Unavailability and Cost Minimization in a Parallel-Series System using Multi-Objective Evolutionary Algorithms

Ferney A. Maldonado-Lopez, Jorge Corchuelo, and Yezid Donoso

Systems and Computer Engineering
Universidad de los Andes
Bogotá D.C., Colombia

Email: {fa.maldonado1897, corchuelo, ydonoso}@uniandes.edu.co

Abstract—Communications networks have become a key component to all aspects of a modern organization. Network service disruption can compromise normal operation, even it can threaten survivability of business or enterprise. Because of business risk generated by communication lost, organizations have to planning a dependable infrastructure and configure an architecture that guarantees survivability. This work is focused on tasks that a system has to deploy for assuring continuity on its communications by redundancy. This paper affords a point of view for modelling a redundant network based on graphs. This is proposal of using multi objective genetic algorithms for figuring out an optimal configuration of a communication network taking into account two particular objectives. First, it is related to cost to implement and maintains redundant devices on a network. Second, the availability measured in terms of probability of damage and mean time to fail.

Keywords—availability; Multi-Objective Evolutionary Algorithms; Parallel-series System; Spare System;

I. INTRODUCTION

Critical infrastructure is a fundamental system needed for keeping in function an organization. Those including: electricity, communications, supply, operations, and so on. The communication is essential and one of the most important subsystem in a modern organization. The reliance on networks became the IT infrastructure on a fundamental asset. Maintaining it on correct function is a guarantee of business operations survivability along time. Nowadays, resilience communication network is an important topic in most enterprises. Of course, it is an interesting thread of research; several scientists have focused their attention to this problem and have proposed different alternatives facing this challenge.

Whereas several researches have studied topics related to resilience and survivability on a network; we define resilience as the characteristic of a network that can continue in operation and maintain an acceptable level of Quality of Service, even when it is under adverse conditions. Some failure conditions are associated to external or internal sources. Internal disorders are consequences of failures in a component of the system, unintentional misconfiguration, operational errors and misuse. Whereas external changes causing disruption like: natural disasters, malicious attacks to hardware or software network infrastructure, network’s environment changes like mobility or fulfilment new services, unusual traffic but legitimate through the network caused by peak demand of a service in a short period of time. As a result, there are different causes of disruption or disorder on normal operations of a network.

If it is focused on offering redundancy as alternative of resilience, it is necessary to answer two questions. First, what is the best redundancy configuration for an organization or system? This work center its attention on two major objectives: cost and availability, what must be the network infrastructure that makes a correct balance between those purposes? It may be modelled as a multi objective problem. There are available techniques for solving that kind of problems; however, the evolutionary algorithms are suitable for this situation. This paper shows how this algorithm fits and solves this problem, and illustrates additional considerations relatives to resilience in networking.

This paper is organized as follow; first, we documented previous works, different techniques such as dynamic programming and genetic algorithms too. The third section includes the definition of parallel-series system and the notation of availability and unavailability. The fourth section presents a model for depicting a secure infrastructure network on a model based on a graph. The fifth section shows how genetic algorithms are interesting methods for solving that multi objective problems. The final section exposes how was implemented the algorithm and there are conclusions and results.

II. RELATED WORKS

First to all, it is necessary to create an abstract model of resilience and establish a set of characteristics related to capacity of recovery. Sterbenz et al proposed a method to characterize the resilience [1]. They created a two-dimensional state model with the goal of developing quantifiable metrics to determine the degree of the network’s resilience. They identified fundamental properties that affect performance and resilience of a network. Also, they included density, mobility, channels, node resources, network traffic, delay and queuing. They established a model based on three
levels of network operation: normal, partially degraded and severely degraded; and three levels of service requirements: acceptable, impaired and unacceptable. They represent a network as a state on two-dimensional model, and they showed how a transition between two nodes is caused by an incident that compromises network survivability.

The goal is to try to solve the problem of optimization of security cost function proposed by Olovsson [2] where exists a proportion between cost and security. Increasing the security of the system implies to increase cost, delay of response, system complexity; moreover, also decrease performance. The Fig. 1 depicts the security cost function has an inherent multi-objective optimization problem and was demonstrated that this problem is a NP-hard problem [3]. The goal is to achieve a solution win-win, the best balance between security and cost.

However, a security schema does not have an only factor; it is composed of an arrangement of factors that influences cost and performance. We can model a security system as a graph composed of vertices and edges that represent a function into the system. Other works have implemented methods for finding an optimal solution including a set of nodes or multi objective problem.

Using genetic algorithm for optimization is not new. Coit and Smith developed a model for a specific problem using genetic algorithm, analysing series-parallel systems and determining the optimal design configuration when there are multiple component choices available for each of several subsystems. The problem was to select components and redundancy-levels to optimize the objective function, given system-level constraints on reliability, cost, and/or weight [4].

For securing critical infrastructure of information network, Kivimaa et. al. have developed a hybrid expert system that enables a user to select security measures in a rational way based on the Pareto optimality computation using the dynamic programming for finding points into an optimal Pareto curve. Also, their model is based three characteristics of security: confidence, redundancy and user training [5].

Recently Kirt and Kivimaa implemented a discrete dynamic programming method for finding optimal solutions using genetic algorithms [6]. Their result displays variants of security profiles for each cost level previously established. Furthermore, Xiaoli presents a spare optimization models for systems with components connected in series and parallel. The component availability at any time is obtained through Poisson process theory and the availability of the series-parallel system is written by system reliability analysis method. The objective of the optimization problem is to maximize the availability of the system satisfying the constraint on cost. Genetic algorithms are used to find the optimal amount of spares for each component [7].

III. NETWORK RESILIENCE MODEL

The security model for networking is constituted by a series of parallel systems. Each of parallel subsystems represents an only device and its redundant components. The details of the system are explained ahead.

A. Security Model

1) Parallel-Series System: The Fig. 2 illustrates a parallel-series system. It consists of an independent group of subsystems each one with a maximal number of components $n$. In the Parallel-Series Systems a subsystem is available if at least one of its subcomponents is working properly.

2) Availability and unavailability: Availability is the probability in that a system is operating properly when it is requested for using [8]. The notation used is presented in the Table. I. The availability of a system can be calculate by the equation 1.

$$A = \frac{MBTF}{MBTF + MTTR}$$

Unavailability is defined as the probability in that a system cannot operate correctly when it is requested for use. It
### Table I
THE NOTATION OF AVAILABILITY/UNAVAILABILITY

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>Availability</td>
</tr>
<tr>
<td>U</td>
<td>Unavailability</td>
</tr>
<tr>
<td>MTTR</td>
<td>Meantime to Repair is the total time spent performing all corrective maintenance divided by the number of repairs performed.</td>
</tr>
<tr>
<td>MBTF</td>
<td>Mean Time Between Failures is the average time (hours) estimated between failures.</td>
</tr>
</tbody>
</table>

Opposes availability. The Unavailability of a system can be calculated by the expression 2.

\[ U = (1 - A) \] (2)

3) Availability/Unavailability in a parallel or series system: Calculating availability (or unavailability) of a system is given by its topology network. When the devices in the system are in series the availability is calculated as the expression 3.

\[ A_s = A_1 * A_2 \] (3)

When two devices are in redundancy(parallel) availability is calculated as:

\[ A_s = A_1 + (U_1 * A_2) \]

If both devices have the same availability then

\[
A_s = A + (U * A) \\
= (1 - U) + (U * (1 - U)) \\
= (1 - U) + (U - U^2) \\
= 1 - U^2
\] (4)

Therefore, the unavailability of a system with two identical devices is redundant and can be expressed as \( U^2 \). This can be reduced from \( A_s = 1 - U \).

**B. Model application example**

A parallel-series system, for example, is composed of 5 subsystems designed to carry data out from a source \( s \) to destination \( t \). Each subsystem is designed to remaining itself available as much time as possible because it contains 4 components, which allows a level of redundancy of \( n + 3 \).

A network redundant design was adapted to a parallel-series system model. This adaptation establishes a useful model able to be optimized by a multi-objective algorithm. The example model proposed is illustrated in Fig. 3.

This system was designed using the following principles:

1) The system is composed of 5 subsystems, each of which can operate properly if at least one of its components is operational.

2) The operational status and unavailability values of components in a subsystem are independent of all other components throughout the system. It avoids representing the model as a Markov Chain of events none related one each other.

3) At least one component of a subsystem is connected to one component of the next subsystem.

4) The graph has to get a unique path from the source \( s \) and destination \( t \). Thus \( t \) can only be reached by a unique component of each subsystem.

The Fig. 3 illustrates an example how a redundant network design can be represented by a path of nodes into our graph abstraction. Associated to this representation, it is necessary to establish how the system is affected by cost and unavailability. Table II exhibits an example of connection between redundancy and unavailability; Table III indicates how cost is affected respectively.

**C. Abstract Model**

The purposed model of network is based on a graph and it was adapted to an optimization model. Optimization model
requires definitions of a variable or array of variables, an objective function and restrictions over the system.

1) Variable Definition: First to all, It is needed to have a variable that allows to know when a link is being used on the path. Variable $x(i,j)$ means utilization of link between two nodes. The variable $x(i,j)$ can take binary values.

$$x(i,j) = \begin{cases} 
1 & \text{when the link is used from node } i \text{ to node } j, \\
0 & \text{in other case}.
\end{cases}$$

2) Objective Function: Objective function in this case consist minimizing cost and maximizing availability. The expression can be transformed from a min-max problem to a min-min problem using the unavailability definition. Then optimal function is described as:

$$\min \sum_{i,j} x(i,j)c(i,j) \quad \text{(Cost)}$$

$$\min \prod_{i,j} x(i,j)u(i,j) \quad \text{(Unavailability)}$$

Where $u(i,j)$ and $c(i,j)$ corresponds to unavailability and cost of using the link from node $i$ to node $j$ respectively. Those values are calculated for each subsystem individually.

D. Constraints for source, final and intermediate nodes

Continuing with the model, constrains guarantee to find only one path from source $s$ to destination $t$. This only path will be used as a chromosome solution on the next section. Two constrains apply to this model: first, only one path came out from source node $s$; second, there is only one way to reach the destination node $t$.

$$\sum_{j \in E} x(s,j) = 1$$

Constraints for all intermediate nodes have the goal of limiting to only one input; this input is processed and produce only one output to the next subsystem. The constraint for intermediate nodes is showed in next expression. Note that $k \neq s $ and $k \neq t$.

$$\sum_{i \in E} x(i,k) - \sum_{j \in E} x(k,j) = 0$$

There is a complete model that includes: an objective function composed of two different expressions cost and unavailability, constraints for source, destination and intermediate nodes.

IV. EVOLUTIONARY ALGORITHM

Genetic algorithm (GA) is a stochastic search method based on states. An state or chromosome represents a valid solution. Successor states are generated by combining two parent states or modifying a single one analogue to natural selection [9]. GA initiates a set of states generated randomly called population. Each chromosome is rated by the objective function or fitness function. After, a pair of solutions or individuals are selected and produce an offspring by mixing their genetic information. Finally, each chromosome has a random mutation with independent probability. Therefore, a new population we generated for the next iteration.

There are two kind of evolutionary algorithms, non-elitist and elitist ones. Non-elitist algorithms use whole latter population for the next iteration; also, this procedure allows exploit non-dominated solutions, it means, found Pareto-optimal solutions. Second, elitist algorithm gives an opportunity of preserve the best solutions, or elite solutions, directly to next generation.

A. Strength Pareto Evolutionary Algorithm

Zitzler and Thiele proposed an evolutionary algorithm called SPEA [10]. This algorithm maintains elitism by an external population. This population is a collection of non-dominated solutions. At new generation, the algorithm finds solutions non-dominated and compares with previous external population until segregate a new external population. Thus, the algorithm preserves elitist population.

Clustering is an important component in Multi objective evolutionary algorithm. It has been used widely at solving networks problems related to network optimization [11]. This particular problem was faced from two phases. First, we used a high-level modelling system for mathematical optimization which allows to solve linear, nonlinear, and mixed-integer optimization problems. Second, a SPEA genetic algorithm was implemented. This method have demonstrated to have polynomial computational complexity $O(MN^2)$
getting better results than a problem considered NP-hard [3]. The algorithm specification implemented for experimental results is described in Algorithm 1.

Algorithm 1: SPEA algorithm

V. SIMULATION AND RESULTS

This section contains basic aspects of software simulation. The implementation and the outcomes obtained were developed following the algorithm shown above. Simulator and its modules were developed on Java and the following results were gotten.

A. Implementation

A software of Strength Pareto Evolutionary Algorithm was implemented with the follow components. First, it was necessary to establish the initial population. The software generated nodes randomly and those are set to chromosome. After, the chromosome’s validity is checked, if the chromosome is a solution, may be optimal or not, it is recorded into population.

1) Chromosome: While parallel-series system as defined in section III, a subsystem is represented as a number that means the redundancy degree for that device. The chromosome is an arrangement of integers that stores the redundancy value for each subsystem. Whole network is mapped into a collection of participants nodes, those nodes build a path from source $s$ to destination $t$.

2) Selection: What are the best chromosomes for getting next generation? Before the non-dominated solutions from population are found. The first time this is the initial population. After, the software calculates fitness function and ranks the chromosomes by the strength of its solutions. Subsequently best ranked chromosomes are inserted into a roulette. The best chromosomes have higher probability of being chosen because are copied more times into roulette. Later solutions are randomly selected from the roulette for generating new chromosomes.

3) Clustering: After obtain non-dominated solutions, the elite population size must be filtered. Clustering do this task. The algorithm uses the minimal Euclidean distances ideology, points belong to a group because they have minimal distances between them. Clustering makes as many groups as limit of elite population, later extracts the most representative solution for each set.

4) Crossover: The crossover and mutation are mechanisms for finding new chromosomes from the existing ones. A crossover probability is set for determining whether the crossover take place. It is possible to select a cutting point by random in the chromosome. Two chromosomes are selected and generate offspring by mixing their alleles. The left part of the first parent chromosome is selected and the right part of the second parent chromosome is selected too; then both are combined to produce a new offspring. The Figure 5 shows the offspring process, here is obtained a new redundant network infrastructure configuration.

5) Mutation: The process of mutation consists in a random change at only one allele of the chromosome. Mutation probability is set by a parameter of simulation input, it means, it possible to choose an allele and generate a new value for that allele randomly. This method creates a new chromosome changing only a component. Finally, it is mandatory to verify that novel chromosome corresponds to a feasible solution.

B. Outcomes

After build the model and run the simulator, it was decided to execute a total of sixty iterations. This number was reached after analysing the solution distance between one generation to another. Also, the population size was set as a result of previous experiences with similar space of solution. Moreover, crossover and mutation probabilities are the intention to included heuristic for creating new offspring.
a negotiation with cost and availability benefits. This type of solution allows to make a robust outline of a redundancy infrastructure in little time. The advantages are the ability of take decisions and evaluate several scenarios “what if ”.

This work could spread to include more decision variables. Also, future works could consider systems that have increasing or decreasing failure rates, planning scenarios for multiple or intermittent failures. This work has open perspectives for tackling other problems associated to pandemic on networks and bio-inspired mechanisms for facing them like autonomic networks, self-organization and dynamic environments.

REFERENCES


