probability of each SRNG event, disclosed in the market, that affecting it. Note that we have exploited the assumption about the SRNG events being mutually exclusive.

### 4.2.3. Reliable Node Placing Problem

The last step in the sequential design method proposed here is to optimally place the selected node technologies on a given topology. The idea of the placing method is that the failure impact of the more vulnerable technologies should be as low as possible on the network connectivity. We remark that, when a node fails, it immediately affects its communication links; therefore, a proper network design must minimize the number of links affected by the failure of an entire set of nodes belonging to the most vulnerable technology. From Network Science theory, we know that the clustering coefficient from each technology in the network is a proper metric to assess the impact of such correlated failures on the network connectivity [43].

With this rationale in mind, we mathematically formulated the reliable node placing problem as:

\[ T^*(V) = \max_{T(V) \in \mathcal{M}} \sum_{k=1}^{\kappa} a_k I_k a \sum_{(u,v) \in E^{(k)}} e^{(k)}_{uv}, \quad (10) \]

subject to:

\[ \sum_{u=1}^{n} 1_{\{T(u)=k\}} = n_k, \quad \text{(the number of technologies constraint)}, \quad (11) \]

where \( k = 1, 2, \ldots, \kappa, \) \( T(V) : V \rightarrow K' \) is a mapping from \( V \) to \( K' \) assigning to the \( u \)th node the \( T(u) = k \) technology, \( a = \sum_{k \in K'} a_k \), and \( \mathcal{M} \) represents the search space of all possible mappings for assigning all \( \kappa \) technologies to \( n \) nodes. \( G^{(k)} = (V^{(k)}, E^{(k)}) \subset (V, E) \) is the network topology resulting after a failure affecting all the nodes of the \( k \)th technology, \( e^{(k)}_{uv} = 1 \) if \( (u,v) \in E^{(k)} \) represents a working link after the failure of the \( k \)th technology, and \( e^{(k)}_{uv} = 0 \) represents otherwise. The number of connected components in \( G^{(k)} \) is represented by \( I_k \), and \( 1_{\{T(u)=k\}} = 1 \) is the indicator function stating that the \( u \)th node belongs to the \( k \)th technology.
We note that in the cost function Equation (10), the inner summation aims to maximize the number of working links after a failure of the kth technology, while the term \( l_k \) penalizes the existence of a large number of connected components after a failure. Lastly, we note also that, by introducing the number of connected components emerging after a failure in the cost function Equation (10), we aim to maximize the resilience of a data communication network according to Definitions 6 and 7.

4.3. Efficient Search Algorithms Based on Transformations and Metaheuristics

In this section, we describe the algorithms we developed for solving the sequential optimization problems formulated in Section 4.2.

4.3.1. Optimal Selection of Technology Set

The technology diversity problem can be transformed into equivalent formulations, to obtain a more convenient representation that reduces it into a well known NP-hard problem termed as “The Clique Problem.” The first step of the transformation is blending the risks and communication protocol matrices into the so-called compatibility matrix \( C \). Here, the element \( C_{ij} \) in the compatibility matrix \( C \) is equal to “1” if the pair of technologies \( i \) and \( j \) meet both constraints jointly, and is equal to “0” otherwise. Figure 3 shows an example of the compatibility matrix for the case depicted in Figure 2.

![Compatibility matrix](image-url)

**Figure 3.** Compatibility matrix obtained from the example depicted in Figure 2.

Based on the compatibility matrix, the problem is transformed in finding the largest set of jointly compatible technologies. If \( C \) is interpreted as an adjacency matrix, then it can be represented by a graph, and the above mentioned problem reduces to the well-known Maximum Clique problem. Clique definitions are rooted in social sciences [44] and the problem is part of Karp’s 21 NP-complete problems [45], also in [46] more information is available. A survey concerning maximum clique and related algorithms to solve it could be found in [47].
The simplest equivalent formulation as an integer programming problem, presented in [47] and termed as “the edge formulation of the problem,” is used in this work:

$$\mathbf{o}^* = \arg\max_{\mathbf{o} \in T} \sum_{i=1}^{N} T_i,$$

subject to:

$$T_i + T_j \leq 1 \quad \forall (i, j) \in E',$$

$$T_i \in \{0, 1\} \quad \forall i,$$

where $G' = (V', E')$ is the graph that transpires from the compatibility matrix, $\overline{G'} = (V', \overline{E'})$ is the complement graph of $G'$, and $\overline{E'} = \{(i, j) | i, j \in V, i \neq j \text{ and } (i, j) \notin E'\}$. $T_i$ is a binary variable that indicates if technology $i$ belongs to the maximum clique.

The edge formulation given before is also NP-hard; however, it has been implemented in software packages and its execution takes an acceptable amount of time for the problem sizes analyzed here.

4.3.2. Fair Technology Distribution

We propose to solve the fair technology distribution problem through the Genetic Algorithm (GA) technique. This technique belongs to the more general evolutionary algorithms, relies on natural selection ideas and genetic operators, such as mutation and crossover, and is highly employed in non-well-structured problems [48].

For the GA technique, we coded the chromosome, which represents a possible solution, in a fixed-length integer-valued string as depicted in Figure 4. The $j$th position in the chromosome of length $N$ denotes the $j$th node in the network. The $j$th chromosome position stores a non-negative integer value, say, $k$, which specifies that the $j$th node belongs to the $k$th technology.

Figure 4. Chromosome coding for the fair technology distribution problem.

Regarding the GA operators, we followed standard guidelines from the GA theory to set the algorithm parameters at recommended values. Thus, population size is set to 500 chromosomes. We employ the single point crossover, with a probability of 0.8 for executing the operation, as the crossover operator. For mutations, one position of the chromosome is selected randomly, and its value is changed, with a probability of 0.01, by one of the other technologies available in the design. For selection, we used the fitness proportional selection, implemented by a roulette wheel.

Following standard procedures to handle constrained optimization problems using GAs, we transformed ours into a non-constrained problem by adding the constraints as penalty functions to the objective function, [49–52]. Following [49], the penalty functions should increase their values as the generation number $g$ does, thereby adding more selective pressure on the GA to converge to feasible solutions. From these ideas, the fitness function, $f_1(g)$, at the generation number $g$ is given by:

$$f_1(g) = \left( \sum_{k=1}^{x} (a_k n_k - a_r n_r)^2 + 1.2^g \left( \max \left\{ 0, \sum_{k=1}^{x} n_k Q_k - B \right\} \right)^2 \right)^{-1},$$

where $G' = (V', E')$ is the graph that transpires from the compatibility matrix, $\overline{G'} = (V', \overline{E'})$ is the complement graph of $G'$, and $\overline{E'} = \{(i, j) | i, j \in V, i \neq j \text{ and } (i, j) \notin E'\}$. $T_i$ is a binary variable that indicates if technology $i$ belongs to the maximum clique.

The edge formulation given before is also NP-hard; however, it has been implemented in software packages and its execution takes an acceptable amount of time for the problem sizes analyzed here.

4.3.2. Fair Technology Distribution

We propose to solve the fair technology distribution problem through the Genetic Algorithm (GA) technique. This technique belongs to the more general evolutionary algorithms, relies on natural selection ideas and genetic operators, such as mutation and crossover, and is highly employed in non-well-structured problems [48].

For the GA technique, we coded the chromosome, which represents a possible solution, in a fixed-length integer-valued string as depicted in Figure 4. The $j$th position in the chromosome of length $N$ denotes the $j$th node in the network. The $j$th chromosome position stores a non-negative integer value, say, $k$, which specifies that the $j$th node belongs to the $k$th technology.

Figure 4. Chromosome coding for the fair technology distribution problem.

Regarding the GA operators, we followed standard guidelines from the GA theory to set the algorithm parameters at recommended values. Thus, population size is set to 500 chromosomes. We employ the single point crossover, with a probability of 0.8 for executing the operation, as the crossover operator. For mutations, one position of the chromosome is selected randomly, and its value is changed, with a probability of 0.01, by one of the other technologies available in the design. For selection, we used the fitness proportional selection, implemented by a roulette wheel.

Following standard procedures to handle constrained optimization problems using GAs, we transformed ours into a non-constrained problem by adding the constraints as penalty functions to the objective function, [49–52]. Following [49], the penalty functions should increase their values as the generation number $g$ does, thereby adding more selective pressure on the GA to converge to feasible solutions. From these ideas, the fitness function, $f_1(g)$, at the generation number $g$ is given by:

$$f_1(g) = \left( \sum_{k=1}^{x} (a_k n_k - a_r n_r)^2 + 1.2^g \left( \max \left\{ 0, \sum_{k=1}^{x} n_k Q_k - B \right\} \right)^2 \right)^{-1},$$

where $G' = (V', E')$ is the graph that transpires from the compatibility matrix, $\overline{G'} = (V', \overline{E'})$ is the complement graph of $G'$, and $\overline{E'} = \{(i, j) | i, j \in V, i \neq j \text{ and } (i, j) \notin E'\}$. $T_i$ is a binary variable that indicates if technology $i$ belongs to the maximum clique.
which clearly addresses a minimization problem through the fitness proportional selection. Note that, the smaller the cost function, the higher the probability that a chromosome should be selected to the next generation and vice versa. We also note that Equation (15) contains both the cost function Equation (6) (at the left-hand side of the denominator) and the penalty function that represents the inequality constraint Equation (7) (at the right-hand side of the denominator). However, the number of nodes constraint Equation (8) is not included as a penalty function in Equation (15) because is directly handled by the chromosome coding.

Lastly, the stopping criteria for the GA considers two options: (i) The number of iterations carried out achieves a maximum, predefined number; and (ii) the absolute difference between the mean values of the cost function, in two consecutive generations, for the entire population is smaller than some predefined tolerance $\epsilon$.

The solution to the optimization problem, which is obtained at the final generation $g'$, will be the chromosome extracted from the population that reaches:

$$
\text{argmin}_{1 \leq h \leq P_1(g')} \sum_{k=1}^{K} \left( \alpha_k n_k^{(h)} - \alpha_c n \right)^2 .
$$

At this point, it is important to note that the fair technology distribution problem in Equations (6)–(8) was formulated considering the information about the nodes yet disregarding the connectivity information provided by the links. In our earlier work, we disclosed that the ATTR metric may not be improved by selecting the best combination of technologies [13]. In fact, sub-optimal technology distributions provided best solutions in terms of the ATTR metric. To overcome this issue, in this paper we have decided to modify the traditional GA methodology and generate, instead of a single solution, a list with the best solutions found by the GA method after its execution. This list of solutions will be used as the input to the reliable node placing problem. More precisely, the modified GA method ranks and stores, at each population number, a list with the 10% best feasible solutions. (The total number of different assignments of $t$ technologies to $n$ nodes is given by: $|\mathcal{N}| = (t + n - 1)!/n!(t-1)!$.) Thus, the list is updated in every generation of the GA, from the current chromosomes and the past results. Algorithm 2 depicts a pseudocode for the GA method proposed here.
Algorithm 2 GA for the fair technology distribution problem.

Require: $p_{\text{mut}}, p_{\text{cross}}, g_{\text{max}}$ (max iteration number), $\epsilon$

Ensure: $SL$ (10% ranked solution list)

$g = 0$;

$P(g)$: Initial population

Evaluate population ($P(g)$)

$SL(g) = 10\%$ ranked ($P(g)$)

Compute $MV(g)$

repeat

$P'(g) \leftarrow$ Single Point Crossover($P(g), p_{\text{cross}}$)

$P''(g) \leftarrow$ Mutation($P'(g), p_{\text{mut}}$)

Evaluate population ($P''(g)$)

$P(g + 1) \leftarrow$ Selection ($P''(g)$, proportional selection)

$g++$

Evaluate population ($P(g)$)

$SL(g) = 10\%$ ranked ($SL(g - 1) \cup P(g)$)

Compute $MV(g)$

until ($MV(g) - MV(g - 1) \leq \epsilon$) or ($g \geq g_{\text{max}}$)

$SL \leftarrow SL(g)$

4.3.3. Reliable Node Placing

We propose to solve also the reliable node placing problem using a GA method, under the proviso that each one of the best solutions found for the fair technology distribution must be used as an input. Thus, all the solutions found for each one of the different inputs are compared to obtain the maximum value for the reliable node placing problem.

For the GA technique, we coded the chromosome, which represents a possible solution, in a positive integer-valued string of length $n$ as depicted in Figure 5. The chromosome represents the list of nodes in the network, and the value in any string position represents the technology associated with that node. Note that the chromosome coding takes into account the problem constraints, which assign a specific number of nodes to each technology.

![Figure 5. Chromosome coding for the reliable node placing problem.](image)

Regarding the GA operators, we followed standard guidelines from the GA theory to set the algorithm parameters at recommended values. Thus, population size is set to 500 chromosomes. We employ the first order crossover, with a probability of 0.8, as the crossover operator for chromosomes. For mutations, swap mutation, which exchanges the value of two randomly chosen positions in the chromosome, was selected with a probability of mutation of 0.01. For selection, the fitness proportional selection was chosen again as in the fair technology distribution problem.
The fitness function was designed to exert selective pressure on the GA, that is, the larger the generation number $g$, the bigger the difference between the chances of selecting a better solution than a poor one, to pass to the next generation. Mathematically, the fitness function is given by:

$$f_2(g) = (1.1 \times 1.01^{g-1}) \sum_{k=1}^{K} \frac{a_k}{I_k^{(h)}} \sum_{(u,v) \in E^{(k)}} e_{uv}^{(k)},$$

(19)

The stopping criteria is the same one employed in Section 4.3.2; however, the mean value of the cost function, $MV_2(g)$, at the generation number $g$ is defined as:

$$MV_2(g) = \frac{1}{P_2(g)} \sum_{h=1}^{P_2(g)} \sum_{k=1}^{K} \frac{a_k}{I_k^{(h)}} \sum_{(u,v) \in E^{(h),(h)}} e_{uv}^{(k),(h)},$$

(20)

where $P_2(g)$ is the total number of chromosomes in the population at the generation $g$, $E^{(k),(h)}$ is the set of links that remain operative after a failure of the $k$th technology as specified by the $h$th chromosome in the population at the generation $g$, $e_{uv}^{(k),(h)}$ is a specific link of $E^{(k),(h)}$ as specified by the $h$th chromosome in the population at the generation $g$, and $I_k^{(h)}$ the number of connected components in $G^{(k),(h)}$, after a failure of the $k$th technology, as specified by the $h$th chromosome in the population at the generation $g$. (For the sake of notation, we have omitted in $E^{(k),(h)}, e_{uv}^{(k),(h)}, I_k^{(h)}$, and $G^{(k),(h)}$ their dependency on the generation number $g$.) Thus, the solution is determined, in the population of the last generation, by:

$$\argmax_{1 \leq h \leq P_2(g)} \sum_{k=1}^{K} \frac{a_k}{I_k^{(h)}} \sum_{(u,v) \in E^{(h),(h)}} e_{uv}^{(k),(h)}.$$ 

(21)

We remark that the solution for the reliable node placing problem is the maximum value among all the results obtained after executing the above mentioned procedure, for the best solutions found for the fair technology distribution problem. For the sake of notation, we have omitted an index in Equations (19)–(21) to denote this dependency. Lastly, Algorithm 3 indicates the way to solve the problem proposed here.

**Algorithm 3** GA for the reliable node placing problem.

**Require:** $p_{mut}$, $p_{cross}$, $g_{max}$ (max iteration number), $\epsilon$

**Ensure:** RNP (Decoded chromosome with maximum fitness)

$g = 0$

$P(g) :$ Initial population

Evaluate population ($P(g)$)

Compute $MV(g)$

repeat

$P'(g) \leftarrow$ First Order Crossover($P(g), p_{cross}$)

$P''(g) \leftarrow$ Swap Mutation($P'(g), p_{mut}$)

Evaluate population ($P''(g)$)

$P(g + 1) \leftarrow$ Selection ($P''(g)$, proportional selection)

$g++$

Evaluate population ($P(g)$)

Compute $MV(g)$

until ($MV(g) - MV(g - 1) \leq \epsilon \quad || \quad g \geq g_{max}$)

RNP $\leftarrow \max \{P(g)\)$
4.4. Resilience Metrics

We use two metrics to assess the resilience of the networks after solving the optimization problems mentioned above.

The ATTR metric quantifies how well-connected is a network after the occurrence of a failure event \cite{2,4,53}. The ATTR is effectively the probability that a pair of networking nodes, chosen at random, is connected. Thus, if a network is fully connected its value is equal to 1. Since we are considering here different failure events for different technologies, we modify first the traditional definition for the ATTR metric by parameterizing it in terms of the different SRNG event probabilities. That is, the ATTR of a network topology when the failure event associated to the $r$th SRNG event occurs is given by:

$$
ATTR(r) = \left( \frac{n}{2} \right) - \sum_{u \neq v} Z_{u,v}^r,
$$

where $\left( \frac{n}{2} \right)$ is the binomial coefficient and $Z_{u,v}^r$ is a binary variable that takes the value 1 if there is a path between the nodes $u$ and $v$ after a failure of the $r$th technology, and takes the value 0 otherwise.

Next, the ATTR of the network is computed as the weighted average over all the technology failures:

$$
ATTR = \alpha \sum_{k=1}^\kappa \alpha_k \cdot ATTR(k),
$$

where $\alpha = \sum_{j=1}^\kappa \alpha_j$.

After the occurrence of a failure event associated to a SRNG, the resulting working topology may remain connected or may be partitioned. Here we assess this effect in terms of the number of connected components arising after failure. A connected component is formally defined as a subgraph where any two nodes are connected to each other by paths, and there are no connections to other nodes in the supergraph modeling the network before failure \cite{43}. If the number of connected components is 1, then network is connected. We note that this quantity is related to the ATR metric commonly used in the networking community since the ATR is defined as 1 if the network is connected and 0 otherwise \cite{2,4}.

4.5. Topologies

In this paper, we have used eight real-world networks to assess the capability of the proposed multicultural design in improving their resilience. Figure 6 depicts the topologies and show their average degree. Networks in Figure 6a–g were extracted from Internet Topology Zoo \cite{54} and are commonly used in the research community as benchmarks. Moreover, infrastructures having different node degrees were selected to study their effect on the multicultural network design. In addition, topology in Figure 6h corresponds to the network connecting all the universities in Chile. Lastly, we comment that networks labeled as Navigata, Kreonet, and Reuna correspond to subgraphs of the original networks.
5. Results

5.1. Optimal Selection of Technology Set

We assessed our algorithm carrying out numerical calculations using the above mentioned topologies. The first experiment we conducted aims to evaluate the ability of the clique method in finding the optimum value, for the problem of selecting the technology set. In agreement with the network topologies in Figure 6, we used in our calculations a set of \( N = 15 \) different technologies and a total of \( M = 15 \) different possible SRNG events and \( L = 15 \) communication protocols. We randomly generated the risk matrix \( X \) sampling zeros and ones from independent and identically distributed (iid) Bernoulli random variables with probability \( p = 0.3 \) for a “1.” Fixing this probability to such value we controlled that the average number of SRNG events, per technology, is 4.5, a value that is consistent with those found in [25,26]. Similarly, the communication protocol matrix \( Y \) was randomly generated from independent and identically distributed Bernoulli random variables with probability \( p = 0.5 \) for a “1.” This likelihood also allows us to control that, on average, we will have communicating protocols between pairs of technologies.

In brief, we state that the method proposed in Section 4.3.1 was able to reach the optimum value: three technologies that do not share risks and can communicate with each other. In addition, we carried out experiments to compute the optimal number of selected technologies while increasing the number...
of available technologies, SRNG events, and communication protocols. In particular, we sampled $X$ from iid Bernoulli random variables with probability $q = 0.1, 0.2,$ and $0.3$ for all the technologies, while the number of available technologies, SRNG events, and communication protocols varied from 15 to 25. Figure 7 shows the results for the optimal number of selected technologies. We comment that when the range of available technologies is between 15 and 25, the proposed algorithm selects, for deploying onto the network topologies, between: (i) 2 and 5 when $p = 0.3$; (ii) 4 and 7 when $p = 0.2$; and (iii) 6 and 11 when $p = 0.1$. Remarkably, we note that as the likelihood of the SRNG events increases the optimal number of technologies decreases. This result is counterintuitive because we expect that as the diversity increases so does the resilience. However, the single SRNG event constraints stated in the optimization problem impose additional restrictions, which, in turn, forces the tradeoff between diversity and number of SRNG events, to reduce the number of technologies in the optimal set.

Since we sampled $X$, for all the technologies, from iid Bernoulli random variables with the same $p$ value, this means that all the SRNG events associated with the available technologies are statistically homogeneous. To introduce some heterogeneity, we randomly sample $p$, for each available technology, from a uniform discrete distribution in the range $[0.15,0.5]$. This sampled value is next used to sample the rows in the risk matrices from iid Bernoulli random variables. In this experiment, we fixed the number of available technologies, SRNG events, and communication protocols to 15. We generated 1000 risk matrices, computed the distribution for the number of selected technologies, and plotted the result in Figure 8. From the histogram, we can observe that the most likely number of selected technologies is again three.

5.2. Fair Technology Distribution

We present now the results associated to the experiments carried out to determine the optimal selection of the number of nodes, per technology, such that the vulnerability of the entire network is minimized subject to a capital expenditure constraint. From Section 5.1, only three different technologies are needed for the networks in Figure 6. In our calculations, we set: Technology 1 with $\alpha_1 = 6/15$ and $Q_1 = 1$ [a.u.], Technology 2 with $\alpha_2 = 5/15$ and $Q_2 = 2$ [a.u.], and Technology 3 with $\alpha_3 = 4/15$ and $Q_3 = 3$ [a.u.]. We comment that the nodes associated to the PSRNGs with the highest vulnerability index are also the most inexpensive ones. Figure 9 shows, for the design process of the Sprint network, the relationship between the maximum CAPEX available and the number of purchased nodes belonging to different PSRNGs. We note that, as the CAPEX amount increases, a better distribution of node technologies can be achieved, according to the technology risk indexes. This behavior comes from the fact that the CAPEX has no direct implication on the objective function.
Figure 7. The optimal number of selected technologies as a function of the available technologies for three different values of risk matrices.

Figure 8. The distribution of the optimal number of selected technologies for 1000 different realizations of the risk matrices. The number of available technologies was set to 15.

Figure 10 shows, for the case of the Sprint network topology, the effect of technologies’ risk index on the number of selected nodes per technology. Here the CAPEX constraint has been relaxed, i.e., the upgrade budget to purchase technology was set to infinity. We arbitrarily selected the values for the risk indexes to show how their different combinations affect the number of nodes to be selected for each technology. As expected, for node technologies with larger risk indexes (0.5 and above) the number of specified nodes is smaller than those with lower risk indexes (below 0.5). This behavior is intensified in situations where the risk indexes exhibit large differences among them. In such cases, technologies with smaller risk indexes turn out to be the most selected ones, as depicted in the four
cases at the right-hand side of Figure 10. On the contrary, when risk indexes exhibit similar values, the number of nodes per technology becomes evenly distributed.

Figure 9. Number of purchased nodes per technology as a function of the maximum CAPEX available during the design process of the Sprint network \((n = 11\) nodes).

Figure 10. Number of nodes per technology parameterized by different selections of risk indexes. The values \(\alpha_1, \alpha_2, \alpha_3\) on the x-axis correspond to different selections for the risk indexes. Topology: Sprint network \((n = 11\) nodes).

5.3. Reliable Node Placing

The third optimization problem was solved using data and results from the problem in the previous sections. The multiculture network design yielded by our method for each topology in Figure 6 are depicted in Figure 11. We note that a clustering effect is clearly observed in each of the designs. Furthermore, we note that, for most of the network topologies, the location of the different technologies ensures post-failure full connectivity for the remaining functioning nodes. In Kreonet and Reuna networks, however, maintaining full connectivity after a failure is not possible due to these topologies exhibit a low node degree. Remarkably, in cases like these two, the placing method properly assigns locations within the network according to the risk index of the technologies. In fact, the most vulnerable ones are allocated in places within the network that, after the occurrence of a failure, would compromise the network connectivity the least. This location mechanism is dictated by the formulation
of the optimization problem, which aims to maximize the number of available links after a failure, and minimize the number of connected components generated after failures.

![Network topologies](image)

Figure 1: Networks topologies used for placement methods evaluation.

The values achieved by our proposed multiculture network design for the resilience metrics regarded here are listed in Table 1. In particular, the results are listed at the white background rows of the Table and, for comparison, in the gray background rows we list the results of the multiculture network design proposed in our previous work [13]. The results clearly show that the design methodology proposed here outperforms our earlier method. Despite Kreonet topology, the values achieved by the ATTR metric are, at least, equal to or larger than those reported previously. The better performance obtained by our new approach relies on the network-resilient node-placing mechanism, which after a failure maximizes the number of available links and minimizes the number of connected components. For example, consider the Reuna topology. From Figure 11h, after a failure, only when the least vulnerable technology fails the network gets disconnected. This can also be observed at column 4 of Table 1 labeled as “Number of Post-Failure Connected Components ($t_1, t_2, t_3$).”

Figure 11. Optimal placing the network node devices for each technology for each network topology in Figure 6.
However, since in our earlier work we placed nodes minimizing the number of links affected after a failure such type of solution was not feasible for the Reuna topology. For the Kreonet topology, we can comment that the configuration that maximizes Equation (10) was not obtained in the top 10% solutions for Equation (6). Such a result is explained by the random nature of the GA method, which does not ensure finding the optimal value.

![Figure 12](image)

**Figure 12.** Relationship between the average degree of the network topologies and the average number of post-failure connected components from Table 1.

**Table 1.** Results from the multiculture network design for the second and third optimization problems.

| Network   | Nodes per Technology \( (n_1, n_2, n_3) \) | Rank Number / Total Number of Configurations | Number of Post Failure Connected Components \( (t_1, t_2, t_3) \) | ATTR Value | ATTR Values among Post-Failure Functional Nodes |
|-----------|-------------------------------------------|---------------------------------------------|-------------------------------------------------|------------|-----------------------------------------------|
| Sprint    | (2,5,4)                                    | 7/7                                         | (1,1)                                           | 0.4545     | 1                                             |
|           | (3,4,4)                                    | 1/7                                         | (1,1)                                           | 0.4327     | 1                                             |
| Abilene   | (3,4,4)                                    | 1/7                                         | (1,1)                                           | 0.4327     | 1                                             |
|           | (3,4,4)                                    | 1/7                                         | (1,1)                                           | 0.4327     | 1                                             |
| Arpanet19706 | (2,2,5)                                    | 5/5                                         | (1,1)                                           | 0.4722     | 1                                             |
| Navigata  | (2,4,5)                                    | 3/7                                         | (1,1)                                           | 0.4618     | 1                                             |
|           | (3,4,4)                                    | 1/7                                         | (1,1)                                           | 0.4327     | 1                                             |
| Kreonet   | (2,4,5)                                    | 3/7                                         | (1,1)                                           | 0.4036     | 0.7867                                        |
|           | (2,3,6)                                    | 3/7                                         | (1,1)                                           | 0.4461     | 0.8110                                        |
| Gridnet   | (2,3,4)                                    | 1/5                                         | (1,1)                                           | 0.4463     | 1                                             |
|           | (2,3,4)                                    | 1/5                                         | (1,1)                                           | 0.4463     | 1                                             |
| Napnet    | (1,2,3)                                    | 2/3                                         | (1,1)                                           | 0.4533     | 1                                             |
|           | (1,2,3)                                    | 2/3                                         | (1,1)                                           | 0.4533     | 1                                             |
| Reuna     | (2,4,5)                                    | 3/7                                         | (1,1)                                           | 0.4230     | 0.8578                                        |
|           | (4,4,3)                                    | 5/7                                         | (1,1)                                           | 0.3382     | 0.8114                                        |

In Figure 12 we depict the relationship between the average node degree of each network topology and the values of the average number of post-failure connected components listed in Table 1. A pattern can be observed since Reuna and Kreonet, which present the lowest average node degree, also happen to be the multiculture networks with higher number of average post-failure connected components. This implies that both network connectivity and heterogeneous technologies are necessary to achieve
network resilience in the face of correlated attacks. A more clear picture of the results of the multiculture network design can be observed from the last column of Table 1. Since ATTR metric is frequently used to assess the connectivity when links fail, the low values listed at the fifth column of the Table could be misleading. Calculating the ATTR among the post-failure functioning nodes is a more accurate approach to describe the remaining network connectivity in our scenario. An ATTR value equal to 1 means the functioning nodes after the failure can all be reached from other working nodes. In addition, ATTR values lower than 1 for Reuna and Kreonet are expected due to there is no single connected component after the occurrence of a failure.

Lastly, at the third column of Table 1, we provide both, the rank number and the total number of configurations provided by the fair technology distribution problem, to the optimal reliable node placing problem. In the column, we list first the rank number of the configuration that achieved the optimal resilience listed at the right-most column, and next, we list the total number of configurations provided by the fair technology distribution problem. Note that the best configurations supplied by the fair technology distribution problem do not always achieve a network topology with maximal resilience, in terms of the ATTR metric. Thus, the figures in Table 1 justify our decision about providing the top 10% solutions to the reliable node placing problem.

6. Conclusions

In our work, we proposed the idea of exploiting multiculture network design, i.e., introducing node technology diversity, as a means to provide resilience during a network upgrade process. The methodology presented here comprises a series of sequential optimization problems that address the different stages in the network design process: The technology selection, the specification of the number of devices per technology, and the network placement of the selected devices. We comment that our work is not only a contribution to the theory of network resilience through software diversity but also provides a practical methodology to network architects for achieving a resilient network design.

The solution to the first optimization problem presented here allowed us to specify, from a set of available technologies, the largest number of node implementations that do not share common risks. The larger the selected set of technologies is, the more the efforts an attacker should make to compromise the network integrity, and the less would be the impact caused by a particular vulnerability attack on the network infrastructure.

The solution to the second problem allowed to optimally calculate the number of network devices, from each technology, that must be deployed on the network. The key idea exploited by our method is to balance the number of SRNG events among the devices, and simultaneously, all those technologies presenting a larger number of vulnerabilities will be less represented in the network. Besides, the effect of the CAPEX assigned to the network architect on the technology diversification was analyzed. The risk index, which accounts for the number of vulnerabilities in one technology, was also redefined here in a practical manner.

The solution to the third problem enabled us to carry out the optimal placement of nodes on the network topology. Since the problem of computing the number of devices per technology was solved disregarding the topological information of the network, the GA-based solver was modified to supply not one but a group of best solutions. Such modification trades off between the number of nodes per technology and their location for increasing the network resilience, as shown by the results listed in Table 1. Results show also that in the 75% of real-world network topologies analyzed in this paper, the optimal multiculture network design proposed here yields networks with an ATTR metric of 1. This means that such networks remain connected after failure, since the ATTR represents the probability that a pair of nodes picked at random are connected. For the remaining 25% of the analyzed topologies, whose average node degree was less than 2, the ATTR was at least 0.7867. The latter results mean that both multiculture design and topology connectivity are necessary to achieve network resilience in the presence of correlated failures. Besides, results also show that certain network properties, like clustering, favor connectivity in the presence of correlated failures triggered
by common node vulnerabilities. Remarkably, the proposed design method locates the nodes on
the network in such a way that the most vulnerable nodes are assigned to locations where network
connectivity is affected the least upon a failure.

Our future research work on this subject will involve developing a new model for improving
network connectivity, which could be solved as a single optimization problem. To achieve feasibly
solutions, we will relax the constraint that technology risks must be exclusive.

Author Contributions: Y.P., N.B. and S.E.R. conducted the research. Y.P. and J.E.P. conceptualized the main
ideas and designed the methodology. Y.P. and N.B. developed algorithms and performed experiments. Y.P., N.B.
and S.E.R. validated the results. J.E.P. carried out project administration and supervision. All authors participated
in writing and reviewing the document.

Funding: This research was funded by CONICYT: FONDECYT Regular 2016 Folio 1160559, PCHA/Doctorado
Nacional Folio 2015-21150775 and PCHA/Doctorado Nacional Folio 2015-21150313.

Acknowledgments: The authors thank Professor Roberto Asín Achá of Department of Computer Science in
Universidad de Concepción for his insightful discussions in the “optimal selection of technology set problem.”

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study;
in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to
publish the results.

Abbreviations
The following abbreviations are used in this manuscript:

ATR All Terminal Reliability
ATTR Average Two-Terminal Reliability
CAPEX Capital Expenditure
GA Genetic Algorithm
iid independent and identically distributed
IoT Internet of Things
M2M Machine-to-Machine
NF Network Function
NFV Network Function Virtualization
PSRNG Probabilistic Shared Risk Node Group
OS Operating System
SDN Software-defined Networking
SRNG Shared Risk Node Group

References
1. Diaz, O.; Xu, F.; Min-Allah, N.; Khodeir, M.; Peng, M.; Khan, S.; Ghani, N. Network survivability for multiple
probabilistic failures. IEEE Commun. Lett. 2012, 16, 1320–1323. [CrossRef]
2. Lee, H.-W.; Modiano, E.; Lee, K. Diverse routing in networks with probabilistic failures. IEEE/ACM
Trans. Netw. 2010, 18, 1895–1907. [CrossRef]
3. Bassiri, B.; Heydari, S.S. Network survivability in large-scale regional failure scenarios. In Proceedings of
the 2nd Canadian Conference on Computer Science and Software Engineering (C3S2E ’09), Montreal, QC,
Canada, 19–21 May 2009; ACM: New York, NY, USA, 2009; pp. 83–87.
4. Neumayer, S.; Modiano, E. Network reliability under geographically correlated line and disk failure models.
Comput. Netw. 2016, 94, 14–28. [CrossRef]
5. Rahnamay-Naeini, M.; Pezoa, J.E.; Azar, G.; Ghani, N.; Hayat, M.M. Modeling stochastic correlated failures
and their effects on network reliability. In Proceedings of the 20th International Conference on Computer
Communications and Networks (ICCCN), Maui, HI, USA, 31 July–4 August 2011; pp. 1–6.
6. ONF White Paper. Software-Defined Networking: The New Norm for Networks; Technical Report; Open Networking Foundation, 13 April 2012. Available online:
https://www.opennetworking.org/images/stories/downloads/sdn-resources/white-papers/wp-sdn-new
norm.pdf (accessed on 25 August 2018).
7. Han, B.; Gopalakrishnan, V.; Ji, L.; Lee, S. Network function virtualization: Challenges and opportunities for
innovations. IEEE Commun. Mag. 2015, 53, 90–97. [CrossRef]
8. CloudNFV Taking NFV to the Cloud. Available online: http://cloudnfv.com/ (accessed on 25 August 2018).
9. ONF White Paper. Use Cases for Carrier Grade SDN; Technical Report; Open Networking Foundation, 20 October 2016. Available online: https://www.opennetworking.org/wp-content/uploads/2014/10/TR-538_Use_Cases_for_Carrier_Grade_SDN.pdf (accessed on 29 May 2019).
10. Anuta Networks. Case Study Network Service Orchestration for Multi-Vendor NFV; Technical Report; Anuta Networks: Milpitas, CA, USA, 2016.
11. Mehmood, Y.; Haider, N.; Imran, M.; Timm-Giel, A.; Guizani, M. M2m communications in 5g: State-of-the-art architecture, recent advances, and research challenges. IEEE Commun. Mag. 2017, 55, 194–201. [CrossRef]
12. Naeem, S.; Li, S. Biodiversity enhances ecosystem reliability. Nature 1997, 390, 507–509. [CrossRef]
13. Prieto, Y.; Pezoa, J.E.; Boettcher, N.; Sobarzo, S.K. Increasing network reliability to correlated failures through optimal multicuture design. In Proceedings of the CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Pucón, Chile, 18–20 October 2017; pp. 1–6.
14. Sterbenz, J.P.; Hutchison, D.; Çetinkaya, E.K.; Jabbar, A.; Rohrer, J.P.; Schöller, M.; Smith, P. Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines. Comput. Netw. 2010, 54, 1245–1265. [CrossRef]
15. Sterbenz, J.P.; Çetinkaya, E.K.; Hammed, M.A.; Jabbar, A.; Qian, S.; Rohrer, J.P. Evaluation of network resilience, survivability, and disruption tolerance: Analysis, topology generation, simulation, and experimentation. Telecommun. Syst. 2013, 52, 705–736. [CrossRef]
16. Smith, P.; Hutchison, D.; Sterbenz, J.P.; Schöller, M.; Fessi, A.; Karaliopoulos, M.; Plattner, B. Network resilience: A systematic approach. IEEE Commun. Mag. 2011, 49, 88–97. [CrossRef]
17. Deswarte, Y.; Kanoun, K.; Laprie, J.C. Diversity against accidental and deliberate faults. In Proceedings of the Computer Security, Dependability and Assurance: From Needs to Solutions, York, UK and Williamsburg, VA, USA, 7–9 July 1998; pp. 171–181.
18. Just, J.E.; Cornwell, M. Review and analysis of synthetic diversity for breaking monocultures. In Proceedings of the 2004 ACM Workshop on Rapid Malcode, New York, NY, USA, 25–29 October 2004; pp. 23–32.
19. Knight, J.C. Diversity. In Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2011; Volume 6875, pp. 298–312.
20. Larsen, P.; Homescu, A.; Brunthaler, S.; Franz, M. SoK: Automated software diversity. In Proceedings of the IEEE Symposium on Security and Privacy, San Jose, CA, USA, 18–21 May 2014; pp. 276–291.
21. Baudry, B.; Monperrus, M. The multiple facets of software diversity: Recent developments in year 2000 and beyond. ACM Comput. Surv. (CSUR) 2015, 48, 16:1–16:26. [CrossRef]
22. Han, J.; Gao, D.; Deng, R.H. On the effectiveness of software diversity: A systematic study on real-world vulnerabilities. In Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 2009; Volume 5587, pp. 127–146.
23. Garcia, M.; Bessani, A.; Gashi, I.; Neves, N.; Obelheiro, R. OS diversity for intrusion tolerance: Myth or reality? In Proceedings of the IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN), Hong Kong, China, 27–30 June 2011; pp. 383–394.
24. Wired Business Media. Security Week: Internet and Enterprise Security News, Insights and Analysis. Available online: http://www.securityweek.com/ (accessed on 25 August 2018).
25. Michael Horowitz. Router Security. Available online: https://routersecurity.org/ (accessed on 25 August 2018).
26. Rapid7. Rapid7 Exploit Database. Available online: https://www.rapid7.com/db/modules/ (accessed on 25 August 2018).
27. CVE Details. Available online: https://www.cvedetails.com/ (accessed on 25 August 2018).
28. Carnegie Mellon University, Software Engineering Institute. The CERT Division. Available online: https://www.cert.org/ (accessed on 25 August 2018).
29. Zhu, Y.; Huang, X. Node robust algorithm study based on graph theory. In Proceedings of the Eighth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Shanghai, China, 26–28 July 2011; pp. 2300–2303.
30. Caballero, J.; Kampouris, T.; Song, D.; Wang, J. Would diversity really increase the robustness of the routing infrastructure against software defects? In Proceedings of the Network and Distributed System Security Symposium, San Diego, CA, USA, 10–13 February 2008.
31. Newell, A.; Obenshain, D.; Tantillo, T.; Nita-Rotaru, C.; Amir, Y. Increasing network resiliency by optimally assigning diverse variants to routing nodes. *IEEE Trans. Dependable Secure Comput.* 2015, 12, 602–614. [CrossRef]

32. Zhang, Y.; Vin, H.; Alvisi, L.; Lee, W.; Dao, S.K. Heterogeneous networking: A new survivability paradigm. In Proceedings of the Workshop on New Security Paradigms, Cloudcroft, NM, USA, 10–13 September 2001; pp. 33–39.

33. Caesar, M.; Rexford, J. Building bug-tolerant routers with virtualization. In Proceedings of the ACM Workshop on Programmable Routers for Extensible Services of Tomorrow, Seattle, WA, USA, 17–22 August 2008; pp. 51–56.

34. Wang, L.; Singhal, A.; Jajodia, S. Toward measuring network security using attack graphs. In Proceedings of the ACM Workshop on Quality of Protection, Alexandria, VA, USA, 29 October 2007; pp. 49–54.

35. Bopche, G.S.; Mehtre, B.M. Exploiting curse of diversity for improved network security. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), Kochi, India, 10–13 August 2015; pp. 1975–1981.

36. Zhang, M.; Wang, L.; Jajodia, S.; Singhal, A.; Albanese, M. Network diversity: A security metric for evaluating the resilience of networks against zero-day attacks. *IEEE Trans. Inform. Forensics Secur.* 2016, 11, 1071–1086. [CrossRef]

37. Touhiduzzaman, M.; Hahn, A.; Srivastava, A. A diversity-based substation cyber defense strategy utilizing coloring games. *IEEE Trans. Smart Grid* 2018. [CrossRef]

38. Hole, K.J. Diversity reduces the impact of malware. *IEEE Secur. Privacy* 2015, 13, 48–54. [CrossRef]

39. O’Donnell, A.J.; Sethu, H. On achieving software diversity for improved network security using distributed coloring algorithms. In Proceedings of the 11th ACM Conference on Computer and Communications Security, Washington, DC, USA, 25–29 October 2004; pp. 121–131.

40. Huang, C.; Zhu, S.; Erbacher, R. Toward software diversity in heterogeneous networked systems. In Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy, Vienna, Austria, 14–16 July 2014; pp. 114–129.

41. Temizkan, O.; Park, S.; Saydam, C. Software diversity for improved network security: Optimal distribution of software-based shared vulnerabilities. *Inform. Syst. Res.* 2017, 28, 828–849. [CrossRef]

42. Chen, F.Y.; Kataria, G.; Krishnan, R. Correlated failures, diversification, and information security risk management. *MIS Q. Manag. Inform. Syst.* 2011, 35, 397–422. [CrossRef]

43. Barabási, A.L. *Network Science*; Cambridge University Press: Cambridge, UK, 2016.

44. Luce, R.D.; Perry, A.D. A method of matrix analysis of group structure. *Psychometrika* 1949, 14, 95–116. [CrossRef][PubMed]

45. Karp, R.M. Reducibility among combinatorial problems. In *Complexity of Computer Computations*; Springer: Boston, MA, USA, 1972; pp. 85–103.

46. Garey, M.R.; Johnson, D.S. *Computers and Intractability: A Guide to The Theory of NP-Completeness*; W. H. Freeman & Co.: New York, NY, USA, 1990.

47. Pardalos, P.M.; Xue, J. The maximum clique problem. *J. Glob. Optim.* 1994, 4, 301–328. [CrossRef]

48. Eiben, A.E.; Smith, J.E. *Introduction to Evolutionary Computing*; Springer: Berlin/Heidelberg, Germany, 2003; Volume 53.

49. Joines, J.A.; Houck, C.R. On the use of non-stationary penalty functions to solve nonlinear constrained optimization problems with GA’s. In Proceedings of the First IEEE Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, Orlando, FL, USA, 27–29 June 1994; pp. 579–584.

50. Eiben, A.E.; Ruttkay, Z. Self-adaptivity for constraint satisfaction: Learning penalty functions. In Proceedings of the IEEE International Conference on Evolutionary Computation, Nagoya, Japan, 20–22 May 1996; pp. 258–261.

51. Goldberg, D. *Genetic Algorithms in Search, Optimization and Machine Learning*; Addison-Wesley: Boston, MA, USA, 1989.

52. Kazarlis, S.A.; Bakirtzis, A.G.; Petridis, V. A genetic algorithm solution to the unit commitment problem. *IEEE Trans. Power Syst.* 1996, 11, 83–92. [CrossRef]

53. Rai, S.; Agrawal, D.P. *Distributed Computing Network Reliability*; IEEE Computer Society: Los Alamitos, CA, USA, 1990.
54. The University of Adelaide, Australia. The Internet Topology Zoo. Available online: http://www.topology-zoo.org (accessed on 7 June 2017).