UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA INDUSTRIAL

DESIGNING RESILIENT POWER NETWORKS AGAINST NATURAL HAZARDS

TESIS PARA OPTAR AL GRADO DE
MAGÍSTER EN GESTIÓN DE OPERACIONES

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL INDUSTRIAL

TOMÁS IGNACIO LAGOS GONZÁLEZ

PROFESOR GUÍA:
FERNANDO ORDÓÑEZ PIZARRO

MIEMBROS DE LA COMISIÓN:
RODRIGO MORENO VIEYRA
DENIS SAURE VALENZUELA
ALEJANDRO NAVARRO ESPINOSA

Este trabajo ha sido parcialmente financiado por UK Research Council y CONICYT por medio del Fondo Newton-Picarte

SANTIAGO DE CHILE
2017

THESIS ABSTRACT FOR THE
DEGREE OF MAGÍSTER EN GESTIÓN DE OPERACIONES
AUTHOR: TOMÁS IGNACIO LAGOS GONZÁLEZ
YEAR: 2017
ADVISOR: FERNANDO ORDÓÑEZ PIZARRO

DESIGNING RESILIENT POWER NETWORKS AGAINST NATURAL HAZARDS

Resiliency of power systems is recently being explored in the literature, its main concern is to provide reliability of the network under high impact low probability (HILP) scenario events. The main contributions of this work are: (1) Provide a novel framework that supports strategic decision making to maximize the resilience of the electricity grid system against natural hazards (the first to the best of our knowledge), in particular earthquakes. (2) Provide an alternative to economic-driven planning that can be contrast for the decision of adding new generation capacity and new lines. (3) Present a Discrete Optimization via Simulation (DOvS) approach that addresses problems having two-steps uncertainty. Our preliminary computational results show that it obtains more robust solutions for this particular problem. We use Industrial Strength COMPASS algorithm to tackle this discrete decision problem, where the measure of resilience corresponds to the expected energy not supplied (EENS). The EENS assessment is undertaken through a simulator that quantifies the impacts of natural hazards on unsupplied demand and contains historical earthquake data, fragility curves of the network components and an operational model of the electricity network (i.e. unit commitment model). Through a case study we demonstrate the applicability of our method, its main features, and ultimately how network planners can design more resilient power networks against earthquakes.

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE MAGÍSTER EN GESTIÓN DE OPERACIONES
POR: TOMÁS IGNACIO LAGOS GONZÁLEZ
FECHA: 2017
PROF. GUÍA: FERNANDO ORDÓÑEZ PIZARRO

DESIGNING RESILIENT POWER NETWORKS AGAINST NATURAL HAZARDS

Resiliencia en sistemas de potencia se está estudiando recientemente en la literatura, su principal preocupación es proporcionar la viabilidad de la red en caso de eventos de alto impacto y baja probabilidad (HILP). Las principales contribuciones de este trabajo son: (1) Proporcionar un marco novedoso que apoye la toma de decisiones estratégicas para maximizar la resiliencia del sistema eléctrico contra la amenaza de desastres naturales (el primero de acuerdo a la investigación realizada), en particular terremotos. (2) Proporcionar una alternativa a la planificación impulsada por incentivos económicos, que puede ser costrastada para decisiones de agregar nueva capacidad de generación y nuevas líneas. (3) Presentar un enfoque de optimización discreta vía simulación (DOvS) que aborda problemas que tienen incertidumbre en dos etapas. Los resultados computacionales preliminares muestran que se obtienen soluciones más robustas para este problema en particular. Se utiliza el algoritmo Industrial Strength COMPASS para abordar este problema de decisión discreto, donde la medida de resiliencia corresponde a la energía no suministrada esperada (EENS). La evaluación de la EENS se lleva a cabo a través de un simulador que cuantifica los impactos de los desastres naturales en la demanda y que contiene datos históricos sobre terremotos, curvas de fragilidad de los componentes de la red y un modelo operacional de la red eléctrica. A través de un caso de estudio, se demuestra la aplicabilidad de este método, sus principales características y, en última instancia, cómo un planificador de la red puede diseñar sistemas de potencia más resistentes frente a terremotos.

A mi hermano, mi padre y mi queridisima madre,
su apoyo me permitió perseguir mi pasión.
A mis amigos más cercanos, que el tiempo no los borra
, a los Tacas, al CEIN y a ĐePeist.
La Salita remembers .

Acknowledgement

This study would not be possible without the support of the UK research council and CONICYT through the grant Newton-Picarte. The project aims to cope for disaster management and resiliency on power systems. It started to develop research on 2016 with the initiative of University of Manchester, Catholic University of Chile and University of Chile. In addition, I would like to acknowledge the work of Rafael Sacaan and Alejandro Navarro in this project, that together, we carried the research and the programming work of these results. Thanks to the research team, specially to the advisors Rodrigo Moreno, Fernando Ordóñez and Hugh Rudnick. I would like to acknowledge the help of Denis Saure, who recommended take a look to reference [14]. "Powered@NLHPC: This research was partially supported by the supercomputing infrastructure of the NLHPC (ECM-02)".

Agradecimientos

Este estudio no sería posible sin el apoyo del UK research council y CONICYT a través del fondo Newton-Picarte. El proyecto tiene como objetivo hacer frente a la gestión de desastres y la resiliencia en los sistemas de potencia. Comenzó a desarrollar investigación en 2016 con la iniciativa de la Universidad de Manchester, la Universidad Católica de Chile y la Universidad de Chile. Además, quisiera agradecer el trabajo de Rafael Sacaan y Alejandro Navarro en este proyecto, que juntos, llevamos a cabo la investigación y el trabajo de programación de estos resultados. Gracias al equipo de investigación, especialmente a los guías del proyecto Rodrigo Moreno, Fernando Ordóñez y Hugh Rudnick. Quisiera agradecer la ayuda de Denis Saure, quien recomendó echar un vistazo a la referencia [14]. "Powered@NLHPC: Esta investigación fue parcialmente apoyada por la infraestructura de supercómputo del NLHPC (ECM-02)".

Contents

Chapter 1

Introduction

We refer to Critical infrastructure (CI) as the infrastructure that provides critical services to the population. The interruption of CI would have a serious adverse effect on society, markets and economy, as a whole, or on a large proportion of the population, and would consequently motivate immediate reinstatement. Like all infrastructure, the CI is vulnerable to natural hazards, such as floods, ice and windstorms, hurricanes, tsunamis, earthquakes and other high impact and low probability events (HILP). The electric power system is particularly important CI, as many other CI depend on it, as shown in Figure 1.1. For example the telecommunications systems may shut down due to the lack of power supply, making the authority less responsive to the emergencies.

In material science, resilience or resiliency is the ability of a material to absorb energy when it is deformed elastically, and release that energy upon unloading. In the context of power systems, as defined in [6], resilience is the capacity of an power system to tolerate disturbances and to continue to deliver affordable energy services to consumers. Also see [26] for a broad discussion on the definition of resilience in power systems. A resilient power system can speedily recover from shocks and can provide alternative means for satisfying energy service needs, in the event of extreme situations. Recent work has focused on building a more resilient system to HILP disasters, see [26]. We provide a framework to incorporate the idea of resiliency in the electric grid expansion planning.

Large power systems are complex dynamic entities, with increasing complexity due to the large adoption of intermittent generation, the additional requirements of reliability, and the need of a more resilient system to face HILP events. Hence, the development of detailed models to mimic real power system operation is complicated, and its incorporation in optimization models is challenging. In fact, we need to balance the accuracy of the model representing the operation of the power system and the flexibility to make optimal design decisions. Thus, to consider a mathematical programming model to make optimal power system design decisions, some simplification of the operation modeling must be made, such as:

- reduction in the number of buses or in the number of constraints in the unit commitment model (UC),

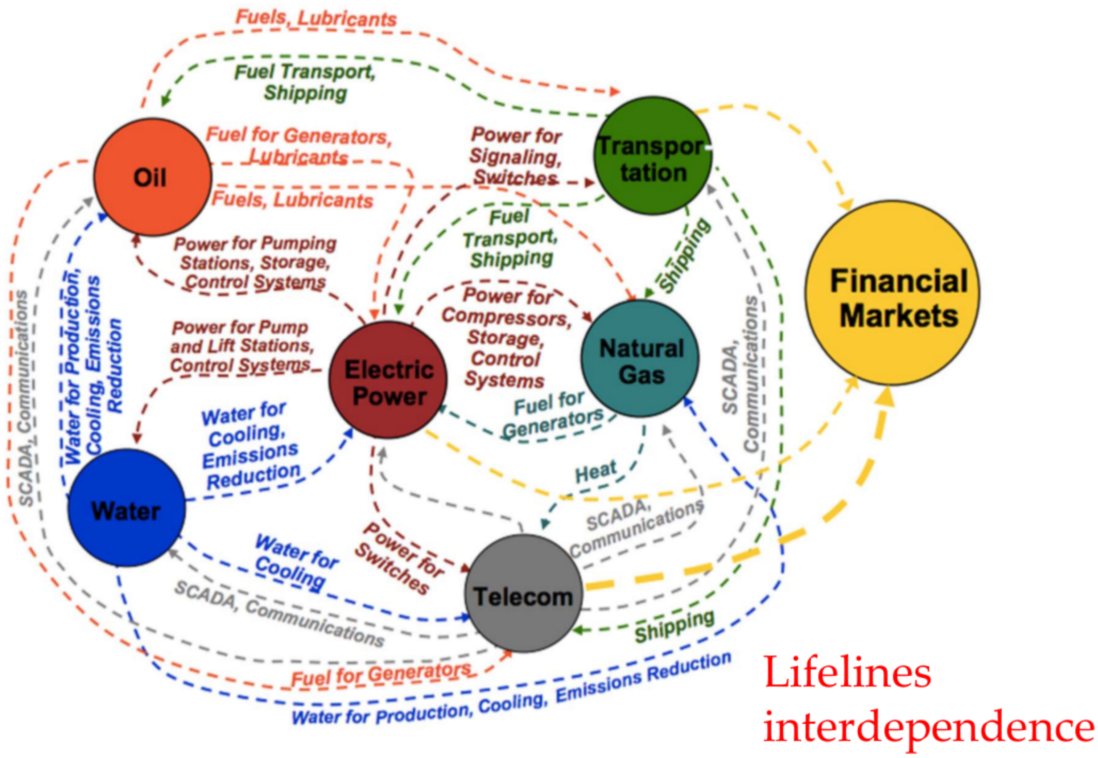

Figure 1.1: lifelines dependency, obtained from [32] and [30].

- simplification of the chronological demand behavior, by aggregating longer time intervals in the time period resolution of the UC [22],
- or simplification of the number scenarios over which to optimize, etc.

On the other hand, if power system operation is modeled with detail, then the resulting model will not have properties that would allow for its optimal solution, and some heuristic methodologies must be used to find good solutions.

We present two frameworks to improve the resilience of the electric grid system against natural hazards. More specifically, we compare models that evaluate and maximize the resilience of the electric power system according to the energy not supplied (ENS), using as decision variables the change of topology (such as building new lines, adding generation capacity, anchoring the substation to make it more resistant to damage). In the first framework, here called the resiliency framework, we use a UC model to represent the behavior of the operator of the system under regular conditions. Using a probability distribution to sample a natural hazard effect, more specifically, an earthquake and damage scenario. A second UC models what the system operator would do in an adverse situation, when it is possible to have some unsupplied demand due to infeasibility.

The second framework we present, here called the reliability framework, uses a model that optimizes once for every period (DC-OPF model). The model binds initial conditions on the variables according to the state of the system given by the solution of the previous period. These DC-OPFs follow the plan of the solution of a UC model. The UC model considered assumes no failures at the initial period, then some failures that were not accounted, occur. Is possible that due to the non-accounted failures, the squence of DC-OPF models deviates from the UC solution. Each period of the DC-OPF model yields an amount of ENS. The ENS sum through all the periods yields the scenario output performance given by the topology of the system. The Optimization via Simulation (OvS) heuristic treats this output as a “black box” function (also called oracle) (see [14] for a brief introduction in OvS).

For the resiliency framework, two different approaches to solve the problem are shown (see Figure 1.2). In one, some of the uncertainty is fixed at the beginning of the optimization procedure. Then the approach is to solve the problem conditioned to that subset of scenarios and repeats fixing the uncertainty and solving the problem many times. The method yields theoretical bounds to the true optimum of the problem. In the second approach, each evaluation samples a scenario leaving the expected value conditioned to all distribution of scenarios. Both the Sample Average Approximation (SAA) and the Full Uncertainty (FU) use the same DOvS heuristic (ISC) to optimize the performance of the simulator evaluation function, see Figure 1.3.

The structure of this dissertation is organized top-down in the sense of the frameworks. A literature review is done in Chapter 2 to show the position of the work with respect to current literature and to motivate the importance of the problem. Chapter 3 shows the Discrete Optimization via Simulation (DOvS) heuristic that is used in this work called Industrial Strength COMPASS (Convergent Optimization via Most Promising Area Stochastic Search). Chapter 4 presents the Unit Commitment (UC) and the DC-OPF mixed integer programs used to model the behavior of the system operator. Chapter 5 focuses on the modeling of the earthquake and the fragility curves used to model the damage on the system components after the shock. The aforementioned chapters present the theoretical framework, the next chapters describe the application. Chapter 6 presents how the framework is combined with the evaluation functions for the ISC algorithm. Chapter 7 presents and analyses the results. Finally the conclusions section presents a discussion and future work to consider.

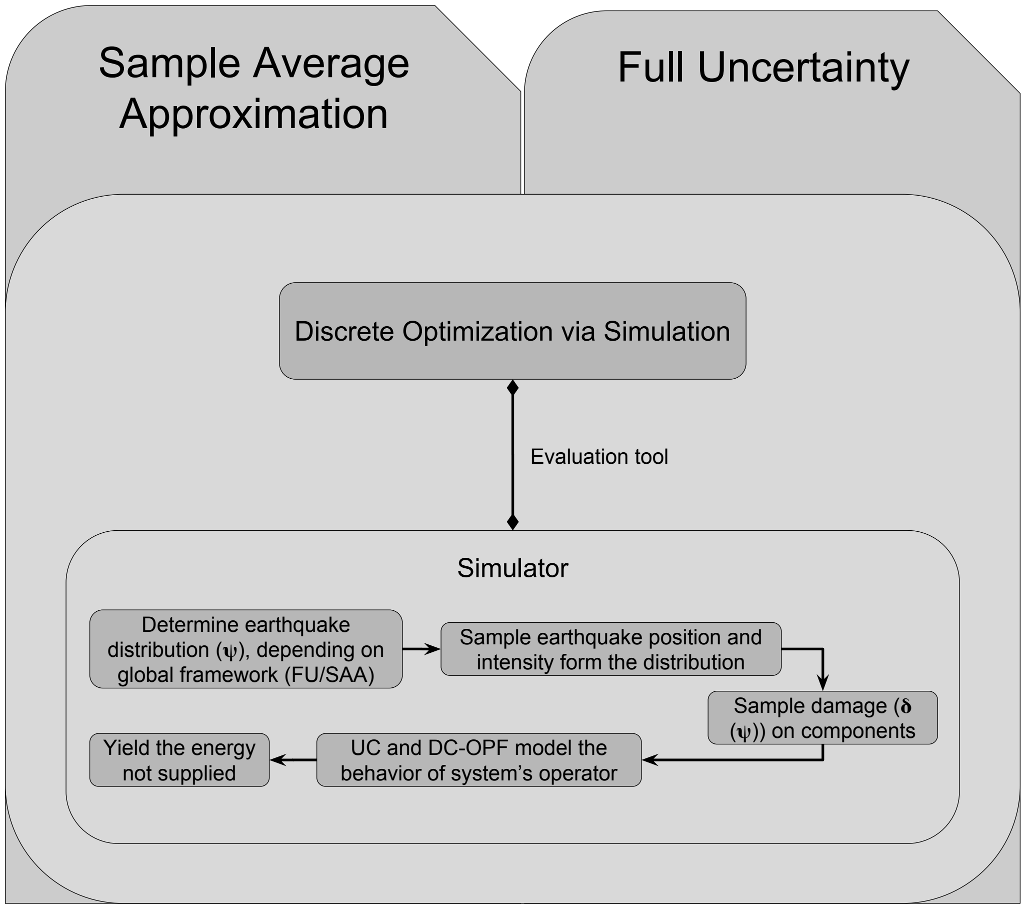

Figure 1.2: Global framework scheme.

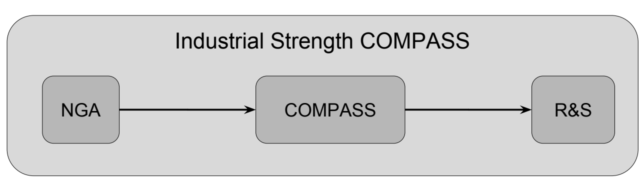

Figure 1.3: Industrial Strength COMPASS main three stages.

Chapter 2

Literature Review

2.1 Resilience and reliability in power systems

The economic development of countries is directly correlated with energy consumption. More developed countries have more energy consumption than emerging economies, but the growth has stalled for the first. In contrast, developing countries is where the investment in capacity is higher. In Chile, due to the occurrence of earthquakes, the study of methods to improve the resiliency of the generation/transmission/distribution system is of great importance (see [2]). In this thesis we focus on how to improve the resiliency of power systems, and develop an optimization via simulation algorithms and a simulation model that tests the preparedness of the system to supply demand taking into consideration stress and high impact low probability (HILP) situation.

Power systems reliability have always been an important subject. In [34] this concept is defined as the probability that a system or component will perform its intended function, under operating conditions, for a specified period of time. In power systems, [1] defines it as the overall ability of the system to supply the demand, meet the constraints of the operation of the system, and provide security to respond to perturbations in the electric system (such as stochasticity of the demand and failures in the system's components). Traditionally, reliability only considers credible failures in the system (e.g. $N-1$ criteria, that is if one of the N components fails, the system is able to satisfy the demand). Reliability does not pursuit protectiveness of HILP failures scenarios, as resilience does. See Table 2.1, it presents the

Reliability	Resiliency
Designed for know failures circumstances	Designed for unforeseen disruptive events
Failures are internal	Failures are external
System cannot reconfigure to avoid failure	System can reconfigure to continue operation

Table 2.1: Summary of main differences between reliability and resiliency, obtain from [23].

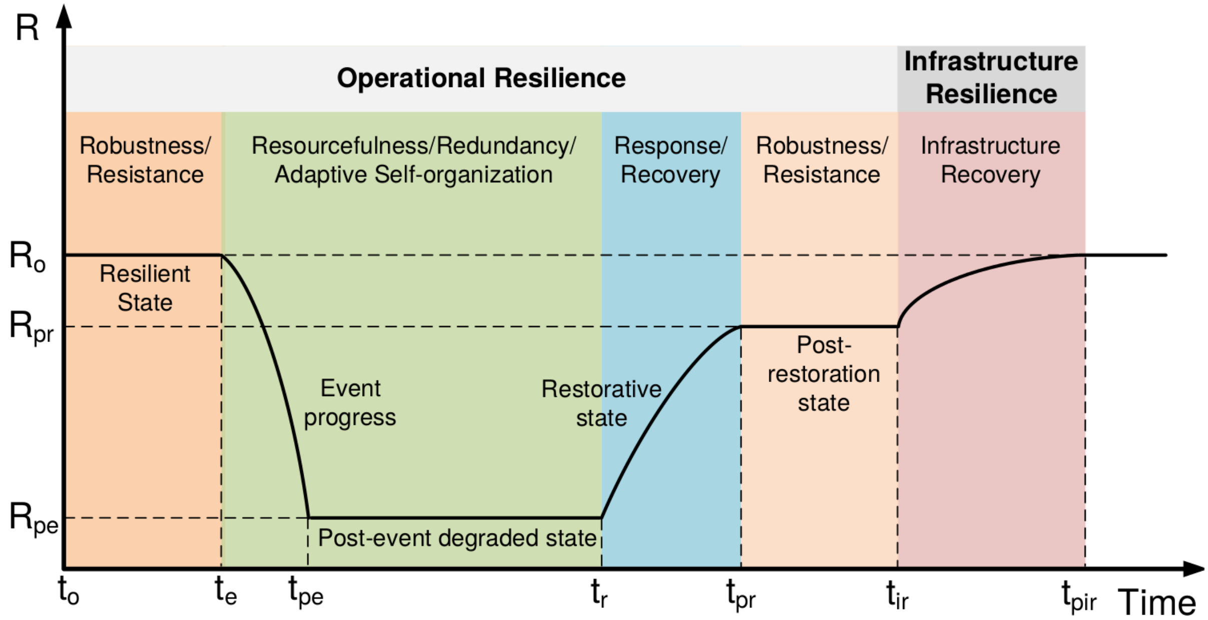

Figure 2.1: Conceptual resilience curve associated to an event ([25])

main differences between reliability and resiliency. A more profound discussion of the concept of resilience in the context of Power Systems is provided in [26]. From which we have the following definition that they obtained from [6]: *"Resilience is the capacity of an energy system to tolerate disturbance and to continue to deliver affordable energy services to consumers. A resilient energy system can speedily recover from shocks and can provide alternative means of satisfying energy service needs in the event of changed external circumstances"*. In a broader context (any critical infraestructure), in [25], the term Resilience refers to the ability of the system to prepare and adapt to changing conditions and withstand and recover rapidly from disruptions, in other words, it refers to the property of being Robust, Redundant and Reliable. Figure 2.1 from Panteli and Mancarella (2015) [25] presents a conceptual resiliency curve. In the y-axis lies the resiliency measure and in the x-axis the measure is time. Between t_0 and t_e the system is prepared to withstand shocking event, at this stage the system should be robust and resistant. Following the event, the system enters a post-event degraded state, where the responsiveness to the emergency helps to minimize the impact of the event $(R_0 - R_{pe})$. Then it enters a restorative state, in which the slope of the curve represents how fast the system recovers to a resilient state. The following stage is the post-restoration state, characterized by having a lower resiliency state and fully operational state. In this stage, the system has a higher fragility and if an second shock disturbs the system it may shut it down completely. The infrastructure may take longer to fully recover. It can get a more resilient and prepared state after/in the infrastructure recovery stage.

We consider energy not supplied as a resiliency measure to be minimized. That is, minimize the area above the curve and below the level R_0, in the graph of Figure 2.1.

This thesis builds on previous work in [10] and [29], and follows the same aim. We also use tools from many different research fields, such as resiliency of electric power systems,

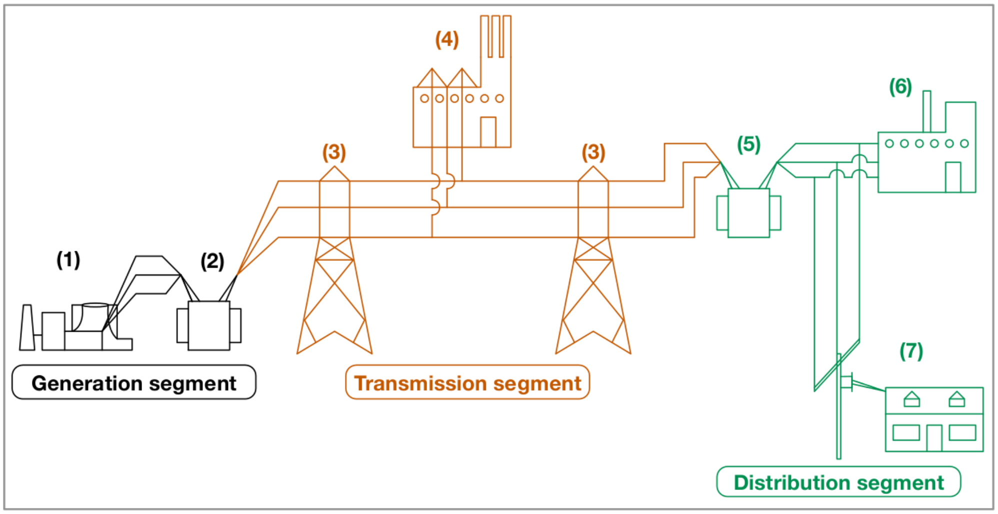

Figure 2.2: Electric power system structure, obtained from [10].

stochastic optimization, optimization via simulation and integer programming. Here we consider a simplified power grid (and represented over a graph) in order to model it with mixed integer linear program (MIP). The power system topology is composed by three main components (see Figure 2.2, obtained from [10]):

- **Generation:** This component includes the generators, connected to the substations. The substations represent a node in the graph.
- **Transmission:** The function of this component is to connect supply and demand, it is constituted by transmission towers, and in the model represent the arcs of the network.
- **Distribution:** It starts at the load substation, and its function is to provide consumers and industries the necessary energy. In the network this component is represented by nodes that demand energy capacity.

Due to the potential failure of the components of the grid, the model considers the use of failure generators. These fictitious generators can satisfy the energy demand at a very high cost, and represent the unmet demand in the system. There is one such failure generator in every demand node. This part of the model provides feasibility of the MIP at any scenario in which part of the demand can not be supplied. The sum of all the failure generators, through all time periods, deliver the energy not supplied in a single simulation. To represent damage to the grid components (the percentage of failure of each component), we use fragility curves developed by Hazus from the Federal Emergency Management Agency (FEMA) [18]. This curves are explained in more detail in chapter 5.

We assume that the operation of the grid aims to minimize cost. The overall cost of not supplying demand is always higher than producing it, if feasible. A full day ahead Unit Commitment (UC) with hourly resolution models this behavior. The UC model used is the one able in Carrión and Arroyo (2006) [4]. The model is explained in more detail in chapter 4. In [9], Mixed Integer Linear Programming was first applied to solve the unit commitment problem. The formulation in [9] was based on the definition of three sets of binary variables to, respectively, model the start-up, shutdown, and on/off states for every unit and every time period. This mixed-integer linear formulation was extended in [3] to the one considered here.

2.2 Optimization via Simulation Algorithms

At a higher level we propose an optimization procedure that considers the output of the simulation as a general function with unknown structure that quantifies the resiliency. The optimization procedure seeks the integer solution (network topology configuration) that minimizes this objective function resilience, the expected energy not supplied (EENS). Two particular Optimization via Simulation (OvS) procedures are introduced:

1. Ranking and Selection-R&S. The feasible set has a finite and small number of solutions and the decision variable may be numerical or categorical. Then all solutions may be simulated and the best one is selected.
2. Discrete Optimization via Simulation (DOvS), considers integer feasible region subset of $\mathbb{Z}^{\mathrm{d}}$ for the decision variables, on which it uses special searching schemes.

A general OvS formulation is the following:

$$\min_{x\in\Theta} \mu(x). \tag{2.1}$$

Where the set Θ has a finite number of feasible solutions. Generally speaking OvS does not require searching schemes over the feasible set, given that one solution that is not visited could be optimal. The R&S algorithms guarantees to find the best solution with a given probability, or with the lowest opportunity cost. DOvS algorithms have implemented searching schemes, that balances the trade-off of exploration versus exploitation, and finally guarantee to converge to a local optimal solution as the simulation effort goes to infinity. Nevertheless, algorithms that have good empirical performance but no convergence guarantees have also been applied to solve OvS problems. Especially when simulation experiments are computationally expensive. R&S procedures have also been used by other OvS optimization algorithms, to improve the efficiency of the optimization process or to make a correct decision at the end of the optimization process. Boesel, Nelson, and Kim (2003) [16] proposed two R&S selection procedure, (clean-up procedures) both implemented in this work. This procedures select the best solution among all solutions evaluated by an OvS algorithm and provides a fixed-width confidence interval for the value of the best solution. Discrete optimization via simulation, DOvS, finds the best solution according to the expected performance of a stochastic system that is represented by a computer simulation model. DOvS algorithms explore more promising solution sets while utilizing some form of randomization to escape local optimal regions. This feature allows to visit all feasible solutions if the computational budget to carry out simulations is large enough [15]. These methodologies move forward from classic heuristic process because they are able to mathematically guarantee correctness of the solution (i.e., the solution found is at least the best visited local optimum, where a local optimum is a solution that is the best among all neighbors) and guarantee the convergence to the global optimum, see [15]. Some important developments have been done in this area since Yan and Mukai (1992) [8] that proposed a DOvS algorithm that delivered true optimal convergence guaranties. Hong and Nelson (2006) [13] proposes a Convergent Optimization via Most Promising Area Stochastic Search (COMPASS) algorithm that is able to converge to a local optimal solution with probability equal to one. That work enabled the development of the Industrial Strength COMPASS (ISC) algorithm, Xu, Nelson and Hong (2009) [15], which gives correctness guarantee. For a brief introduction to OvS research field see Hong and Nelson (2009) [14].

We describe next the methodology by Shapiro *et al* (2009) [33] of Sample Average Approximation (SAA). SAA for discrete optimization is introduced in [19]. The SAA assumes that it is possible to solve to optimality the problem with a fixed set of scenarios. If the assumptions are met, this procedure gives a confidence gap for the true optimum of the problem. Chapter 6 provides a discussion of how this assumptions are met in the experimental settings.

2.3 Sample Average Approximation Estimators

Consider the problem of minimizing the expectation of $F(x,\xi)$:

$$\min_{x\in\Theta}\{f(x) = \mathbb{E}(F(x,\xi)]\}. \tag{2.2}$$

Here Θ represents the set of integer feasible solutions, ξ is a random vector whose probability distribution P is supported on a set $\Xi \subset \mathbb{R}^{\mathrm{d}}$. We assume that the expectation function $f(x)$ is well defined and finite valued $\forall x \in \Theta$.

Suppose that we have a sample of scenario realizations $\xi_1, ..., \xi_n$ of ξ. This leads to the sample average approximation (SAA)

$$\min_{x\in\Theta}\{\hat{f}_n(x) = \frac{1}{n}\sum_{j=1}^{n} F(x,\xi_j)\} \tag{2.3}$$

of the true problem 2.2. Note that each scenario occurs with probability $1/n$ within the SAA function. For a particular realization of the random sample, the corresponding SAA problem is a stochastic programming problem with respective scenarios $\xi_1, ..., \xi_n$ each taken with probability $1/n$.

By the Law of Large Numbers, under the condition that $\hat{f}_n(x)$ converges point-wise almost surely to $f(x)$ as $n \to \infty$ (sample is iid), we have that $\mathbb{E}[\hat{f}_n(x)] = f(x)$, i.e., $\hat{f}_n(x)$ is an unbiased estimator of $f(x)$. We denote by ϑ^* the optimal value of the true problem 2.2 and by ϑ_n the optimal value of the SAA problem 2.3. Suppose that we are given a feasible point $\bar{x} \in \Theta$ as a candidate for an optimal solution of the true problem. Then using the SAA it is possible to estimate the gap

$$gap(\bar{x}) = f(\bar{x}) - \vartheta^*.$$

associated with $\bar{x}$, by solving many optimization problems like problem 2.3.

Consider the optimal value ϑ_{nk} of the k-th SAA problem 2.3. Note that

$$\hat{f}_n^k(x) \geq \vartheta_{nk}, \forall x \in X,\ \forall k, \tag{2.4}$$

where $\hat{f}_n^k(x)$ is the sample average of n scenarios fixed in the k-th SAA problem. One can take expectancy to the inequality above, with respect to k:

$$\mathbb{E}_k[\hat{f}_n^k(x)] \geq \mathbb{E}[\hat{\vartheta}_{nk}],$$

provided that $\mathbb{E}[\hat{f}_n^k(x)] = f(x)\ \forall x \in \Theta$ and that all is valid also for the optimal solution of the true problem, then

$$\vartheta^* \geq \mathbb{E}[\hat{\vartheta}_{nk}]. \tag{2.5}$$

Let $\bar{v}_{nm} = \frac{1}{m}\sum_{k=1}^m \vartheta_{nk}$, which is an unbiased estimator of $\mathbb{E}[\vartheta_{nk}]$. Let $\hat{\sigma}^2_{nm} = \frac{1}{m}[\frac{1}{m-1}\sum_{k=1}^m(\hat{\vartheta}_{nk} - \bar{v}_{nm})^2]$ an estimator of variance of $\bar{v}_{nm}$.

A $100(1-\alpha)\%$ confidence lower bound of the expectation $\mathbb{E}[\hat{\vartheta}_{nk}]$ (and hence of the optimal value of the true problem provided condition 2.5) is

$$L_{nm} = \hat{v}_{nm} - t_{\alpha,m-1}\hat{\sigma}_{nm}.$$

Where $t_{\alpha,m-1}$ is the value of a t-student with a $100 \times \alpha\%$ significance and $m-1$ degrees of freedom. An approximate $100 \times (1-\alpha)\%$ upper bound estimate can be obtained from

$$U_{n'}(\bar{x}) = \hat{f}_{n'}(\bar{x}) + z_\alpha \bar{\sigma}_{n'}(\bar{x}),$$

where $\bar{x}$ is the candidate solution given, n' is the number giving the sample average estimate of $f(\bar{x})$, and

$$\bar{\sigma}^2_{n'}(\bar{x}) = \frac{1}{n'(n'-1)}\sum_{j=1}^{n'}[F(\bar{x},\xi_j) - \hat{f}_{n'}(\bar{x})]^2,$$

a estimate of the variance of $\hat{f}_{n'}(\bar{x})$. The value of n' is supposed to be very large, hence the critical value z_α is from a standard normal distribution rather than a t-student distribution.

Finally, the SAA is used to estimate an upper bound on the value of the gap of the candidate solution $\bar{x}$:

$$\mathbb{E}[\hat{f}_{n'}(\bar{x}) - \bar{v}_{nm}] = f(\bar{x}) - \mathbb{E}[\hat{\vartheta}_n] = gap(\bar{x}) + \vartheta^* - \mathbb{E}[\hat{\vartheta}_n] \geq gap(\bar{x}), \tag{2.6}$$

therefore $\hat{f}_{n'}(\bar{x}) - \bar{v}_{nm}$ is a biased estimator of the $gap(\bar{x})$. And a $100 \times (1-\alpha)\%$ confidence upper bound for the $gap(\bar{x})$ can be obtained from

$$\hat{f}_{n'}(\bar{x}) - \bar{v}_{nm} + z_\alpha\sqrt{\hat{\sigma}^2_{nm} + \bar{\sigma}^2_{n'}(\bar{x})}.$$

There are two types of error in using $\bar{v}_{nm}$ as an estimator of ϑ^*, namely, the bias $\vartheta^* - \mathbb{E}[\bar{\vartheta}_n]$ and variability of $\bar{v}_{nm}$ measured by its variance. Both errors can be reduced by increasing n, and the variance $\hat{\sigma}^2_{nm}$ can be reduced also by increasing m. Note, however, that the computational effort in computing $\bar{v}_{nm}$ is proportional to m, since the corresponding SAA problems should be solved m times, and to the computational time for solving a single SAA problem based on a sample of size n. In cases where computational complexity of SAA problems grows fast with increase of the sample size n, it may be more advantageous to use a larger number of repetitions m. The bias $\vartheta^* - \mathbb{E}[\hat{\vartheta}_n]$ does not depend on m, when the optimization procedure is exact to obtain the optimal solution of problem 2.3. See Proposition 5.6 of [33]. If the sample is iid, then $\mathbb{E}[\hat{\vartheta}_n] \leq \mathbb{E}[\hat{\vartheta}_{n+1}]$ for any $n \in \mathbb{N}$. It follows that the bias $\vartheta^* - \mathbb{E}[\hat{\vartheta}_n]$ decreases monotonically with an increase of the sample size n, see Theorem 5.7 of [33].

Chapter 3

Industrial Strength COMPASS

The Industrial Strength COMPASS (ISC) algorithm considers the problem:

$$\max_{x \in \Theta} G(x) = \mathbb{E}_{\xi}[F(x, \xi)]. \tag{3.1}$$

Were the feasible region is integer valued (e.g. the decision of constructing a new transmission line):

$$\Theta = \left\{ \begin{array}{cc} x: & \\ \sum_{\mathrm{i}=1}^{q} a_{\mathrm{i}j} x_{\mathrm{i}} \leq b_j, & j = 1, .., p \\ 0 \leq l_{\mathrm{i}} \leq x_{\mathrm{i}} \leq u_{\mathrm{i}} \leq \infty, & \mathrm{i} = 1, ..., q \\ l_{\mathrm{i}}, x_{\mathrm{i}}, u_{\mathrm{i}} \in \mathbb{Z}^+ \cup \{0\}, & \mathrm{i} = 1, ..., q, \end{array} \right\} \tag{3.2}$$

x is the vector of decision variables, $\mathbb{Z}^+$ denotes the positive integers. It is assumed that the function $F(x, \xi)$ is unknown, but it can be estimated through realizations of $F(x, \xi_{\mathrm{i}})$ via a simulation experiment.

ISC consists on a three stage procedure. The first stage is based on Genetic Algorithms (called Niching Genetic Algorithm-NGA). It serves as a global search engine to find good neighborhoods of solutions (niches). This sets of solutions are defined by a niche center (a local minimum) surrounded by neighbors with poorer performance. The second stage applies the COMPASS algorithm to each of this local-minimal structures in order to improve locally the solution. It starts with the center of the niche yielded by NGA, and ends when it has found a local minimum with high confidence. The third stage, the clean-up phase, apply a Ranking and Selection (R&S) procedure to select the best of the solutions identified in the local phase with a certain probability. ISC guarantees that all the solutions will be visited infinitely often (this is, if the algorithm is run infinitely many times) because of the sample scheme of the NGA. Although it might be the case that the convergence solution is not a global optimum. As pointed in Chapter 2, COMPASS provides correctness guaranties (to guarantee local optimality of the solution yielded by the procedure). Each of the stages are explained in further detail in each of the following sections.

3.1 NGA

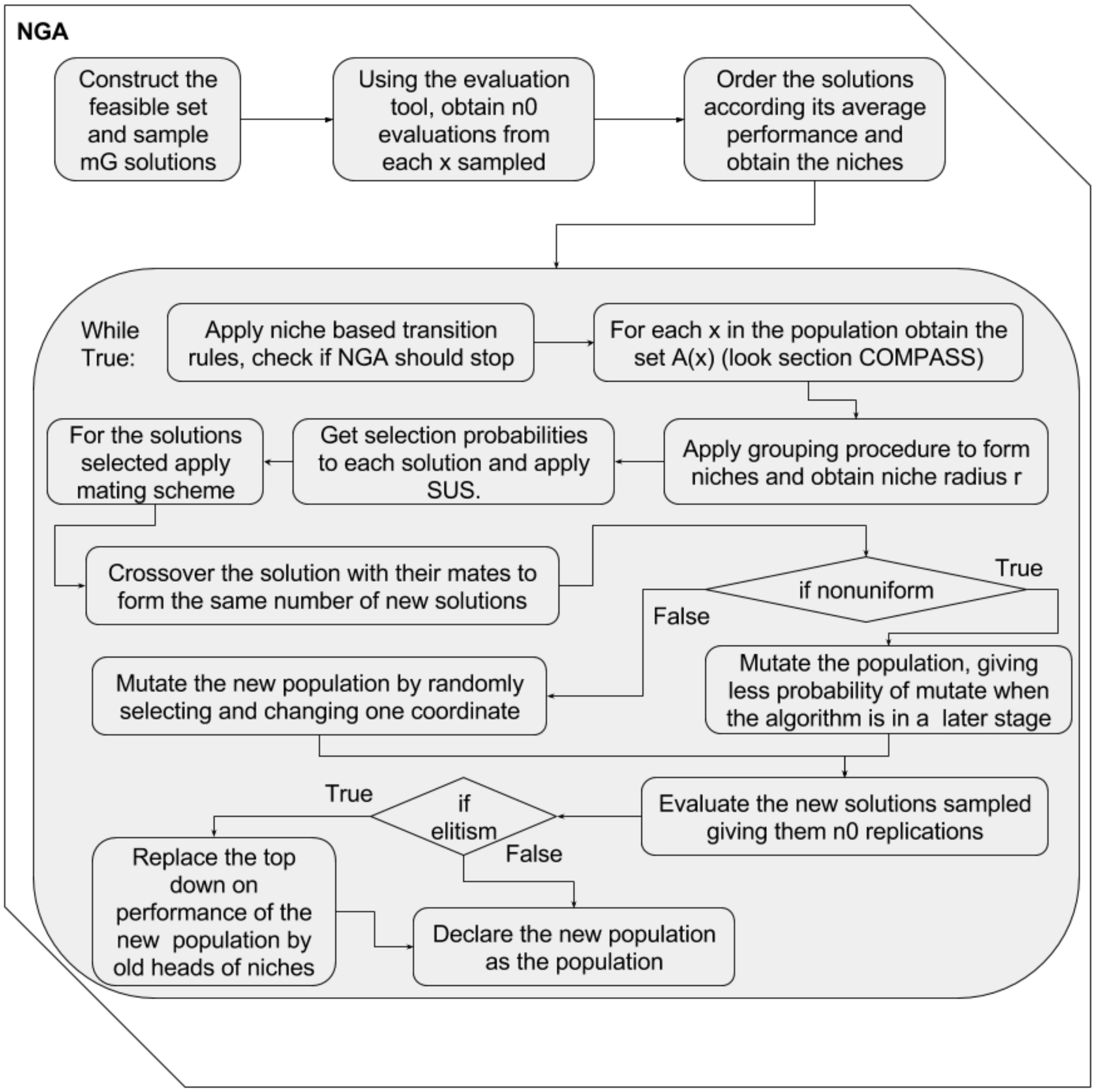

Figure 3.1: Flow-chart of the NGA procedure.

The role of the NGA is to serve as a global search engine. It forms niches that have always at the center a local optimal solution of the niche. The transition rules to the local convergence part of the algorithm could be:

- Niche Rule: If at any time there is only one niche.
- Improvement Rule: If there is no new solutions found in T_G consequent iterations.
- Dominance Rule: If the solutions within one niche dominate all other niches. See more detail on the implementation in 7.5.8.
- Budget rule: If the number of samples is exceeded.

For an example of the NGA stage see Figures 3.2-3.3-3.4, the legends provide the procedure exemplified.

Let the set $\Omega \subseteq \Theta$ and

$$\begin{aligned} \mathscr{C}(x_{\mathrm{i}}) = \{x : x \in \Theta \ and \\ ||x - x_{\mathrm{i}}|| \leq ||x - y||, \ \forall y \in \Omega \setminus \{x_{\mathrm{i}}\}\} \end{aligned} \tag{3.3}$$

and $A(x_{\mathrm{i}}) = \{j : x_j \in \Omega$, such that x_j defines an active constraint in the set $\mathscr{C}(x_{\mathrm{i}})\}$. For a graphical representation of the set $\mathscr{C}(x_{\mathrm{i}})$ just defined see Figure 3.2. A **Fitness Sharing** scheme (see Appendix 7.5.2) is implemented in the algorithm. The basic idea is that if a niche is populated with too many solutions, these solutions should be given less chance to reproduce than they would have in an ordinary GA. Thus allowing solutions in less populated niches to have higher probabilities of being selected to generate new solutions. A **Grouping** procedure (See Algorithm 2) is done to form groups that are similar in their fitness sharing value. For each of these groups is calculated an average probability of selection ($m_j^{(\mathrm{i})} = \frac{1}{N_{\mathrm{i}}}\sum_{j=1}^{N_{\mathrm{i}}} s_j^{(\mathrm{i})}$. Where $s_{\mathrm{i}} = \frac{1}{m_G}(\eta - 2(\eta-1)(\frac{\mathrm{i}-1}{m_G-1})))$ holds as the probability of selection of that particular individual over all current solutions. Given the selection probabilities of each individual (see implementation in Appendix 7.5.3), **Stochastic Universal Sampling** (SUS) (see implementation in Appendix 7.5.4) constructs a roulette wheel where the area for each individual is proportional to its selection probability. Then the roulette wheel is spun once and an individual is selected as a parent. Other individuals are selected by advancing the pointer at a regular spacing until it wraps back to its starting point. The pointer is advanced by a spacing of $2/m_G$. For each solution (x_{i}) selected in the SUS, use a **Mating Restriction** (Appendix 7.5.5) scheme to select its partner. That is, sample m individuals from the population, and select the best one among the ones that are on the same group as x_{i}. If there is not such an individual, select the closest one to x_{i}. A **Crossover** scheme is applied to obtain two new solutions from x_{i} and x_j. By generating $\beta \sim U(0,1)$, and rounding to integers the values of $x'_{\mathrm{i}} = \beta x_{\mathrm{i}} + (1-\beta)x_j$ and $x'_j = \beta x_j + (1-\beta)x_{\mathrm{i}}$. In the case of binary components of vector x, this crossover procedure produces the same sequence of gens that each one of the parents for each child, here it is proposed to combine the components that are one between the parents **(for more detail see the function crossover in the Appendix section 7.5.6)**. If some of these values are infeasible then we kept its parent value. For each new solution apply a **Mutation** scheme (uniform and non-uniform, see [15] and see Appendix 7.5.7) that yields a new solution. This randomly changes the value of a coordinate of the solution. With the new population, after evaluating their performance, there is the option of **elitism**. This is replacing the lower performance ranking by the previous population head of niches.

Algorithm 1 NICHE IDENTIFICATION ON ITERATION k

1: Reindex the solutions in the current population so that $\hat{G}_k(x_1) \leq \hat{G}_k(x_2) \leq .. \leq \hat{G}_k(x_{m_G})$. Let $I = \{1, .., m_G\}$, $L = \emptyset$.
2: **for** i $\in I$ **do**
3:     **if** $\hat{G}_k(x_{\mathrm{i}}) \leq \hat{G}_k(x_j), \ \forall j \in A(x_{\mathrm{i}})$ **then**
4:         $L = L \cup \{\mathrm{i}\}$, $I = I \setminus A(x_{\mathrm{i}})$.
5:     $I = I \setminus \{\mathrm{i}\}$.
6: **return** $r = \min_{\mathrm{i},j \in L, \mathrm{i} \neq j} \frac{1}{2}||x_{\mathrm{i}} - x_j||$ and L.

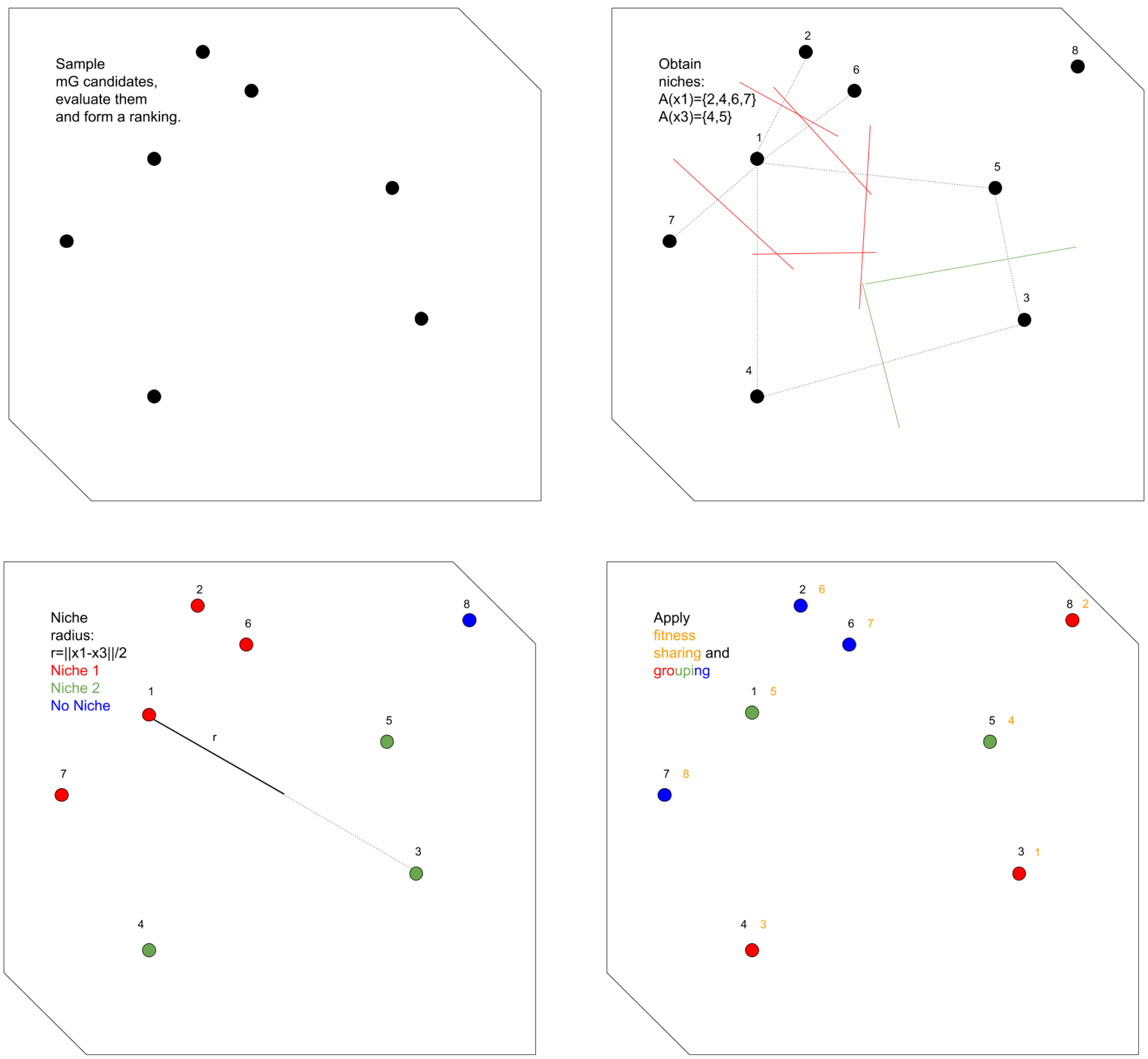

Figure 3.2: Here $m_G = 7$. After simulating the candidates two niches are obtained. The niche radius is half of the minimum between all pairs of niches. The 8-th solution in the ranking is within the no-niche set structure. Note that the grouping procedure may obtain groups that are quite different from the original niches.

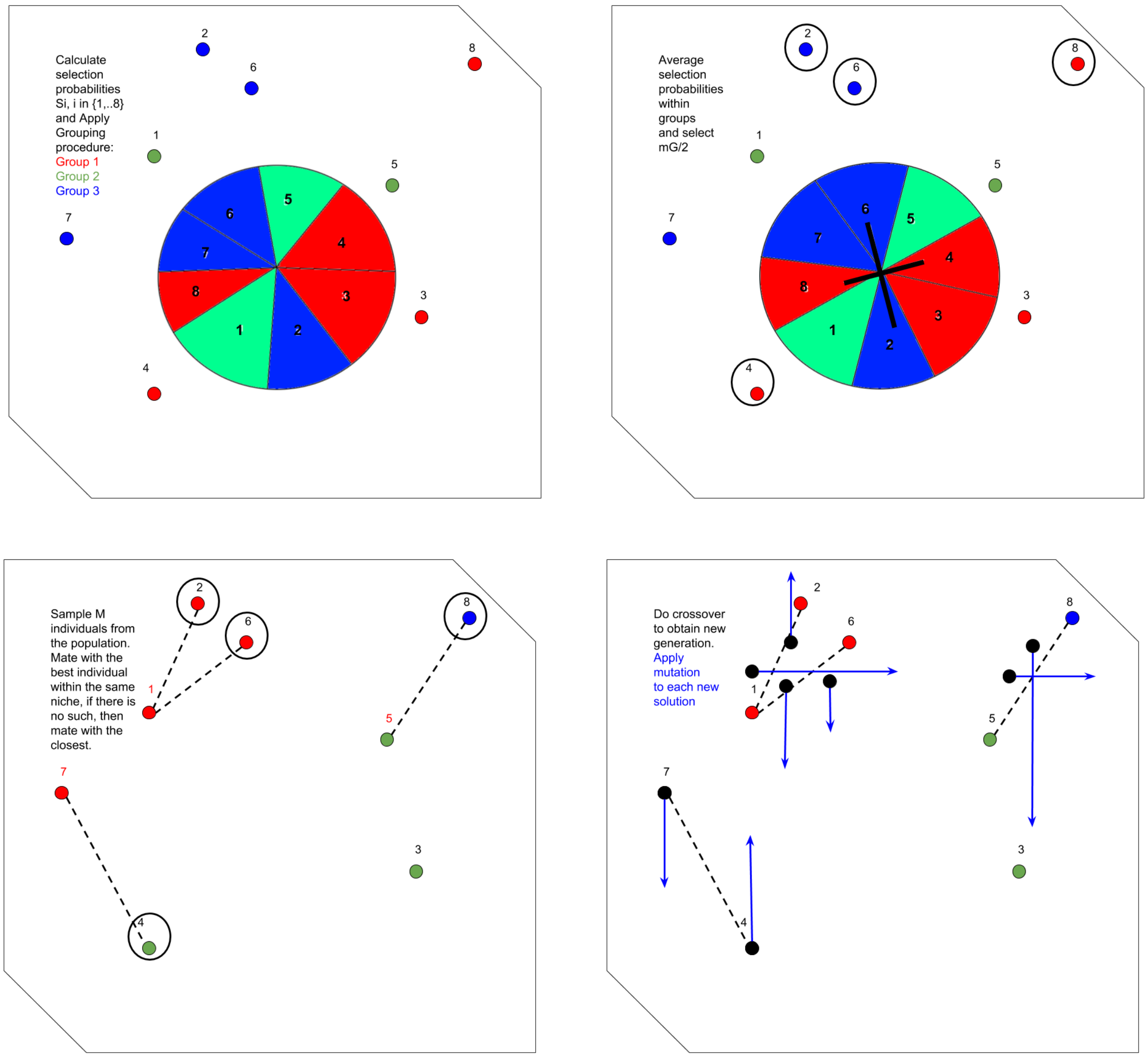

Figure 3.3: Selection probabilities are averaged within niches, so every solution in the same niche has the same probability of selection. SUS builds a roulette wheel and selects $m_G/2$ solutions to be reproduce to obtain next generation. Mates selection is done in order to select with higher probability solutions of better performance. After obtaining the new solutions sampled a mutation step is done.

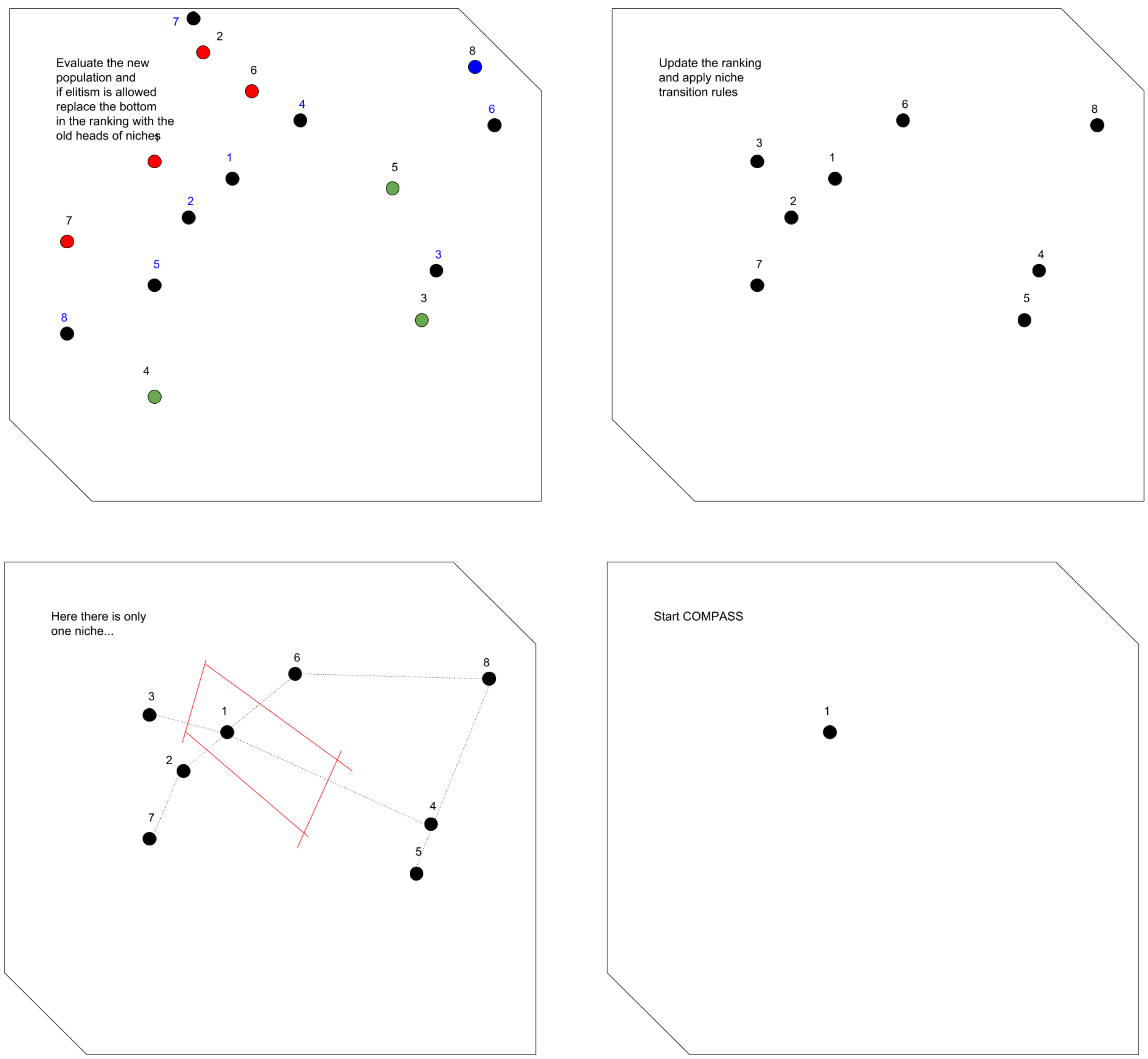

Figure 3.4: New solutions are evaluated and ranked. An optional elitism step can be done to replace the worst ranked candidates by the old local minimums. If no transition rule applies, the iteration should be repeated.

Algorithm 2 Grouping procedure for NGA, See [15] online appendix

Require: The sample average $\hat{G}(x_{\mathrm{i}})$ and the sample variance $S^2(x_{\mathrm{i}})$ are updated with the fitness sharing procedure. A minimum number of groups to be formed g_m. A global search significance α_G and a global search indifference zone δ_G.

1: Given current population $\{\hat{G}(x_{\mathrm{i}}), S^2(x_{\mathrm{i}}), n_{\mathrm{i}}\}_{\mathrm{i}=1,..,m_G}$, let $\bar{n} = \frac{1}{m_G}\sum_{\mathrm{i}=1}^{m_G} n_{\mathrm{i}}$.

2: Sort and reindex so that $\hat{G}_k(x_1) \le \hat{G}_k(x_2) \le .. \le \hat{G}_k(x_{m_G})$.

3: Let

$$S^2 = \frac{1}{m_G}\sum_{\mathrm{i}=1}^{m_G} S^2(x_{\mathrm{i}})$$

and

$$R = \frac{S}{\sqrt{\bar{n}}} Q^{1-\alpha_G}_{m_G, \sum_{\mathrm{i}=1}^{m_G}(n_{\mathrm{i}}-1)}$$

where $Q^{1-\alpha_G}_{m_G, \sum_{\mathrm{i}=1}^{m_G}(n_{\mathrm{i}}-1)}$ is the upper $1-\alpha_G$ quantile of the sudentized range distribution (see Miller (1981) [21]) with m_G degrees of freedom and $\sum_{\mathrm{i}=1}^{m_G}(n_{\mathrm{i}}-1)$ samples.

4: $g = 1,\ \mathrm{i} = 1$.

5: **while** $\mathrm{i} < m_G$ **do**

6: $\quad G_g = \{\mathrm{i}\}$, *bottom* $= \mathrm{i}$, $\mathrm{i} = \mathrm{i} + 1$.

7: $\quad$ **while** $\hat{G}(x_{bottom}) - \hat{G}(x_{\mathrm{i}}) < R$ and $\mathrm{i} < m_G$ **do**

8: $\quad\quad G_g = G_g \cup \{\mathrm{i}\},\ \mathrm{i} = \mathrm{i} + 1$.

9: If the number of groups formed $g < g_m$, let

$$\hat{R}_{\mathrm{i}} = \max_{j \in G_{\mathrm{i}}} \hat{G}(x_j) - \min_{j \in G_{\mathrm{i}}} \hat{G}(x_j),\ \mathrm{i} = 1, .., g$$

$$\hat{m}_{\mathrm{i}} = |G_{\mathrm{i}}|.$$

10: For any group with $|\hat{R}_{\mathrm{i}}| < \delta_G$, do nothing further with G_{i}.

11: **for** groups with $|\hat{R}_{\mathrm{i}}| \ge \delta_G$ **do**

12: $\quad \hat{R} = \max \hat{R}_{\mathrm{i}}$, $\hat{\ell} = arg \max_{\mathrm{i}=1,..,g} \hat{R}_{\mathrm{i}}$.

13: $\quad$ Set

$$\hat{n} = \lceil \frac{Q^2 S^2}{\hat{R}^2} \rceil,$$

where

$$Q = Q^{1-\alpha_G}_{\hat{m}_{\hat{\ell}}, \sum_{j \in G_{\hat{\ell}}}(n_{\mathrm{i}}-1)},$$

$$S^2 = \frac{1}{\hat{m}_{\hat{\ell}}} \sum_{j \in G_{\hat{\ell}}} S_j^2.$$

14: $\quad$ Obtain $\max\{\hat{n} - n_j, 0\}$ observations for $j \in G_{\hat{\ell}}$.

15: $\quad$ Apply fitness sharing.

16: $\quad$ Go to step 1 but with group $\hat{\ell}$ as the population, and update $g_m = g_m - g + 1$.

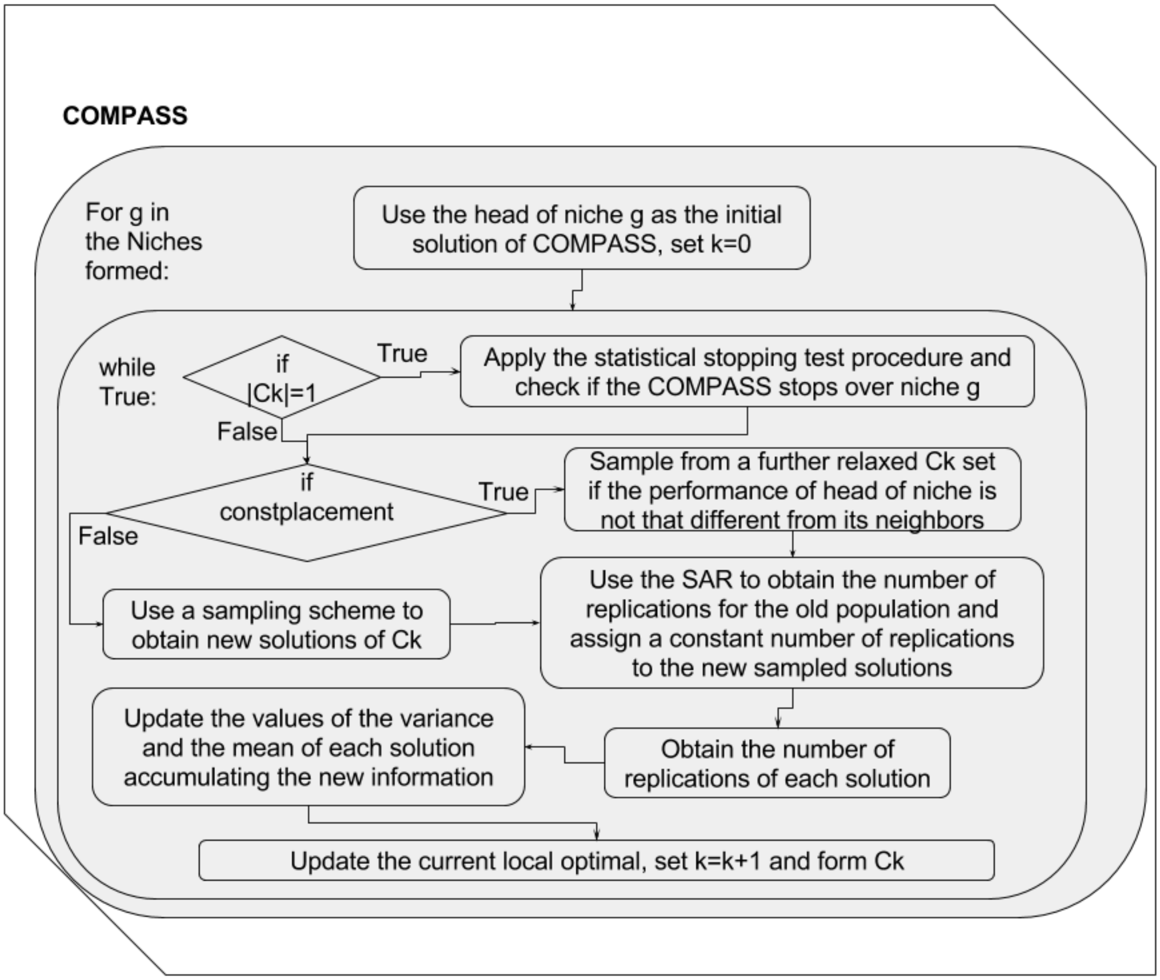

Figure 3.5: Flow-chart COMPASS algorithm

3.2 COMPASS

The next step in the algorithm is to converge locally for each niche. *Convergent optimization via most-promising-area stochastic search* procedure starts with a population of individuals within a niche. Denote the set $\mathscr{V}_k$ as all solutions visited at iteration k and let $\mathscr{C}_k = \{x : x \in \Theta \text{ and } ||x - \hat{x}_k|| \leq ||x - y||, \; \forall y \in \mathscr{V}_k, \; y \neq \hat{x}_k\}$ be the most promising area at iteration k (see Figure 3.2).

Algorithm 3 Sampling Allocation Rule $a_k(x)$

1: At iteration k, let $\Delta N = |A_{k-1}(\hat{x}^*)| - 2$. Let $\hat{\delta}(x) = G(x) - G(\hat{x}^*)$.
2: Allocate two additional replications to $\hat{x}^*$.
3: Let $R = \sum_{x \in A_{k-1}(\hat{x}^*)} \frac{S^2(x)}{\hat{\delta}(x)}$.
4: Set $\Delta N(x) = \frac{S^2(x)\Delta N}{\hat{\delta}(x)R}$. If $\Delta N(x) < 1$, round down to zero, otherwise, round to the nearest integer.

The COMPASS algorithm improves locally the solutions obtained in the NGA stage. The

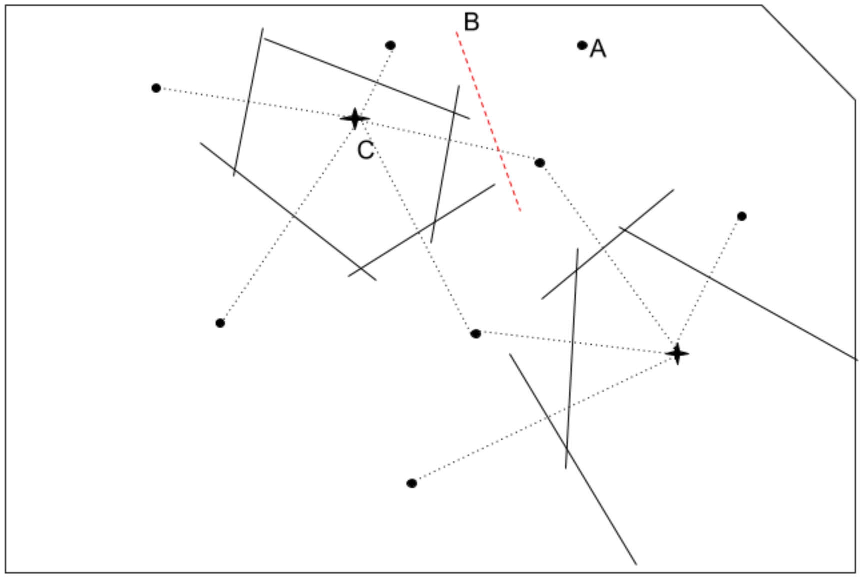

Figure 3.6: Most Promising Area (MPA) around point C, note point A does not define any active constraint, since half-space B it is outside the MPA.

Algorithm 4 COMPASS

Require: Set k=0, find $x_0 \in \Theta$, set $\mathscr{V}_0 = \{x_0\}$. Determine $a_0(x_0)$, and take $a_0(x_0)$ observations from $g(x_0)$. Calculate $\bar{G}_0(x_0)$ and $N_0(x_0) = a_0(x_0)$.

1: Let k=k+1. Sample $x_{k1}, ..., x_{km} \in \mathscr{C}_{k-1}$ uniformly and independently. Let $\mathscr{V}_k = \mathscr{V}_{k-1} \cup \{x_{k1}, ..., x_{km}\}$. For every $x \in \mathscr{V}_k$ take $a_k(x)$ observations and update $N_k(x)$ and $\bar{G}_k(x)$.

2: Let $\hat{x}_k^* = arg\min_{x \in \mathscr{V}_k}\{\bar{G}_k(x)\}$, and construct $\mathscr{C}_k$.

3: If $\mathscr{C}_k = \{\hat{x}_k^*\}$ go to transition rule, else go to line 1

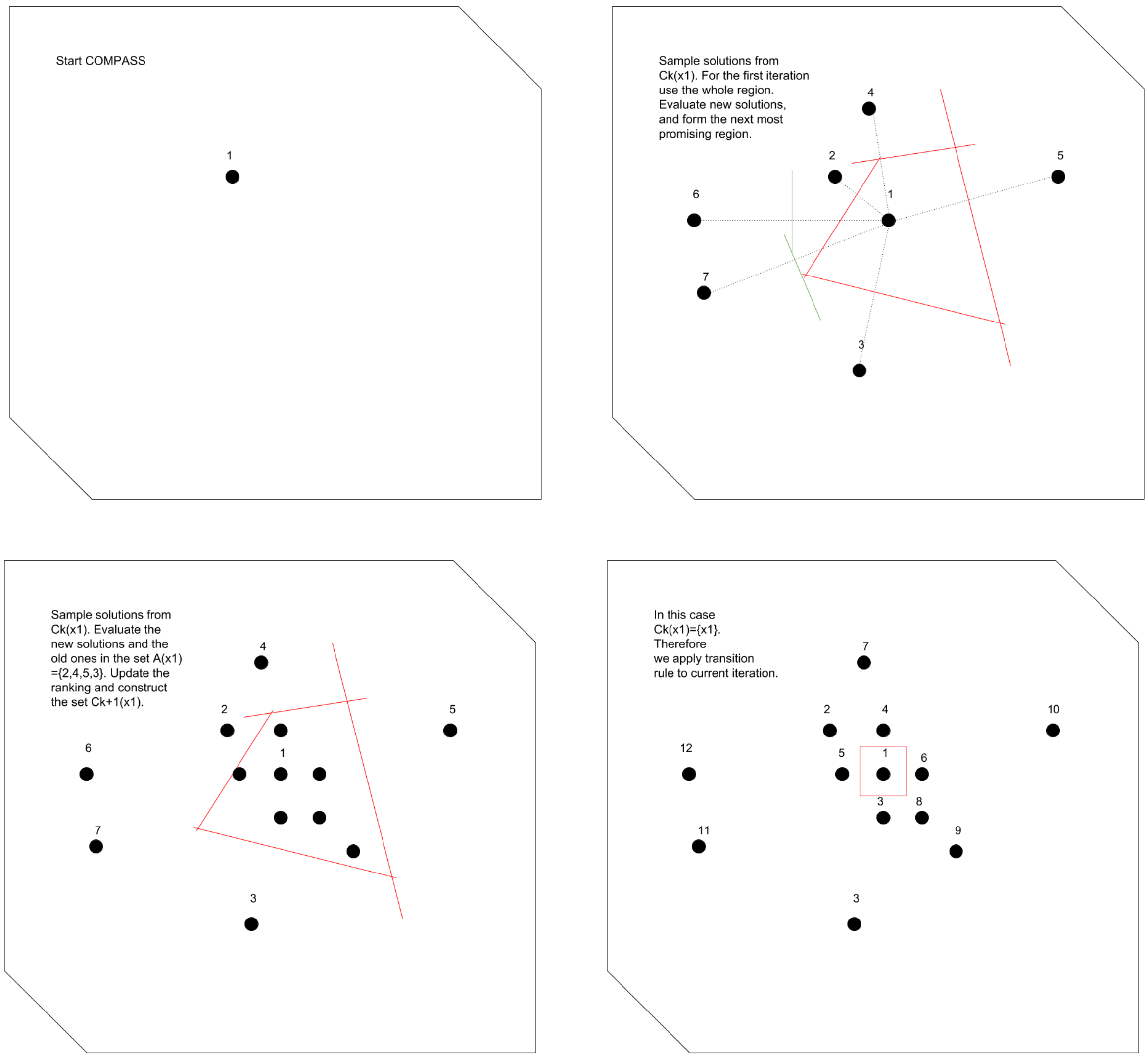

Figure 3.7: For each niche head obtained in NGA, a local improvement is done. The procedure accumulates solutions until the transition rule is met.

Sampling Allocation Rule (SAR Algorithm 3) yields the number of evaluations dedicated to previously visited (and already evaluated in previous iterations) solutions.

Let

$$N(x) = \{y : y \in \Theta, ||x - y|| \leq 1\},$$

given a local solution $\hat{x}^*$ at iteration k of a specific niche, the **Transition Rule** follows the hypothesis test:

$$\begin{array}{ll} H_0 : & g(\hat{x}^*) \leq \min_{y \in N(\hat{x}^*)} g(y) \\ H_1 : & g(\hat{x}^*) > \min_{y \in N(\hat{x}^*)} g(y) \end{array} \tag{3.4}$$

The type I error is set to α_L and the power of the test to be at least $1 - \alpha_L$ if $g(\hat{x}^*) \geq \min_{y \in N(\hat{x}^*)} g(y) + \delta_L$, where δ_L is a tolerance user specified. If the solution passes the test, COMPASS is stopped in that niche. Figure 3.7 presents an example of two iterations of the COMPASS and Algorithm 4 presents COMPASS algorithm.

Transition rule test 3.4 is done by a procedure called comparison with standards. Two different approaches were implemented here, see algorithms 5 and 6 for more detail. The Generic Procedure referenced from [17] differences from the Minimum Switching Sequential Procedure in that the second considers the cost of switching the system being evaluated. This last feature helps, for a given solution, when there is a fixed cost on time at the initialization of a number of evaluations. It also helps when there are marginal gains in time when a system is evaluated many times. Both procedures where adapted to COMPASS framework. If at any moment the local optimal solution being test is discarded to be the best among its neighbors, the procedures stop and uses the current best solution to keep iterating COMPASS from step 1.

3.3 R&S

The objective of this step is to compare the local optimal solutions obtained in the previous step.

Screening. Using whatever data are already available, after local stage COMPASS, form the resulting set of solutions L. Discard any solutions that can be shown to be statistically inferior to others. Let L_C be the surviving solutions.

Selection. Acquire enough additional replications on the solutions in L_C to select the best. Let x_B be the selected solution.

Estimation. With confidence level $1 - \alpha_C$, x_B is the best, or within δ_C of the best. $\pm\delta_C$ is the confidence interval.

Algorithm 7 and 8 show clean-up procedures, both of which were implemented. The first is the one presented in [15] online appendix. Here, they recommend to select Algorithm 7 as the clean-up procedure (which is also call Sort and Iterative Screen in [16]). The second one is presented in [16] as Screen-Restart and Select.

Algorithm 5 GENERIC PROCEDURE [17]

Require: Select confidence level $1-\alpha_L$, an indifference zone parameter δ_L. A family $(x_{\rm i}, \hat{G}(x_{\rm i}), S^2(x_{\rm i}), n_{\rm i})_{{\rm i}\in I=\{1,..,k\}}$ of sampled solutions. Let first stage sample size $r = \max_{{\rm i}\in I}\{n_{\rm i}\}$, c is a non-negative integer constant ($c=1$ is recommended) and η is the solution of the equation

$$\sum_{\ell=1}^{c}(-1)^{\ell+1}(1-\frac{1}{2}\mathcal{I}(\ell=c))(1+\frac{2\eta(2c-\ell)\ell}{c})^{-(r-1)/2}=\beta$$

where $\mathcal{I}$ is the indicator function and β is set to have an overall confidence $1-\alpha_L$, if the systems are independent $\beta = 1-(1-\alpha_L)^{1/k}$.

1: Let each system have r observations, and update the values of $(x_{\rm i}, \hat{G}(x_{\rm i}), S^2(x_{\rm i}), n_{\rm i})_{{\rm i}\in I}$. Compute $S^2_{{\rm i}\ell}$ the sample variance of the difference between systems i and ℓ.

2: Let

$$a_{{\rm i}\ell} = \tfrac{\eta(r-1)S^2_{{\rm i}\ell}}{\delta_{{\rm i}\ell}} \quad and \quad \lambda_{{\rm i}\ell} = \tfrac{\delta_{{\rm i}\ell}}{2c}$$

where

$$\delta_{{\rm i}\ell} = \begin{cases} \delta_L/2 \text{ if i} = 0 \text{ or } \ell = 0 \\ \delta_L \text{ otherwise} \end{cases} .$$

and

$$\bar{G}_{\rm i} = \begin{cases} \hat{G}(x_{\rm i}) - \delta_L/2 \text{ if i} = 0 \\ \hat{G}(x_{\rm i}) \text{ otherwise} \end{cases} ,$$

and

$$W_{{\rm i}\ell} = \max\{0, a_{{\rm i}\ell} - \lambda_{{\rm i}\ell} r\}.$$

3: For each i $< \ell$, i, $\ell \in I$.

4: **if** $\bar{G}(x_{\rm i}) - \bar{G}(x_\ell) \le -W_{{\rm i}\ell}(r)$ **then**

5: Eliminate ℓ from I.

6: **else**

7: **if** $\bar{G}(x_{\rm i}) - \bar{G}(x_\ell) \ge W_{{\rm i}\ell}(r)$ **then**

8: Eliminate i from I.

9: If $|I| = 1$ and the index in I is the same best as in the beginning, then stop and select the system whose index is in I. Otherwise, if the best system from the beginning was eliminated from I, then it was not a local optimum, so stop this procedure and return to COMPASS.

10: Set $r = r+1$ and update the values of $(x_{\rm i}, \hat{G}(x_{\rm i}), S^2(x_{\rm i}), n_{\rm i})_{{\rm i}\in I}$ and compute $S^2_{{\rm i}\ell}$ the sample variance of the difference between systems i $\in I$ and $\ell \in I$. Go back to step 2.

Algorithm 6 Minimum Switching Sequential Procedure [12]

Require: Let L be the index set of the systems being compared, $k = |L|$ and select confidence level $1/k < 1 - \alpha_L < 1$, an indifference zone parameter δ_L and zeroth-stage sample size n_0 (it could be set to $n_0 = \max_{\mathrm{i} \in L}\{n_{\mathrm{i}}\}$). Select λ such that $0 < \lambda < \delta_L$ ($\lambda = \delta/4$ is recommended by Paulson (1964) [27]). Let $(x_{\mathrm{i}}, \hat{G}(x_{\mathrm{i}}), S^2(x_{\mathrm{i}}), n_{\mathrm{i}})_{\mathrm{i} \in L}$ be the family of sampled solutions, and B be the best solution index in I.

1: Let each system have n_0 observations, and update the values of $(x_{\mathrm{i}}, \hat{G}(x_{\mathrm{i}}), S^2(x_{\mathrm{i}}), n_{\mathrm{i}})_{\mathrm{i} \in L}$. Compute $S^2_{\mathrm{i}\ell}$ the sample variance of the difference between systems i and ℓ in L. Let

$$a_{\mathrm{i}j} = \frac{(n_0 - 1)S^2_{\mathrm{i}j}}{4(\delta - \lambda)}\{[1 - (1-\alpha)^{1/(k-1)}]^{-2/(n_0-1)} - 1\}$$

and

$$N_{\mathrm{i}j} = \max\{0, \lceil \frac{a_{\mathrm{i}j}}{\lambda} \rceil - n_0\}.$$

2: For all combinations of $\mathrm{i}, j \in L$, $\mathrm{i} \neq j$, calculate first stage summary statistic

$$Z_{\mathrm{i}j}(n_0) = n_0(\hat{G}(x_{\mathrm{i}}) - \hat{G}(x_j))$$

and the set of systems still in play

$$I = \{\mathrm{i} : Z_{\mathrm{i}j}(n_0) \geq \min\{0, -a_{\mathrm{i}j} + n_0\lambda\},\ \mathrm{i}, j = 1, .., k,\ \mathrm{i} \neq j\}.$$

3: **if** $I = \{B\}$ **then**
4: Stop and select the system whose index is in I.
5: **else if** $B \notin I$ **then**
6: Stop the procedure and go back to COMPASS.
7: **else**
8: Sort the systems based on the zeroth-stage sample means $\hat{G}(x_{\mathrm{i}})$ and let B and S be the index of the best and second best system in I.
9: Let $N_B = \max_{j \in I, j \neq B} N_{Bj}$, be the maximum number of additional observations required for system B to be compared against all other systems in I. Take N_B additional samples from system B. Let $r = 0$.
10: Take one sample from system S, and let $r = r + 1$. Update the value of the combined two-stage summary statistic Z_{BS} and let $W_{BS} = \max\{0, a_{BS} - \lambda(n_0 + r)\}$.
11: **if** $Z_{BS}(n_0 + r) \geq W_{BS}$ **then**
12: Stop this procedure and go back to COMPASS.
13: **else if** $Z_{BS}(n_0 + r) \leq -W_{BS}$ **then**
14: $I = I \setminus \{S\}$ and update S.
15: **else**
16: Go to step 10.
17: **if** $|I| = 1$ **then**
18: Stop and select the system whose index is in I as the best.
19: **else**
20: Let $N_B = \max_{j \in I, j \neq B} N_{Bj}$ and take $\max\{0, N_B - r\}$ from system B and go to step 10.

Algorithm 7 SORT-AND-ITERATIVE-SCREEN PROCEDURE

Require: $\mathcal{L}$ the set of local optimal obtain from local search, constants α_C and δ_C. $n_{\rm i}$ is the number of observations, $\hat{G}(x_{\rm i})$ is the sample mean and $S^2(x_{\rm i})$ is the sample variance.

1: Let

$$\underline{\rm n} = \min_{{\rm i}\in\mathcal{L}} n_{\rm i}$$

$$t_{\rm i} = t_{(1-\alpha_C/2)^{\frac{1}{|\mathcal{L}|-1}}, n_{\rm i}-1}$$

$$h = h(2, (1-\alpha_C/2)^{\frac{1}{|\mathcal{L}|-1}}, \underline{\rm n})$$

$$w_{{\rm i}\ell} = (\frac{t_{\rm i}^2 S^2(x_{\rm i})}{n_{\rm i}} + \frac{t_\ell^2 S^2(x_\ell)}{n_\ell})^{1/2}, \ \forall {\rm i} \neq \ell$$

where $h(2, (1-\alpha_C/2)^{\frac{1}{|\mathcal{L}|-1}}, \underline{\rm n})$ is Rinott's constant ([31]) in the special case of two solutions, confidence level of $(1-\alpha_C/2)^{\frac{1}{|\mathcal{L}|-1}}$ and $\underline{\rm n}$ degrees of freedom (see Boesel et al. (2003) [16]).

2: Let $I = \{{\rm i} : \hat{G}(x_{\rm i}) \leq \hat{G}_\ell + w_{{\rm i}\ell}, \ \forall \ell \neq {\rm i}\}$.

3: $\forall {\rm i} \in I$, compute $n_C(x_{\rm i}) = \max\{n_{\rm i}, \lceil h^2 S^2(x_{\rm i})/\delta_C^2 \rceil\}$. Collect $n_c(x_{\rm i}) - n_{\rm i}$ more observations and update $\hat{G}(x_{\rm i})$.

4: Let $B = arg\min_{{\rm i}\in I} \hat{G}(x_{\rm i})$. Report x_B as the best solution and claim $\hat{G}(x_B) \pm \delta_C$ as the $(1-\alpha_C/2) \times 100\%$ confidence interval for $G(x_B)$.

3.4 ISC Parameters

For a summary of the parameters needed to be set, see Table 3.1.

Algorithm 8 SCREEN-RESTART-AND-SELECT PROCEDURE

Require: $\mathcal{L}$ the set of local optimal obtain from local search, constants α_C and δ_C. n_i is the number of observations, $\hat{G}(x_\mathrm{i})$ is the sample mean and $S^2(x_\mathrm{i})$ is the sample variance. Set $1-\alpha_0 = 1-\alpha_1 = \sqrt{1-\alpha_C}$ (although any decomposition whose product is $1-\alpha_C$ could be used).

1: Let

$$\underline{\mathrm{n}} = \min_{\mathrm{i}\in\mathcal{L}} n_\mathrm{i}$$

$$n_{r\mathrm{i}} = \lceil (\frac{h(|\mathcal{L}|, 1-\alpha_C, n_\mathrm{i})S(x_\mathrm{i})}{\delta_C})^2 \rceil$$

$$t_\mathrm{i} = t_{(1-\alpha_0)^{\frac{1}{|\mathcal{L}|-1}}, n_\mathrm{i}-1}$$

$$w_{\mathrm{i}\ell} = (\frac{t_\mathrm{i}^2 S^2(x_\mathrm{i})}{n_\mathrm{i}} + \frac{t_\ell^2 S^2(x_\ell)}{n_\ell})^{1/2}, \ \forall \mathrm{i} \neq \ell$$

2: Let $\mathcal{H} = \{\mathrm{i} : \hat{G}(x_\mathrm{i}) \leq \hat{G}_\ell + w_{\mathrm{i}\ell}, \ \forall \ell \neq \mathrm{i}\}$, then

$$h = h(2, (1-\alpha_1)^{\frac{1}{|\mathcal{H}|-1}}, \underline{\mathrm{n}})$$

where $h(2, (1-\alpha_1)^{\frac{1}{|\mathcal{H}|-1}}, \underline{\mathrm{n}})$ is Rinott's constant in the special case of two solutions, confidence level of $(1-\alpha_1)^{\frac{1}{|\mathcal{H}|-1}}$ and $\underline{\mathrm{n}}$ degrees of freedom.

3: Restart current sample of solutions, and take $n_{r\mathrm{i}}$ observations to each $\mathrm{i} \in \mathcal{H}$. Restart also the values of $\hat{G}(x_\mathrm{i})$ and $S^2(x_\mathrm{i})$.

4: $\forall \mathrm{i} \in \mathcal{H}$, compute $n_C(x_\mathrm{i}) = \max\{n_\mathrm{i}, \lceil h^2 S^2(x_\mathrm{i})/\delta_C^2 \rceil\}$. Collect $n_c(x_\mathrm{i}) - n_{r\mathrm{i}}$ more observations and update $\hat{G}(x_\mathrm{i})$.

5: Let $B = arg\min_{\mathrm{i}\in\mathcal{H}} \hat{G}(x_\mathrm{i})$. Report x_B as the best solution.

ISC input	Legend
TT	N° of solutions discarded in sampling scheme in the NGA. The bigger the TT the more uniform is the sampling (thus randomizing better, in order to escape local regions in this scheme).
m_G	Cardinality of the set of solutions in every generation of the NGA.
n_0 (NGA)	N° of replications for initializing solutions sampled.
N_0 (COMPASS)	N° of replications for initializing solutions sampled.
T_G	Parameter for the tolerance in the NGA improvement transition rule.
gm	Minimum N° of groups to be form in the grouping procedure.
α_P (NGA)	Dominance transition rule parameter.
α_G (NGA)	Quantile parameter for the grouping procedure.
δ_G (NGA)	Indifference zone parameter for the grouping procedure.
α_L (COMPASS)	Local significance for the transition rule test in COMPASS.
δ_L (COMPASS)	Local indifference zone parameter for the transition rule test in COMPASS.
α_C (R&S)	Clean-up significance for declaring a global optimum.
δ_C (R&S)	Clean-up indifference zone parameter.
η (NGA)	Penalization parameter (between 1 and 2) for selecting rather good than bad solutions in the SUS procedure in the NGA phase.
M (NGA)	N° of mates to be sampled as presented in algorithm mating.
Budget NGA	Maximum N° of evaluations permitted in this phase.
km (COMPASS)	N° of samples allowed to take in the local stage.
nonuniform	Boolean that if true allows non uniform mutation in the NGA stage.
elitism	Boolean that if true allows elitism in the NGA stage.
Constraint placement	Boolean that allows constraint placement in the COMPASS stage.
MSSP	Boolean if set to true determines that COMPASS transition rule is *Minimum Switching Sequential cost*, else transition rule is *Generic procedure*.
Sort and Iterative Screen	Boolean if set to true determines that R&S clean-up procedure is done with *Sort and Iterative Screen procedure*, otherwise *Screen, Restart and Select procedure* is done.
K	Parameter that should be bigger that the N° of epochs of the NGA.
be	Attenuation parameter for the non-uniform mutation.

Table 3.1: ISC input parameters.

Chapter 4

Deterministic Resilient Network Design Problem Formulation

Here we present the MIP models used to represent the behavior of the System Operator under stressed operation. We warn to the reader that this chapter is not aimed to describe the mixed integer models on all their perspective of costs modeling. Cost are important indeed, they allow to model the regular behavior of the operator during the time horizon. This provides a state of the system when the shock arrives. After the shock, the operator considers that the start up, the production, the shut down costs are a couple of orders of magnitude less costly than not supplying the demand to be met. (Here the demand is modeled as a soft constrain rather than as hard constraint.) First, two discrete time Mixed Integer Linear Problems (MIP) are introduced, the Unit Commitment (UC) and the DC Optimal Power Flow (DCOPF). The first model differentiates from the second in that it decides which units are turned on in every period in the time horizon (and assume that a reserve can provide enough protectiveness under regular conditions). The second optimizes each period separately and receives the available units to be turned on as an input. Let us introduce some nomenclature for the parameters (Table 4.1), the variables (Table 4.2) and the sets (Table 4.3) of the models.

4.1 Nomenclature

The models consider an electric grid defined over a graph $G = (V, E)$. Some nodes in the network have generators ($g \in G(j) \subseteq G$, $j \in V$, for $g \in G(j)$ let $j(g) = j$) and others have demand (d_j^t, $j \in V$) on every period of the time horizon ($t \in T$). Every edge of the network ($\mathrm{e} \in E$) represent the transmission system, where the high tension network transport the energy. The conductors that are supported by the towers have some capacity associated with them ($\bar{F}_{\mathrm{i}j}$ on any direction). Every node on the grid should satisfy a power balance equation, where the sum of the demand, minus the generation, plus all the flow that goes out of the node, minus all the flow that comes into the node should be zero. The model also decides which generation units are turned on. If the unit $g \in G$ is turned on in the current period,

Parameters	
A_g	Coefficient of the piece-wise linear production cost function of unit g.
$\hat{A}_{\mathrm{i},j}$	Admitance parameter between nodes $\mathrm{i}, j \in V$.
a_g, b_g, c_g	Coefficients of the quadratic production cost function of unit g.
$B_{\ell g}$	Slope of the ℓ block of the piece-wise linear production cost function of unit g.
$\bar{C}_g$	Shutdown cost of unit g.
d_j^t	Load demand in bus j in period t.
$\mathrm{d}_j^t(\xi)$	Load demand in bus j in period t under scenario ξ.
$\hat{\mathrm{d}}_j$	Peak demand in bus j.
DT_g	Minimum down time of unit g.
K_g^ℓ	Cost of the interval ℓ of the stair-wise startup cost function of unit g.
L_g	Number of periods unit g must be initially offline due to its minimum down time constraint.
ND_g	Number of intervals of the stair-wise startup cost function of unit g.
NL_g	Number of segments of the piece-wise linear production cost function of unit g.
$\underline{\mathrm{P}}_g$	Minimum power output of unit g.
Q_g	Number of periods unit g must be initially online due to its minimum up time constraint.
R^t	Spinning reserve requirement in period t.
RD_g	Ramp-down limit of unit g.
RU_g	Ramp-up limit of unit g.
S_g^0	Number of periods unit g has been offline prior to the first period of the time span (end of period 0).
SD_g	Shutdown ramp limit of unit g.
SU_g	Startup ramp limit of unit g.
T	Number of periods of the time span.
$T_{\ell g}$	Upper limit of block ℓ of the piece-wise linear production cost function of unit g.
t_{offj}^t	Number of periods unit g has been offline prior to the startup in period t.
U_g^0	Number of periods unit g has been online prior to the first period of the time span (end of period 0).
UT_g	Minimum up time of unit g.
V_g^0	Initial commitment state of unit g (1 if it is online, 0 otherwise).
$x = (w, y, z)$	Solution matrix for the topology of the system.
$w_{\mathrm{i}j}$	Equal to one if line from bus i to bus j is constructed, $(\mathrm{i}, j) \in E_c$.
y_j	Integer number between 0 and 10 that represents a $\frac{\hat{\mathrm{d}}_j y_j}{10}$ of fixed distributed capacity installed in load bus $j \in D$.
z_j	Binary variable equal to one if offer bus j's resistance to the earthquake is increased, $j \in O$.

Table 4.1: Model parameters for MIP programs.

Variables	
$\delta_\ell(g,t)$	Power produced in block of the piece-wise linear production cost function of unit g in period t.
θ_{i}^t	Reference angle bus i in period t.
cd_g^t	Shutdown cost of unit g in period t.
cp_j^t	Production cost of unit g in period t.
cu_j^t	Startup cost of unit g in period t.
ENS_{i}^t	Energy not supplied in node i in time period t.
$f_{\mathrm{i}j}^t$	Flow represented in generated capacity units from bar i to j in time period t.
P_g^t	Power output of unit g in period t.
$\overline{P}_g(\xi,t)$	Capacity of unit g given scenario realization ξ and period t.
r_g^t	Maximum available power output of unit g in period t.
v_g^t	Binary variable that is equal to 1 if unit g is online in period t and 0 otherwise.

Table 4.2: Model variables for MIP programs.

Sets	
D	Set of nodes with demand.
$\delta^+(\mathrm{i})$	Set of edges adjacent to node $\mathrm{i} \in V$.
E	Set of edges of the grid network currently available.
E_c	Set of edges that is possible to build a line.
$E^t(x,\xi)$	Set of edges of the grid network given scenario realization ξ and topology x at period t.
G	Set of indexes of the generating units.
$G^t(\xi)$	Available units at period t in scenario ξ.
$G(\mathrm{i})$	Set of indexes of generating units on node $\mathrm{i} \in V$.
T	Set of indexes of the time periods.
V	Set of nodes on the grid network.
O	Set of nodes with offer.
Ξ	Support of scenarios for earthquake and damage realization.

Table 4.3: Model sets for MIP programs.

then the maximum generation increase (or decrease) it could get is determined by a slope. Electric grids are highly complex to model, every node in the system has associated a tension angle (θ). The difference of the angles of two nodes in the system determines the flow in the arc between the two nodes, (this is the parameter that can be controlled by the operator of the system). This is done by an admittance restriction. This restriction would not let any flow to be feasible in the system (as simple flow of commodities), flow must follow decrease in potential energy. This is the main difference between AC OPF and DC OPF, because in reality this restriction is not linear (Kirchhoff law [20]). Approximating it using some admittance parameter allows to simplify the model and solve it with mixed integer linear solvers. Here the random parameters are the availability of a line (lets say $E(\xi, x)$) and the capacity of the generating units ($\bar{P}_g(\xi)$), under scenario ξ. Some previous decision making yielded a topology expansion plan x, from which the model gets the parameters (w, y, z) for the model.

Using the notation above, let any solution (a plan of dispatch), then it should fulfill the following restrictions:

$$(\theta_{\mathrm{i}}^t - \theta_j^t)\hat{A_{\mathrm{i},j}} = f_{\mathrm{i},j}^t,\ \forall (\mathrm{i}, j) \in E^t(x, \xi),\ \forall t \in T, \tag{4.1}$$

the difference of angles should follow Kichcoff's law for every line available in the network,

$$-\bar{F}_{\mathrm{i},j} \leq f_{\mathrm{i},j}^t \leq \bar{F}_{\mathrm{i},j},\ \forall (\mathrm{i}, j) \in E^t(x, \xi),\ \forall t \in T, \tag{4.2}$$

every line available must meet its capacity,

$$\sum_{k \in \delta^+(\mathrm{i})} f_k^t + \sum_{g \in G(\mathrm{i})} P_g^t = 0,\ \forall \mathrm{i} \in O,\ \forall t \in T, \tag{4.3}$$

$$\sum_{k \in \delta^+(\mathrm{i})} f_k^t + \sum_{g \in G(\mathrm{i})} P_g^t = \max\{\mathrm{d}_{\mathrm{i}}^t(\xi) - \frac{y_{\mathrm{i}}}{10}\hat{\mathrm{d}}_{\mathrm{i}}, 0\} + ENS_{\mathrm{i}}^t,\ \forall \mathrm{i} \in D,\ \forall t \in T, \tag{4.4}$$

is the power balance on every node of the network. The demand nodes my have some distributed energy allocated, this decision modifies the parameters in the model.

$$\theta_0^t = 0,\ v_g^t \in \{0, 1\},\ P_g^t \geq 0,\ ENS_{\mathrm{i}}^t \geq 0,\ \forall\ g \in G^t(\xi),\ \forall t \in T. \tag{4.5}$$

The restriction $\theta_0^t = 0$ is a reference angle allocated in the node 0 in the system. Suppose that $\hat{P}_g^{-1}, v_g^{-1}$ is given as an initial state for period 0. The production cost function $\sum_{t \in T, g \in G} cp_g^t$ is modeled using a piece-wise linear (approximating a quadratic cost function).

$$cp_g^t = A_g v_g^t + \sum_{\ell=1}^{NL_g} B_{\mathrm{ell},g} \delta_{\ell,g}^t,\ \forall g \in G^t(\xi),\ \forall t \in T, \tag{4.6}$$

$$P_g^t = \sum_{\ell=1}^{NL_g} \delta_\ell(g,t) + \underline{\mathrm{P}}_g,\ \forall g \in G^t(\xi),\ \forall t \in T, \tag{4.7}$$

$$\delta_1(g,t) \leq T_{1,g} - \underline{\mathrm{P}}_g,\ \forall g \in G^t(\xi),\ \forall t \in T, \tag{4.8}$$

$$\delta_\ell(g,t) \leq T_{\ell,g} - T_{\ell-1,g}\ \forall g \in G^t(\xi),\ \ell \in \{2,..,NL_g\},\ \forall t \in T, \tag{4.9}$$

$$\delta_{NL_g}(g,t) \leq \bar{P}_g - T_{NL_g-1,t},\ \forall g \in G^t(\xi),\ \forall t \in T, \tag{4.10}$$

$$\delta_\ell(g,t) \geq 0,\ \forall g \in G^t(\xi)\ ,\ \ell \in \{1,..,NL_g\},\ \forall t \in T. \tag{4.11}$$

Where $A_g = a_g + b_g\underline{\mathrm{P}}_g + c_g\underline{\mathrm{P}}_g^2$. Generation is also modeled here, it has minimum and maximum capacity generation values per unit. The generation capacity depends on the scenario realization of some damage on the grid. The generator can not exceed its ramp-up and -down capacity per period.

$$\underline{\mathrm{P}}_g^t v_g^t \leq P_g^t \leq \bar{P}_g^t(\xi) v_g^t,\ \forall g \in G^t(\xi),\ \forall t \in T, \tag{4.12}$$

$$P_g^t \leq (\hat{P}_g^{t-1} + RU_g) v_g^t,\ \forall\ g \in G^t(\xi),\ \forall t \in T, \tag{4.13}$$

$$P_g \geq (\hat{P}_g^{t-1} - RD_g) v_g^t,\ \forall\ g \in G^t(\xi),\ \forall t \in T. \tag{4.14}$$

4.2 DC Optimal Power Flow

The AC Optimal Power Flow (AC OPF) problem is a non-linear optimization model that optimizes the flow dispatch in a single period of the energy in a electric grid transmission and generation system. Given the state of the system in the previous period. The DC OPF is a linearization (simplification to allow the model to be solved with Mixed Integer Linear solvers) of the AC-OPF model. The DC OPF MIP is the following:

$$\begin{array}{ccc} (DCOPF_t) & \min_{\theta_{\mathrm{i}}^t, f_{\mathrm{ij}}^t, P_g^t, v_g^t, ENS_{\mathrm{i}}^t} & \{\sum_{g \in G^t(\xi)} cp_g^t + \sum_{\mathrm{i} \in V} \zeta_{\mathrm{i}}^t ENS_{\mathrm{i}}^t\} \\ s.t. & 4.1 - 4.14 & \end{array} \tag{4.15}$$

Here the cost of the energy not supplied ζ_{i}^t is modeled a couple of orders of magnitude bigger than production. The model incorporates the idea that it is more costly not supplying energy than assuming extra generation cost.

4.3 Unit Commitment

The goal of the unit commitment problem is to minimize the total operation cost, which is defined as the sum of the production cost, the startup cost, and the shutdown cost. The production cost is typically expressed as a quadratic function of the power output. While the startup cost could be modeled as a nonlinear (exponential) [35]. The cost production function is modeled as piece-wise linear convex function, while the start-up cost function is modeled stair-wise. The shut-down cost of any unit is modeled as constant.

The unit commitment problem can be formulated as in [35]:

$$\begin{array}{ll} \min & \sum_{t\in T, g\in G^t(\xi)} cp_g^t + cu_g^t + \mathrm{cd}_g^t + \sum_{\mathrm{i}\in V, t\in T} \zeta_{\mathrm{i}}^t ENS_{\mathrm{i}}^t \\ s.t. & \end{array} \tag{4.16}$$

$$4.1 - 4.11,$$

$$cu_g^t \geq K_g^\ell(v_g^t - \sum_{n=1}^{\ell} v_g^{t-n}) \quad \forall g \in G \; \forall t \in T \; \forall \ell \in \{1, .., ND_g\} \tag{4.17}$$

$$\mathrm{cd}_j^t \geq \bar{C}_g(v_g^{t-1} - v_g^t) \quad \forall g \in G, \; t \in T \tag{4.18}$$

$$\mathrm{cd}_g^t, cu_g^t \geq 0 \quad \forall g \in G \; \forall t \in T \tag{4.19}$$

$$\sum_{g\in G} \bar{P}_g^t \geq \sum_{\mathrm{i}\in V} \mathrm{d}_{\mathrm{i}}^t + R_t \quad \forall t \in T \tag{4.20}$$

Last restriction 4.20 models the reserve requirement of the operator. The generation segment takes care of ramping up the units, or shutting down them:

$$P_g^t \leq \hat{P}_g^{t-1} RU_g v_g^{t-1} + SD(v_g^t - \hat{v}_g^{t-1}) + \underline{\mathrm{P}}_g(1 - v_g^t) \quad \forall g \in G, \; \forall t \in T. \tag{4.21}$$

$$P_g^t \leq \bar{P}_g v_g^{t+1} + SD_g(v_g^t - v_g^{t+1}) \quad \forall g \in G, \; \forall t \in T \setminus \{|T|\}. \tag{4.22}$$

$$P_g^{t-1} - P_j^t \leq RD_g v_g^t + SD_g(v_g^{t-1} - v_g^t) + \bar{P}_g(1 - v_g^{t-1}) \quad \forall g \in G, \; \forall t \in T. \tag{4.23}$$

The minimum up-time constraints:

$$\sum_{n=1}^{Q_g}(1-v_g^n)=0 \quad \forall g \in G. \tag{4.24}$$

$$\sum_{n=t}^{t+UT_g-1} v_g^n \geq UT_g(v_g^t - v_g^{t-1}) \quad \forall g \in G,\ \forall t \in \{Q_g+1,..,T-UT_g+1\} \tag{4.25}$$

$$\sum_{n=t}^{|T|}(v_g^n - v_g^t + v_g^{t-1}) \geq 0 \quad \forall g \in G,\ \forall t \in \{T-UT_g+2,..,T\} \tag{4.26}$$

where $Q_g = \min\{|T|, (UT_g - U_g^0)V_g^0\}$. Analogously minimum down-time constraints:

$$\sum_{n=1}^{L_g} v_g^n = 0 \quad \forall g \in G. \tag{4.27}$$

$$\sum_{n=t}^{t+DT_g-1}(1-v_g^n) \geq DT_g(v_g^{t-1} - v_g^t) \quad \forall g \in G,\ \forall t \in \{L_g+1,..,T-DT_g+1\} \tag{4.28}$$

$$\sum_{n=t}^{|T|}(1-v_g^n + v_g^t - v_g^{t-1}) \geq 0 \quad \forall g \in G,\ \forall t \in \{T-DT_g+2,..,T\} \tag{4.29}$$

where $L_g = \min\{|T|, (DT_g - S_g^0)(1-V_g^0)\}$.

The z_j, $j \in O$, parameter is considered in the next Chapter 5, where we introduce the earthquake and damage distribution modeling.

Chapter 5

Earthquakes and Fragility Curves

In this chapter we describe the models of uncertainty that represent the HILP events that affect the power system. The uncertainty is caused by the occurrence of earthquakes and the subsequent failure of network components according to given fragility curves. The scenario realization of parameters that affect the model considered in Chapter 4 are:

- the intensity and positioning earthquake parameter realization (ψ),
- the damage realization on the components ($\delta(\psi)$) modeling.

The realization of the random vector of parameters above determines $\xi = (\psi, \delta(\xi))$, the scenario, which yields the values of $\bar{P}_g^t(\xi)$, the maximum available capacity of generation of a specific generation unit $g \in G$, and the available lines $E(\xi)$.

5.1 Spatial Distribution and Intensity Distribution

The earthquake model provides uncertainty in its spatial-probability distribution and in its intensity distribution. We use a simplified model for the earthquake distribution . The important property of this distribution is that is continuous, providing the simulator infinitely many scenarios from where to sample.

The location of the earthquake is modeled using some fixed points in the map (say ℓ points), then one is picked randomly and then perturbed to move around within a certain radius, the resultant point is the epicenter sampled. Each of these ℓ points are associated with different earthquakes parameters of intensities. For the experiments we consider $\ell = 3$ (north, center and south, see Table 7.22 in the Appendix).

Algorithm 9 presents the pseudo code of the scenario parameters generation in the simulation. Its features are further described in the following sub-sections. The code first samples an epicenter and intensity value, then according to these parameters it calculates the peak ground acceleration (PGA) in every location where there are components of the system. The PGA allows to compute thresholds that determine the probability of system's components to operate at a certain capacity level. The pseudo code returns the level of operation capacity of generating units and a boolean parameter for every line indicating if it fail.

5.2 Fragility Curves and Components Failures

For the lines failure modeling, we assumed that they would fail following some fixed probability (reliability model). It simplifies its behavior by throwing a charged coin that yields it operand state (true-false). In the case of substations and generators failures, their operation capacity will follow some states that occur according a random distribution ruled by fragility curves (resiliency model). This sates could be minor (or slight, 95% operand), moderate (60% operand), extensive (30% operand) or complete (0% operand), the percentage of operation capacity makes sense in this framework since after a seismic event a power plant that is composed of more than one generation unit might have just a portion of the units out of service, the power plant will be able to work at a degraded maximum generation capacity.

As pointed out in [10], the concept of Fragility Curves has origins as a structural reliability concept ([24], [5]). Is a useful tool for a stand-alone analysis of each component. Fragility curves are lognormal distribution functions that describe the probability of reaching, or exceeding, structural and nonstructural damage states, given median estimates obtained from real data of damage of hazards. For the fragility curves parameters see Appendix 7.21, every component that can fail due to the ground acceleration has two pairs of lognormal distribution parameters for every damage state. The first pair are used when variable $z_j = 0$ ($j \in O$, see Table 4.1, this is when bus has regular seismic norm), the second pair are used when $z_j = 1$ (there is strengthened seismic norm).

The fragility curves of the component characterizes the probabilities of being in the different states, are explained in Figure 5.1. In the x-axis the PGA is plotted, and it represents the intensity of the earthquake at the surface of a given point in the map. The PGA parameter is calculated from the following equation

$$PGA(r,h,M) = \frac{\mathrm{e}^{-2.73\log(r+1.58\exp(0.608*M))}\mathrm{e}^{6.36+1.76*M+0.00916*h}}{980.665}$$

where M is an intensity magnitude in the Gutenberg-Richter scale, it also depends on the specific position in the map (x, y). The Hypocenter is $(\mathrm{e}x, \mathrm{e}y, h)$, and $r = \sqrt{(\mathrm{e}x - x)^2 + (\mathrm{e}y - y)^2}$. The result is divided in 980.665 to have the result in units of g ($1g = 980.665[gals]$). The first term in the exponential above is called attenuation. The second is the magnitude at the surface.

Algorithm 9 SCENARIO GENERATION $lfr = 5\%,\ R_{\max} = 150[Km],\ (l_{\mathrm{i}}, u_{\mathrm{i}}) = (8.0, 9.0)$

Require: Fragility curves parameter for lognormal functions for the buses $((\mu_b^{\mathrm{d}s}, (\sigma_b^{\mathrm{d}s})^2))$ and for the generating units $((\mu_g^{\mathrm{d}s}, (\sigma_g^{\mathrm{d}s})^2))$ for every $\mathrm{d}s \in \{complete, extensive\ , moderate, minor\}$ damage state, percentage of available $(\alpha_{\mathrm{d}s})$ capacity for every given damage state $\mathrm{d}s$ for the buses and $(\beta_{\mathrm{d}s})$ for generating units, ℓ points that will act as epicenter points (with coordinates $(x_{\mathrm{i}}, y_{\mathrm{i}})$ see Table 7.22), for each point $\mathrm{i} \in \{1, .., \ell\}$ upper and lower bounds $(l_{\mathrm{i}}, u_{\mathrm{i}})$ for the intensity of an earthquake, and maximum radius $R_{\max}$.

Sample an epicenter mean $\mathrm{i} \in \{1, .., n\}$. // First ψ is generated.
Sample $r \sim unif[0, R_{\max}]$ and $\theta \sim unif[0, 2\pi]$ and let $\mathrm{e} = (r\cos(\theta), r\sin(\theta)) + (x_{\mathrm{i}}, y_{\mathrm{i}})$. Sample an intensity realization $\mathcal{I}_{\mathrm{i}} \sim unif[l_{\mathrm{i}}, u_{\mathrm{i}})$.
for $j \in V$ **do**
    Calculate PGA in the bus in node j, PGA_j, use it to calculate the value of its probability of causing any damage state. Generate a random seed $p \sim unif(0, 1)$, the next for statement is done in the order presented.
    for $\mathrm{d}s \in \{complete, extensive\ , moderate, minor\}$ **do**
        Let $q = \phi(\frac{\log(PGA_j) - mu_b^{\mathrm{d}s}}{\sigma_b^{\mathrm{d}s}})$, where $\phi()$ is the probability distribution function of a standard normal random variable.
        if $p \le q$ **then**
            Let $\gamma_{1j}^t = \alpha_{\mathrm{d}s}\ \forall t \in \{0, .., t_r - 1\}$ and equal 1 for $t \in \{t_r, .., T\}$.
            Break
for $g \in G$ **do**
    Calculate PGA in the bus in node $j(g)$, $PGA_{j(g)}$. Generate a random seed $p \sim unif(0, 1)$, the next for statement is done in the order presented.
    for $\mathrm{d}s \in \{complete, extensive, moderate, minor\}$ **do**
        Let $q = \phi(\frac{\log(PGA_{j(g)}) - mu_g^{\mathrm{d}s}}{\sigma_g^{\mathrm{d}s}})$, where $\phi()$ is the probability distribution function of a standard normal random variable.
        if $p \le q$ **then**
            Let $\gamma_{2g}^t = \beta_{\mathrm{d}s}\ \forall t \in \{0, .., t_r - 1\}$ and equal 1 for $t \in \{t_r, .., T\}$.
            Break
Let $\bar{P}_g^t(\xi) = \bar{P}_g \min\{\gamma_{1j(g)}^t, \gamma_{2g}^t\}$. If some $\bar{P}_g^t(\xi)$ is 0 take them out of the set G to form $G(\xi)^0$. Let $E(x, \xi)^0 = E$.
for t in T **do**
    for $\mathrm{e} \in E$ **do**
        Generate a random seed $p \sim unif(0, 1)$.
        if $p \le lfr$ **then**
            $E(x, \xi)^t = E(x, \xi)^{t-1} \setminus \{\mathrm{e}\}$.
        Add back to the set $E(x, \xi)^t$ lines that met their recovery time.
return $([\bar{P}_g^t(\xi), E(x, \xi)^t]_{g \in G, t \in T}, \xi)$

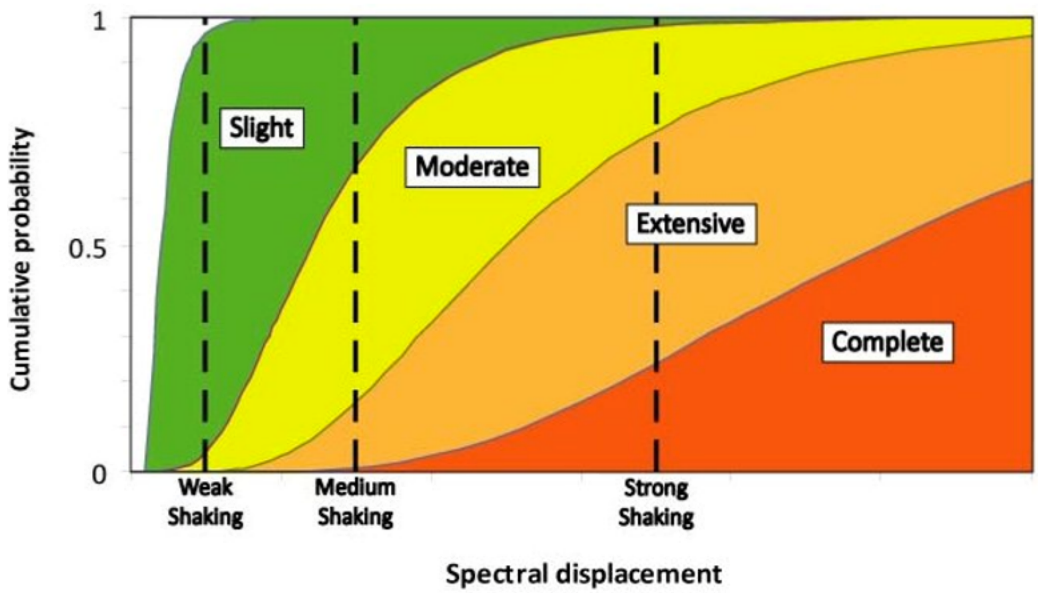

Figure 5.1: Given the PGA in the x-axis, the probability of being in a certain state is the difference between the curve and the curve below it. There is also a no-damage state, whose probability is the difference between 1 and the curve above all the others (minor state curve).

Chapter 6

Combined Framework

Here we present the big picture of how the settings presented in previous chapters are combined, for both the reliability and the resiliency models. We use ISC for trying to find a good feasible solution for these problems instances. In addition, two different procedures are performed and compared. The first estimates the SAA problem 2.3 viewed in chapter 2, fixing some of the uncertainty given by the earthquake distribution (ψ) at the beginning of the optimization (performing many optimizations). The second procedure lets all the uncertainty available (full uncertainty - FU) to be sampled in each simulation. Here, the model to solve is:

$$\begin{array}{cl} \min & \{\mu(x)\} \\ s.t & x \in \Omega \end{array}$$

where the calculation of the function $\mu(x)$ and the set Ω is framework (reliability-resiliency) specific.

6.1 Reliability Model

This model consist in the simulation process of the system operation with inner failures not caused by external shock hazards. The simulation represents how the operator would work under abnormal conditions, that the system should have been prepared to account for, given some components failure rates, within a time horizon. At the beginning of the simulation we obtain a plan by solving the UC dispatch planning model, which assumes no failure of components. We also let the system to protect itself against some uncertainty allowing some reserve allocated in aggregated production capacity (see restriction 4.20 of the UC model). In the course of the operation the system's components that fail are recovered on the following time periods until some recovery time is meet (it could be either deterministic or modeled as a random variable). Provided the strategic planning of the UC, a sequence $\{(DC\text{-}OPF)_t\}_{t\in T}$ of models represents each time period separately. Due to the inner failures of the components the sequence of models may deviate from UC dispatch plan. The DC-OPF sequence is constrained to follow the decision of which units are on and which units are off of the UC solution. The maximum energy available from a unit when is turned on is given by constraint Equation 4.13. In which we let the unit provide energy the same period that is turned on.

Algorithm 10 THE SEQUENTIAL MONTECARLO (SMC) PROCEDURE FOR RELIABILITY MODEL

Require: The topology of the system $x \in \Omega$.
1: Solve the UC given the topology of the system in the scope of the plan horizon T and assuming no failures.
2: Let $ENS = 0$.
3: **for** $t \in T$ **do**
4: Generate a failure scenario for the t-th period.
5: Solve (DC-OPF) using model 4.15.
6: Retreat $\sum_{\mathrm{i} \in V} ENS_{\mathrm{i}}^{t}$ and sum it to ENS term.
7: **return** ENS

In the reliability framework we consider the decision of building new lines:

$$\Omega(budget) = \{x \in \{0,1\}^n | \sum_{\mathrm{i}=1}^{n} x_{\mathrm{i}} \leq budget\}. \tag{6.1}$$

Function $F(x, \xi)$ is described in algorithm 10. ISC is used to find optimal solution for this problem. We do not consider the SAA framework in this optimization procedure, therefore all the uncertainty is available to be sampled on every simulation. The only components allowed to fail are the lines, and both original lines in the system and the ones added by the solution fail with the same rate.

6.2 Resiliency Model

This model approximates the behavior of the system operator under initial state damage conditions caused by an external shock. Here we do not incorporate the reliability point of view of the previous section (this combination is left proposed for further work). A first plan is done by a UC model, assuming no failures (nor inner, nor shocks), and letting some reserve capacity, as in restriction 4.20. The last period constraints can be design to meet some initial set of the state of the system, or to be the initial condition for the first period variables. A demand period is sampled randomly, let it be t_{random}, in which an earthquake occurs. The earthquake carries some damage to the grid, causing it to operate in some percentage of its capacity. Given by the fragility curves model in Chapter 5. The optimization is performed for the following T periods after t_{random}. We assume the system recovers with deterministic recovery times, and a new plan is obtained considering the new availability. The initial conditions are state of the system at time t_{random}. The output of the simulation is the sum of the unsupplied demands through the time periods in the second UC.

The decisions allowed are to buil new lines, to add anchored seismic norm to the buses and/or to add distributed generation capacity to the buses. The feasible solution set is

$$\Omega(budget) = \{(w,y,z) \in \{0,1\}^{p+q} \times \mathbb{Z}^r | \sum_{\mathrm{i}=1}^{p} w_\mathrm{i} + \sum_{j=1}^{q} y_j + \sum_{k=1}^{r} z_k \leq budget,\ 0 \leq z_k \leq 10\ \forall k \in \{1,..,r\}\}, \tag{6.2}$$

where x represent the decision of building new lines, y represent the decision of anchoring buses ($q = |V|$). z is an integer number between 0 and 10 that represents a $z_k \times 10\%$ of distributed generation capacity with respect to the peak demand in node $k \in D$ (where k is such that it represents a demand node $D \subseteq V$). Both the full uncertainty (FU) and the SAA evaluation function are shown in Algorithm 11, the value returned is the sum of the ENS given by the UC model plus the ENS of the demand that cannot connect to the system due to the shock damage in the load bars. The difference between the SAA and the FU is in the earthquake scenario sampling (ψ). For the FU, ψ is sampled from all its support, for each simulation one new sample of ψ is obtained. On the other hand, every SAA evaluation requires simulating the n scenarios fixed $\{psi_1, .., \psi_n\}$ from which we obtain the average of the system performance. This follows the same idea presented in Chapter 2.3, the scenarios have to be sampled using the original probability distribution. In the case of earthquakes, sampling them from the fixed n-sized population with probability $\frac{1}{n}$ imitates the distribution, (scenarios are independent and identically distributed). In the SAA, instead of sampling scenarios with probability $\frac{1}{n}$ we evaluate all of them, the simulator function yields the average through all scenarios. This is, $\sum_{\xi \in \{\xi_1,..,\xi_n\}} \frac{1}{n} F(x,\xi)$ to evaluate a simulation of the performance of x in Algorithm 11.

Algorithm 11 The Sequential Montecarlo (SMC) procedure for Resiliency Model

---

Require: The topology of the system $x \in \Omega$.

1: Solve the UC given the topology of the system in the scope of the plan horizon T.
2: Sample an earthquake scenario given by $\psi \in \Psi(x)$ (the support of the parameters that describe the occurrence of a certain earthquake with certain characteristics, and time period). Then obtain the realization of the damage in the grid $\delta(\psi(x))$.
3: Solve a UC using the new damaged and an on-recovery topology. Let $[ENS_\mathrm{i}^t]_{\mathrm{i} \in V,\, t \in T}$ be the energy not supplied of the optimal solution (or near by optimal).
4: **return** $ENS(x, \xi = (\psi, \delta(\psi))) = \sum_{\mathrm{i} \in V, t \in T} ENS_\mathrm{i}^t + \mathrm{d}_\mathrm{i}^t - \mathrm{d}_\mathrm{i}^t(\xi)$

6.3 Combined Aspects

One may wonder why not fixing before optimization the seeds used to sample the damage on the grid (as we did with the earthquake distribution). In notation, we fix some set of scenarios $\xi_1, .., \xi_n = \delta_1(\psi_1), .., \delta_n(\psi_n)$, and use these scenarios to build the SAA objective function in problem 2.3. The resulting problem is a deterministic model in which we minimize $\frac{1}{n} \sum_{\mathrm{i}=1}^{n} F(x, \xi_\mathrm{i})$ by selecting x. Note that the polyhedral function $F(x, \xi_\mathrm{i})$ is a deterministic MIP with an exact solution. The problem of this approach is that size of the deterministic equivalent model (DEM) grows rapidly with the number of scenarios. Suppose the dimension of the decision variables in problem embedded in $F(x, \xi_\mathrm{i})$ is bounded below by an integer d.

Therefore the DEM is bounded below by $n \times$ d. Now, as shown in next section, average performance of the first stage decisions stabilizes in around 6,000 evaluations, which would mean that we would have to solve a model with $6,000 \times$ d variables to obtain an exact solution.

The SAA setting requires an optimization procedure that gives the true exact optimum of problem 2.3. ISC is not able to provide this guarantee, since is heuristic. The only guaranties that ISC is able to provide are asymptotic (in the sense of running infinitely many times the algorithm). The SAA combined framework inherits this guaranties, therefore equation 2.4 is not granted. The results in the following section show unvalid lower bounds, that do not guarantee to be a lower bound of the true optimum of problem 2.2.

In order to validate the results for the SAA framework, the ISC results are compared with the results obtained using complete enumeration (CE) for a small instance.

After performing $m = 10$ optimizations for the SAA framework, all solutions reported as candidates are further evaluated 6,000 times and sort increasingly in a ranking, whose first solution in the list is selected as candidate solution $\bar{x}$. In order to calculate the upper bound of the SAA we use Equation 2.3. This candidate solution is compared with the best candidate solution obtained by performing m optimizations in the full uncertainty framework and then giving 6,000 evaluations for each of the solutions reported. This is done to obatin a fair comparisson between the FU and the SAA. Throughout all the experiments involving SAA, we use $n = 20$ for the number of scenarios fixed before each of the $m = 10$ optimizations. Is important to note that as each evaluation of the SAA requires n simulations, the evaluation budget assigned to a SAA experiment in the ISC framework should be proportional to $\frac{1}{n}$ of the budget assigned to a FU experiment (for the same instance).

Chapter 7

Results and Analysis

Four different experiments settings are considered. Three of them were already mentioned in the previous sections. The one not previously mentioned, aims to validate the correct implementation of the ISC. It shows how the main features of ISC allow finding good solutions. Each evaluation is easily done in the first experiment setting, which permits to allocate a large number of evaluations to every newly sampled solution. The other three experimental settings are:

- the reliability framework with full uncertainty (FU) available to be sampled in each simulation,
- the resiliency framework with full uncertainty available within the simulator,
- and the fixing-uncertainty-before-optimizing (SAA) for the Resiliency framework setting.

The three settings are compared with the solution of the complete enumeration (CE) of all solutions for the $budget = 1$ instance, in order to validate the algorithm for the resolution of this problem.

This chapter concludes that ISC gives more robust solutions when using a SAA methodology to represent the earthquake uncertainty. Then we show that the SAA lower bounds obtained when solving the instances with ISC are unvalid, since ISC is a heuristic procedure. The result was expected, as discussed in Section 6.3. The SAA needs guarantees for convergence to the optimum of problem 2.3. These guarantees are not provided by ISC, but they are provided by the CE approach. The results discussed here also include the CE combined with the SAA (CE-SAA), illustrating how the ISC-SAA obtains unvalid lower bounds.

Table 7.1 presents the hardware used to run each experiment. FU stands for Full Uncertainty allowed to be sampled in each simulation. SAA stands for the Sample Average Approximation.

Problem	Instance	Num. Exp.	Machine
Optimal queue capacity allocation	-	6	Samsung NP870Z5G-X01CL, Intel i7, 2.4 GHz processor, 8GB of RAM
Reliability	$budget = 1, 3$	11	Leftraru Node HP ProLiant SL230s Gen8, two Intel Xeon E5-2660, 20 cores, 48 GB of RAM
Resiliency FU-SAA, ISC-FE	$budget = 1, 4, 7$	71	Leftraru Node HP ProLiant SL230s Gen8, two Intel Xeon E5-2660, 20 cores, 48 GB of RAM

Table 7.1: Experiments done and hardware used to run them.

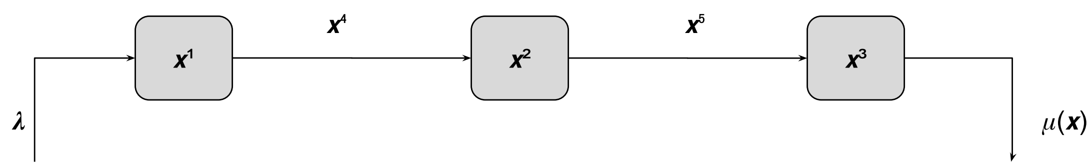

Figure 7.1: Flow diagram Queue Capacity Allocation problem

7.1 Validating ISC: The optimal queue capacity allocation

Now we present a different simulation via optimization problem, and present results that validate the implementation of the algorithm described in Chapter 3. A three-stage flow line (see Figure 7.1) has finite buffer storage space in front of stations 2 and 3 (the number of spaces in the buffers are denoted by x_4 and x_5 respectively) and an infinite number of jobs in front of station 1. There is a single server at each station. The service time distribution at station i has service rate x_i (assuming exponentially distributed time between attentions), $\mathrm{i} = 1, 2, 3$. If the buffer of station i is full, then station $\mathrm{i} - 1$ is blocked and a finished job cannot be released from station $\mathrm{i} - 1$. The total buffer space and the service rates are limited by constraints on space and cost. The objective is to find a buffer allocation and service rates such that the expected throughput over a 1,000 periods planning horizon is maximized. Given an arrival rate $\lambda = 10\frac{arrivals}{period}$, we assume that the times between arrivals distributes $\exp(\lambda)$. The deterministic constraints are $x_1 + x_2 + x_3 \leq 20$, $x_4 + x_5 = 20$, $1 \leq x_\mathrm{i} \leq 20$ and $x_\mathrm{i} \in \mathbb{Z}^+$ for $\mathrm{i} = 1, 2, ..., 5$, implying 21,660 feasible solutions of capacity allocation. The problem is to decide which is the optimum configuration in order to maximize the Flow-Line Throughput rate $\mu(x)$ [11]. The optimal solutions are $(6, 7, 7, 12, 8)$ and $(7, 7, 6, 8, 12)$ with an expected throughput of 5776. Throughput should be estimated after the first 2000 units have been produced (see [28]).

ISC experiment was ran 6 times, on which every run reported a single solution that

Stage-run	Solution	Throughput	CV	N
NGA-1	$[5, 7, 6, 4, 16]$	4699	0.013	40
COMPASS-1	$[6, 7, 7, 12, 8]$	5779	0.010	955
R&S-1	$[6, 7, 7, 12, 8]$	5779	0.010	955
R&S-2	$[7, 7, 6, 8, 12]$	5779	0.010	1119
R&S-3	$[6, 7, 7, 12, 8]$	5777	0.010	952
R&S-4	$[7, 7, 6, 8, 12]$	5778	0.010	892
R&S-5	$[7, 7, 6, 8, 12]$	5775	0.010	1116
R&S-6	$[6, 7, 7, 12, 8]$	5775	0.010	766

Table 7.2: Results experiment ISC. Average solution time 8.86 minutes.

belonged to the optimal set. Table 7.2 shows the solutions reported after every step of the ISC first run, then it shows the results of the rest of runs. The parameters used to set the heuristic are presented in Table 7.3, note that the confidence and the indifference zone parameters (α's and δ's) are set to make each transition step with high precision, this makes the heuristic spend a higher effort on every stage, and explains why every run got a global optimum. On every run the procedure is able to obtain a single solution without having to use the R&S step, the main cause is that in every NGA run a single solution is reported, thereafter this solution is improve locally (were we observe the procedure obtains a local minimum), then, as there is only one solution there is no need for more comparisons. This experiment is useful to validate the first two stages of the procedure, since it is able to replicate results of [28] for a different algorithm.

7.2 14 Busbar Case Study IEEE

Table 7.4 introduces some acronyms nomenclature used in this chapter. We consider the 14 busbar case study of IEEE, see [7], see Figure 7.2 for a diagram. Every line in the system has assigned 100[MW] of capacity and its original impedance value (the official case did not contain buses locations, the admittance value used is arbitrary). We give locations to the buses for the earthquakes, and these are presented in Table 7.19, euclidean distance between coordinates are in kilometers. The peak demand and generation capacity on each node is presented in Table 7.5. Generation capacity installed in bus 1 represents 43% of the offer and 36% of the demand is in bus 3. We consider t_{random} of Chapter 6 set to 0 (which is the peak demand in the system). The probability of line failures is fixed at 5% and the restoration times for every component are deterministic and presented in Table 7.20 in the Appendix.

7.3 Improving Power System Reliability

The experiments for this setting consider the feasible region in Equation 6.1 with $budget = 1$ and 3. For the problem instance in which $budget = 1$, a complete enumeration scheme permits to find its global optimum (if the number of replications is large enough). The experiment

Parameter	Queue cap. alloc.
TT	2
m_G	150
n_0	40
N_0	40
T_G	3
gm	3
α_P (NGA)	0.05
δ_G (NGA)	10.0
α_G (NGA)	0.05
α_L (COMPASS)	0.05
δ_L (COMPASS)	10.0
α_C (R&S)	0.05
δ_C (R&S)	10.0
η	1.5
M	20
Budget NGA	20000
km	10
nonuniform	True
elitism	True
Constraint placement	False
MSSP	True
Sort and Iterative Screen	True
K	20

Table 7.3: Set values for ISC parameters validation experiment.

Notation	Meaning
CL	Case Line(s)
NL	New Line(s)
SB	Seismic Bus(es)
ADC	Added Distributed Capacity in bus(es)
AENS	Average Performance ENS
N	Number Evaluations
CV	Coefficient of Variation

Table 7.4: Nomenclature used in tables' headers.

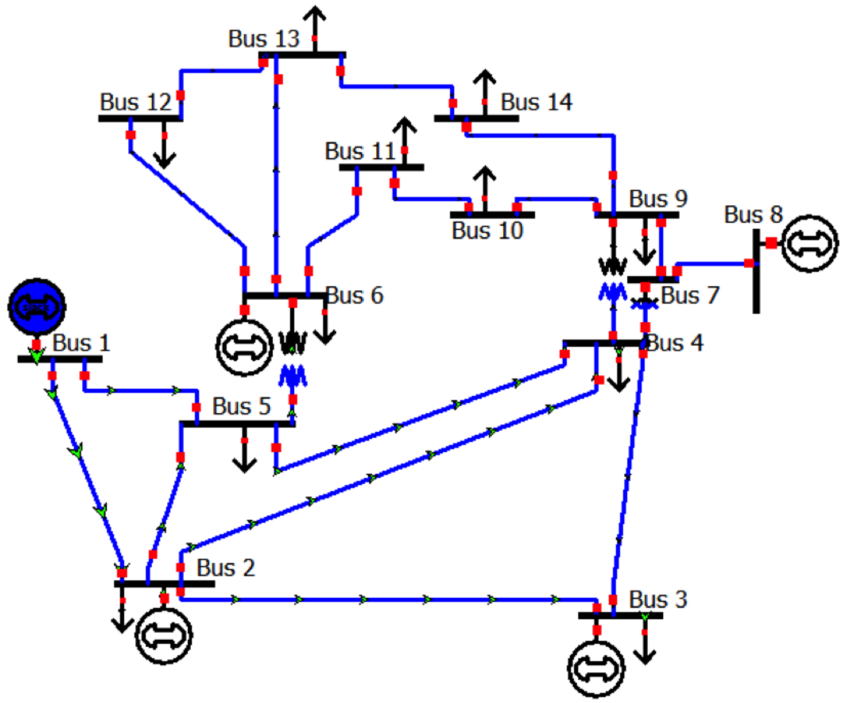

Figure 7.2: Base case diagram.

Bus	Demand	Generation
1	0	332.4
2	21.7	140
3	94.2	100
4	47.8	0
5	7.6	0
6	11.2	100
7	0	0
8	0	100
9	29.5	0
10	9	0
11	3.5	0
12	6.1	0
13	13.5	0
14	14.9	0

Table 7.5: Peak demand and generation capacity allocated on each bar.

allows to illustrate how well the ISC performs by identifying how it finds the top ranked solutions of the CE (Complete Enumeration). For the $budget = 3$ experiment, only the ISC framework is used for optimizing, and the results are presented for a single optimization. The reliability evaluation function is Algorithm 10.

Table 7.6 presents the parameters used to solve each instance.

Parameter	$budget = 1$	$budget = 3$
m_G	35	100
M	10	35
δ_L (COMPASS)	10[MWh]	1[MWh]
δ_C (R&S)	10[MWh]	1[MWh]
TT	3	3
n_0	50	50
N_0	50	50
T_G	3	3
gm	10	10
α_P (NGA)	0.05	0.05
δ_G (NGA)	10[MWh]	10[MWh]
α_G (NGA)	0.05	0.05
α_L (COMPASS)	0.05	0.05
α_C (R&S)	0.05	0.05
η	1.5	1.5
Budget NGA	20000	20000
km	5	5
nonuniform	False	False
elitism	True	True
Constraint placement	False	False
MSSP	True	True
Sort and Iterative Screen	True	True
K	10	10

Table 7.6: Parameter Settings for Reliability setting. Note the first four parameters are different among the instances.

7.3.1 Results

Table 7.7 present a summary of the full ranking of solutions for the $budget = 1$ instance. Table 7.8 presents the effort spent to solve the instance of $budget = 1$ and each of the solutions obtained using ISC. For the 10 ISC's runs, the average solution time is 3.67 minutes using an average of 11991.9 evaluations. Table 7.9 presents the averages of this efforts and is compared with a single run of a CE scheme. The CE evaluates 2000 times each solution and then compares all-against-all, spending approximately eleven times more effort than the ISC on average. For the CE to spend as much as the ISC in computational evaluation budget, it would have to approximately give 200 evaluations to each solution, for this instance. The line (1, 14) does not appears in the set of optimum solutions for ISC experiments, best solution

ISC reports according to the CE is line (5,14) (top 3), however note that these solutions are not statistically different. The ISC finds the top 10 solutions in the CE ranking in 10 different runs of the same instance. When comparing times and number of evaluations in Table 7.9 is easy to note that ISC spends less resources than the CE. For the $budget = 3$ instance, Tables 7.10 and 7.11 present the results. Most of the simulation effort is used in the local stage (COMPASS), as in all the "large" instances in this dissertation, and every stage meets the transition rule with solution ((3, 10),(10, 14),(12, 14)). ISC seems to find patterns that contain 30% of solutions founded in the CE. For the $budget = 3$ instance the solution ENS coefficient of variation value is 1.310.

We observe that the UC does not commit units in buses 6 and 8 to be turned on, therefore, the sequence of DC-OPF's have to make the dispatch by generating capacity in buses 1,2 and 3. Since the system is not prepared to face lines failures, by making bus 14 better connected the system gains more redundancy on available paths for the flow of electricity. As the lines fail in the sequence of periods, the north of the system gets isolated from the supply in the south, hence adding lines that connect the generation (whose main source is bus 1) with a central node in the demand side (bus 14) seems to be a very good solution. When the budget to construct lines increases line (12,14) is added, clearly to use the direct source coming from bus 1, then line (3,10) is added, again to connect directly supply and demand. Note that the worst solution is adding line (1,8), in fact lines (2,8), (5,8), (2,6), (1,6) and (1,3) also appear in the bottom of the ranking, these lines are redundant (very similar as not adding new lines) because the system has enough capacity to provide energy to buses 3,6 and 8 from 1 and 2.

Rank.	NL	AENS	CV	N
1	1, 14	97.7	0.883	2000
2	6, 14	98.0	0.907	2000
3	5, 14	100.0	0.890	2000
4	10, 14	101.0	1.040	2000
5	2, 14	101.3	0.882	2000
...	...	...	...	...
70	2, 8	145.7	0.807	2000
71	()	148.1	0.817	2000
72	1, 8	149.3	0.826	2000

Table 7.7: Results of Full Enumeration for Reliability framework. () means no new lines. Total Time: 33.7 Minutes.

Total Time [min]	Total N	Solution	Value	N	RankCE.
5.12	18045	4, 14	100.7	2288	10
4.89	15919	10, 13	98.2	884	6
8.23	25357	10, 13	99.5	2627	6
3.09	10212	5, 14	96.2	1909	3
2.22	7064	10, 13	96.1	465	6
0.76	2758	5, 14	73.0	54	3
3.00	9219	11, 14	99.8	765	8
1.37	4729	2, 14	99.3	2013	5
5.22	17375	10, 13	98.1	2000	6
2.82	9241	11, 14	102.6	2121	8

Table 7.8: Performances for instance $budget = 1$, ISC, reliability. () means no new lines.

	Full Enumeration	ISC
Time [min]	33.70	3.67
Num. Eval.	144000	11991.9

Table 7.9: Comparison between Full Enumeration and COMPASS for Reliability framework $budget = 1$. ISC average results for 10 runs.

Stage	NL	AENS	N
NGA	(3, 10),(10, 14),(12, 14)	39.9	50
COMPASS	(3, 10),(10, 14),(12, 14)	50.0	229906
R&S	(3, 10),(10, 14),(12, 14)	50.0	229906

Table 7.10: Instance $budget = 3$. Solutions obtained in each stage of the ISC procedure.

	NGA	COMPASS	R&S	Total
Time[min]	2.15	45.94	0.0	48.09
Evaluations	9750	237523	0	247273

Table 7.11: Instance $budget = 3$. Effort spent.

7.4 Improving Power System Resiliency: FU and SAA

As in the reliability approach above, consider the feasible region 6.2. The ISC setting is compared with the CE approach for $budget = 1$. For *budget=4,7* results are obtained only for ISC.

The ISC's parameters are set with the same parameters for both full uncertainty (FU) and SAA frameworks. As explained in section 6.3, the budget of the NGA and initial number of evaluations in SAA are set in a proportion of $n = 20$ with respect to the FU. Since 20 scenarios are fixed in the SAA, therefore a single evaluation requires 20 simulations for every solution. This technique allow to lower variance for small number of samples. For $budget = 1$

Parameter	$budget = 4, 7$ FU	$budget = 4, 7$ SAA	$budget = 1$ FU	$budget = 1$ SAA
m_G	130	130	35	35
n_0	80	4	100	5
N_0	80	4	100	5
Budget NGA	20000	1000	40000	2000
M	40	40	10	10
km	5	5	3	3
δ_L (COMPASS)	0.004	0.004	0.001	0.001
δ_C (R&S)	0.004	0.004	0.001	0.001
nonuniform	True	True	False	False
TT	2	2	2	2
T_G	3	3	3	3
gm	10	10	10	10
α_P (NGA)	0.1	0.1	0.1	0.1
δ_G (NGA)	0.001	0.001	0.001	0.001
α_G (NGA)	0.1	0.1	0.1	0.1
α_L (COMPASS)	0.1	0.1	0.1	0.1
α_C (R&S)	0.1	0.1	0.1	0.1
η	1.5	1.5	1.5	1.5
elitism	True	True	True	True
Constraint placement	False	False	False	False
MSSP	True	True	True	True
Sort and Iterative Screen	True	True	True	True
K	10	10	10	10

Table 7.12: Parameter Settings for ISC experiments. First 8 parameters are different between instances.

the full uncertainty framework ISC is done 10 times (the same amount of optimizations done in the SAA framework), in order to give a fair comparison of the solutions reported and the execution times between the frameworks.

Table 7.12 presents parameters used in the ISC settings for this section. Values δ are set as a fraction of the maximum energy that can be not supplied, i.e., the sum of all demands in the system through all periods and through all demand bars set $D \subseteq V$.

7.4.1 Results

Table 7.13 presents the true ranking obtained by the CE approach for the $budget = 1$ instance. A summary of the results obtained for the CE-FU and the CE-SAA is presented in Table 7.14. A summary of the results in detail shown in Appendix Tables 7.23 to 7.40, that contain solutions reported by each run, how they performed at the final SAA 6,000 evaluations, and detail of each run time solution performance, is presented in Tables 7.15, 7.17 and 7.18.

SAA found more robust solutions than the FU, for the same parameters and number of runs. The results show that solutions performance have more variance in the FU than in the SAA. Solutions obtained by both approaches are not statistically different for $budget = 1, 4$ instances, in the case of $budget = 7$ instance ISC-SAA gave better solutions, statistically speaking. Time performance grows faster when using SAA, with respect to the instance size. For $budget = 1$ performance is better with the SAA. For $budget = 4$ time performance is similar in the different approaches. And for $budget = 7$ time performance is better for the FU.

budget 1 Complete Enumeration (CE)

The procedure was ran a single time for the FU and ten times for the SAA. Note from Table 7.13 that solutions are very near between each other, having approximately a 50% of coefficient of variation. The objective function is flat between solutions and there is one single global optimum that differentiates from the rest. For the FU optimization the "true" ranking is presented and the "true" optimum is revealed to be adding seismic norm (SB) to bus number 3. This bus is critical since most of the demand is allocated there.

In each of the ten optimizations the SAA the results of CE yield the same solution, but with different values (see table 7.14). The results of SAA setting are presented in Table 7.15. Our experiments results present similar rankings between the different optimizations done in the SAA for this instance, this suggests that with 20 scenarios fixed at the beginning we are representing well enough the earthquake uncertainty for $budget = 1$.

budget 1 ISC

For this instance FU and SAA are compared by giving both approaches the same amount of optimizations (10 optimizations). For these instance, ISC works better with the SAA framework, it get less percentage of wrong answers (considering bus 3, the correct answer), as presented in Tables 7.23 and 7.24 in the appendix. The SAA framework improves ISC time results, for this instance, apparently by stabilizing the uncertainty and lowering the variation of the performance. The SAA allows to the algorithm to identify easier the better solutions, when enough representative scenarios are sampled in every optimization. In the case of instance $budget = 1$ SAA used less computational simulation effort than the FU to terminate ISC. The average solution time and the number of evaluations of the runs are presented in Tables 7.17 and 7.18 for the FU and the SAA. The lower bound obtained is shown to be unvalid by Table 7.15. As expected since the CE is able to provide the global optimum for each of the SAA optimizations. The bounds of the solutions obtained by FU (and SAA) are presented in Table 7.25 (and 7.26), in the Appendix. The results lead to the conclusion that, individually comparing solutions, neither of both approaches yield better solutions. The bounds overlap for both of the bests solutions of the approaches.

Rank.	NL	SB	ADC	AENS	N	CV	5%Q	95%Q
1		3		245.9	2000	0.504	241.3	250.5
2		9		260.5	2000	0.537	255.4	265.6
3		4		260.8	2000	0.498	256	265.6
4	8, 13			264.6	2000	0.51	259.6	269.6
5			9,10.0%	264.6	2000	0.514	259.6	269.6
6	5, 14			264.7	2000	0.511	259.7	269.7
7		14		265	2000	0.541	259.7	270.3
8		2		265.9	2000	0.515	260.9	270.9
9	1, 14			266.3	2000	0.505	261.4	271.2
10	5, 10			266.4	2000	0.514	261.4	271.4
11	7, 14			266.5	2000	0.51	261.5	271.5
12	2, 13			267.3	2000	0.506	262.3	272.3
13	5, 9			267.4	2000	0.51	262.4	272.4
14	3, 12			267.5	2000	0.504	262.5	272.5
15			4,10.0%	267.7	2000	0.509	262.7	272.7
16		11		267.8	2000	0.525	262.6	273
17		6		268.5	2000	0.52	263.4	273.6
...	...	...	...	...	...	...	...	...
95	1, 8			278	2000	0.518	272.7	283.3
96	8, 11			278.5	2000	0.525	273.1	283.9
97	2, 11			280.4	2000	0.505	275.2	285.6

Table 7.13: Full Enumeration for FU framework

Framework	SB	Time [min]	AENS	CV	N
FU	3	72.6	245.9	0.503	2000
SAA 1	3	76.7	265.9	0.066	100
SAA 2	3	73.8	263.6	0.073	100
SAA 3	3	72.9	248.8	0.067	100
SAA 4	3	79.6	199.9	0.065	100
SAA 5	3	73.8	233.3	0.067	100
SAA 6	3	73.9	240.2	0.069	100
SAA 7	3	72.4	257.9	0.081	100
SAA 8	3	72.2	233.4	0.082	100
SAA 9	3	71.1	236.8	0.067	100
SAA 10	3	73.5	267.4	0.079	100

Table 7.14: Results of FU and SAA for the Full Enumeration of all solutions solving scheme.

Method	NL	SB	ADC	AENS	N	LB	UB
CE-FU-b1		3		245.9	2000	-	-
CE-SAA-b1		3		244.2	2000	230.1	250.7
ISC-FU-b1		3		247.0	6000	-	-
ISC-SAA-b1		3		245.3	6000	265.5*	248.5
ISC-FU-b4	((5, 9))	3, 4, 6		224.6	6000	-	-
ISC-SAA-b4	((6, 10))	2, 3, 9		225.5	6000	225.4*	228.4
ISC-FU-b7		2, 3, 4, 9	(2, 10.0%), (4, 10.0%)	202.5	6000	-	-
ISC-SAA-b7		3, 4, 9	(2, 30.0%), (4, 10.0%)	194.3	6000	189.0*	196.5

Table 7.15: Results of SAA and FU for ISC for all instances and for CE in instance $budget = 1$, this table summarizes the results of Appendix 7.23 to 7.40. Values with * are unvalid lower bounds.

It is no that big surprise that reinforcing bus 3 is an optimal decision. As noticed in table 7.5 it has 36% of the demand of the system, therefore if the bus fails, then the infrastructure of the distribution part of the system can only supply a level of the demand (leaving the rest unsupplied, see Algorithm 11). Note that bus 3 almost supplies itself, so by strengthening that bus the resiliency of 36% of the demand of the system increases directly. The more connected components in the system are buses 2, 4, 5, 6, 9, (see Table 7.16), now note that all the decisions involving strengthening these buses are almost indistinguishable similar (see Table 7.13). Our results indicate also that we cannot conclude that reinforcing demand distribution against seismic events is better than adding 10% distributed generation, as confidence intervals in Table 7.13 overlap. Almost every new line addition, shown in Table 7.13, sees to connect generation with demand directly, one has to be careful here, since the failures of lines are not incorporated in this framework, if we take the reliability aspect into account, the new lines solutions reported may not be as good. On other hand many of the solutions considering seismic buses and distributed generation capacity probably will remain invariant to consider lines failures, because some of them help to decrease ENS directly.

For the *budget=4* instance, Table 7.29 (resp. 7.30), in the Appendix, presents the results in detail of 10 optimizations done for FU-inside the simulator (resp. SAA). In these instance the results of the ISC-FU and the ISC-SAA are very similar, though it can be shown that the ISC-SAA is still able to provide better percentage of "correct answer" (note the ranking in Table 7.13). On other hand, Tables 7.17 and 7.18 show very similar time and evaluation performance results, for $budget = 4$. Again, as in the $budget = 1$ case, lower bound is proven to be unvalid, see 7.15, since the optimal solution candidate of the SAA yield lower statistical bound, therefore it could be the case, the performance of that solution to be below the SAA lower bound. Finally, when comparing solutions individually from each approach, note that both yield solutions that are not different statistically speaking, this solutions are both local optimal. Solution of the FU seems to prefer better connecting demand in bus 9 to the system rather than strengthening it as in SAA solution, and prefers to strength bus 6 instead to connect it more to the rest of the system as in SAA solution. The results for the $budget = 4$ are different from the ones obtained for the $budget = 7$ instance. The time performance using SAA is worse than the one obtained using FU (see Tables 7.17 and 7.18), still, the quality of results are shown to be more correct in the SAA case and different

Bus	Number connections
1	2
2	4
3	2
4	5
5	4
6	4
7	3
8	1
9	4
10	2
11	2
12	2
13	3
14	2

Table 7.16: Number of lines adjacent to each bus.

Instance	T. NGA	T. COMPASS	T. R&S	Total T.
ISC-FU-b1	3.3	47.1	6.3	56.7
ISC-SAA-b1	2.7	9.4	0.6	12.7
ISC-FU-b4	10.0	17.4	0.2	27.7
ISC-SAA-b4	7.9	16.1	0.0	24.0
ISC-FU-b7	9.0	57.5	0.1	66.7
ISC-SAA-b7	8.9	74.0	0.0	83.0

Table 7.17: Solution time of FU optimizations for all instances

Instance	N. NGA	N. COMPASS	N. R&S	Total N
ISC-FU-b1	7650	115134.6	16503.7	139288.3
ISC-SAA-b1	1115.9	76.2	0	1192.1
ISC-FU-b4	22088	28082.6	486.9	50657.5
ISC-SAA-b4	869.6	1483.7	0	2353.3
ISC-FU-b7	19448	80611.7	246.7	100306.4
ISC-SAA-b7	902.8	5423.7	0	6326.5

Table 7.18: Evaluation performance of FU optimizations for all instances

statistically speaking. As shown in Tables 7.37 and 7.38, the first four solutions of the SAA are better than the best solution of the FU.

The patterns in buses selected to be strengthened seems to be repeated from the other instances when budget is increased, even for the ADG buses selected ISC seems to be selecting critical buses in the system. Note, however, that although best solutions for $budget = 7$ instance do not contain new lines, line (6,10), that appears in table 7.15 for the $budget = 4$ instance was reported by the procedure in Table 7.38 in the Appendix, for $budget = 7$. Line (6,10) could be a good line to add in this setting because it provides an alternate path from the south to the north of the system, since failure of bus 9 decreases capacity that pass through that bus, and then to the north of the system by bus 9 side. Although one should take into account that the conclusions obtained for lines may be incorrect due to the lack of information of the buses location (that should be related to the lines impedance), this will not be a problem if we consider the Chilean SING-SIC system, where full information is available.

7.4.2 Analysis

For these settings, results allow to conclude that fixing uncertainty gives more robust solutions, though it does not yield good confidence intervals. Particularly, when fixing uncertainty it is necessary to face a trade-off that depends on the instance complexity (remember that solution time increases faster with instance size for ISC-SAA in this problem). The question that arises is, what are the different features between the instances that explains why SAA helps in one setting and not in the other. One of this differences is the amount of computational budget used in the local stage for $budget = 1, 4$ instances. For the $budget = 7$ instance, SAA does not help improve solution time performance in the any stage. Therefore we conclude that it is not always truth that the combined framework helps the heuristic in terms of solution time. The results obtained show that solutions obtained with ISC-SAA are more robust that the ones obtained with ISC-FU, this effect is more pronounced in larger instances. Note that the lower bound obtained for the $budget = 1$ instance using ISC in comparison with the FE for the SAA, says something about the performance to obtain good solutions of the heuristic (of course this is consequence of a strictly positive gap on some runs of the ISC-SAA). Solutions analysis allows to conclude which are the critical components of the system and which are its weaknesses, for example, in the 14 bus-bar case study, buses 2, 3, 4, 5, 6 and 9 act as important hubs in the system and the segmentation of supply and demand together with the fact that generation is cheaper in the south of the system makes the system less reliable to inner failures.

Note that the results of the reliability and resiliency frameworks are not comparable, since the source of failures is of different nature. Resilience, although, should also consider reliability, this may lead to different results regarding the construction of new lines in the system.

Conclusions

Perhaps, one of the most important result is that ISC and ISC-SAA are able to provide solutions for the problem of reliability and resiliency in reasonable time. For both frameworks proposed there are some instances that seem to work well empirically, in the proposed optimization frameworks. Even for instances where the objective function is flat and with a low effect from different solutions. The structural behavior of the problem is an important feature to be analyzed. In a real case study, for example the Chilean SING-SIC, detection of this behavior (flat average effect) may lead to revisit the variables in consideration (for example we may add the decision of increasing capacity of some lines or some generating buses).

On other hand it seems that the problem behaves like assumed when applying the ISC heuristic, this is, good performance solutions have good performance neighbors, which allows the COMPASS stage of the algorithm to improve locally the solutions yielded by NGA stage. The SAA helps the ISC framework by lowering the uncertainty. We propose to test a combination of the heuristic local stage with the SAA, by only including it in the local stage. Under the proposed framework one can not calculate the lower bounds of the SAA to estimate a theoretical gap for the solution, since at this point the heuristic produces local optimums, guarantied to be worse or equal to global optimum. The SAA lower bound showed to be biased in some instances experiments, though this bound can have some utility depending on the instance and the problem. Time performance show to decrease faster on the instance for the SAA framework.

One of the models discussed here, approximates the behavior of the system operator under damaged initial conditions. It does not incorporates the reliability point of view of the model discussed in Chapter 6. The combination of both allows to add detail to the operating behavior under simulation. This combination is left proposed for further work. Note that the results of the reliability and resiliency frameworks are not comparable, since the source of failures is of different nature. Resilience, although, should also consider reliability, this may lead to different results regarding the construction of new lines in the system.

Bibliography

[1] Ron Allan. Power system reliability assessment—a conceptual and historical review. *Reliability Engineering & System Safety*, 46(1):3–13, 1994.

[2] Juan C Araneda, H Rudnick, Sebastian Mocarquer, and Pedro Miquel. Lessons from the 2010 chilean earthquake and its impact on electricity supply. In *Power System Technology (POWERCON), 2010 International Conference on*, pages 1–7. IEEE, 2010.

[3] José M Arroyo and Antonio J Conejo. Optimal response of a thermal unit to an electricity spot market. *IEEE Transactions on power systems*, 15(3):1098–1104, 2000.

[4] Miguel Carrión and José M Arroyo. A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. *IEEE Transactions on power systems*, 21(3):1371–1378, 2006.

[5] F Casciati and L Faravelli. Fragility analysis of complex structural systems. 1991.

[6] Modassar Chaudry, Paul Ekins, Kannan Ramachandran, Anser Shakoor, Jim Skea, Goran Strbac, Xinxin Wang, and Jeanette Whitaker. "building a resilient uk energy system". 2011.

[7] Rich Christie. *Power Systems Test Case Archive*, August 1993.

[8] Di Yan and H. Mukai. Stochastic discrete optimization. 1992.

[9] Tharam S Dillon, Kurt W Edwin, H-D Kochs, and RJ Taud. Integer programming approach to the problem of optimal unit commitment with probabilistic reserve determination. *IEEE Transactions on Power Apparatus and Systems*, (6):2154–2166, 1978.

[10] Sebastián Andrés Espinoza. Multi-phase resilience assessment and adaptation of electric power systems throughout the impact of natural disasters. Master's thesis, PONTIFICIA UNIVERSIDAD CATOLICA DE CHILE, 2015.

[11] Michael Fu. *Handbook of simulation optimization*, volume 216 of *International Series in Operations Research & Management Science*. Springer, 2015.

[12] Hong, L. J., and Nelson, B. L. . The tradeoff between sampling and switching: New sequential procedures for indifference-zone selection. 2005.

[13] Hong, L. J., and Nelson, B. L. . Discrete optimization via simulation using compass. *Operations Research 54, 115–129*, 2006.

[14] Hong, L. J., and Nelson, B. L. A brief introduction to optimization via simulation. 2009.

[15] Jie Xu, Barry L. Nelson, Jeff Hong. Industrial strength compass: A comprehensive algorithm and software for optimization via simulation. *Modeling and Computer Simulation, Vol. 20, No. 1, Article 3*, 2010.

[16] Justin Boesel, Barry L. Nelson, Seong-Hee Kim. Using ranking and selection to "clean up" after simulation optimization. *Operations Research*, 2003.

[17] Seong-Hee Kim. Comparison with a standard via fully sequential procedures. 2005.

[18] Charles A Kircher, Robert V Whitman, and William T Holmes. Hazus earthquake loss estimation methods. *Natural Hazards Review*, 7(2):45–59, 2006.

[19] Anton J Kleywegt, Alexander Shapiro, and Tito Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

[20] S. H. Low. Convex relaxation of optimal power flow: A tutorial. In *2013 IREP Symposium Bulk Power System Dynamics and Control - IX Optimization, Security and Control of the Emerging Power Grid*, pages 1–15, Aug 2013.

[21] Miller, R. G. *Simultaneous Statistical Inference.* Springer, 1981.

[22] Rodrigo Moreno and Goran Strbac. Integrating high impact low probability events in smart distribution network security standards through cvar optimisation. In *Resilience of Transmission and Distribution Networks (RTDN) 2015, IET International Conference on*, pages 1–6. IET, 2015.

[23] Mayada Omer. *The Resilience of Networked Infrastructure Systems: Analysis and Measurement*, volume 3. World Scientific, 2013.

[24] Joint Committee on Structural Safety. *General principles: general principles on quality assurance for structures: general principles on reliability for structural design: report*, volume 35. IABSE, 1981.

[25] Mathaios Panteli and Pierluigi Mancarella. The grid: Stronger, bigger, smarter?: Presenting a conceptual framework of power system resilience. *IEEE Power and Energy Magazine*, 13(3):58–66, 2015.

[26] Mathaios Panteli and Pierluigi Mancarella. Modeling and evaluating the resilience of critical electrical power infrastructure to extreme weather events. *IEEE Systems Journal*, 2015.

[27] Edward Paulson. A sequential procedure for selecting the population with the largest mean from k normal populations. *The Annals of Mathematical Statistics*, 35(1):174–180,

1964.

[28] Juta Pichitlamken and Barry L Nelson. A combined procedure for optimization via simulation. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 13(2):155–179, 2003.

[29] A Poulos, S Espinoza, JC de la Llera, and H Rudnick. Seismic risk assessment of spatially distributed electric power systems. In *16th World Conf. on Earthquake Eng., Santiago*, 2017.

[30] Steven M Rinaldi, James P Peerenboom, and Terrence K Kelly. Identifying, understanding, and analyzing critical infrastructure interdependencies. *IEEE Control Systems*, 21(6):11–25, 2001.

[31] Yosef Rinott. On two-stage selection procedures and related probability-inequalities. *Communications in Statistics - Theory and Methods*, 7(8):799–811, 1978.

[32] Hugh Rudnick. "the impact on electricity supply of a major earthquake in chile". Portland, Oregon, 2016. Conference: US Department of Energy: Clear Path IV - Energy Focused Disaster Response Exercise.

[33] Alexander Shapiro, Darinka Dentcheva, and Andrzej Ruszczynski. Lectures on stochastic programming. *MPS-SIAM series on optimization*, 9:1, 2009.

[34] Peng Wang and Lalit Goel. *Power Distribution System Reliability Evaluation Using Both Analytical Reliability Network Equivalent Technique and Time-sequential Simulation Approach*, pages 145–172. Springer London, London, 2010.

[35] Allen J Wood and B Wollenberg. Power generation operation and control—2nd edition. In *Fuel and Energy Abstracts*, volume 37, page 195. Elsevier, 1996.

Appendix

7.5 Some of the code implementation

7.5.1 Revised Mix-D (RMD)

```
import random

def sample(R,n,x0,T):
    t=0
    tt=0
    xx=
    for i in range(0,len(x0)):
        xx.append(x0(i))
    xxx=
    while(True):
        t+=1
        I = random.randint(0,len(x0)-1)
        b=calculatelbubj(xx,R,I)
        while b(0) == b(1):
            I = (I+1)%len(x0)
            b=calculatelbubj(xx,R,I)
        try:
            xx(I)=1.0 * random.randint(b(0),b(1))
        if(t == T):
            xxx.append()
            for em in range(0,len(xx)):
                xxx(tt).append(xx(em))
            tt+=1
            if(tt==n): break
            t=0
    return xxx

def calculatelbubj(xx,R,j):
    ub="a"
    lb=None
    for r in range(0,len(R)):
        if(R(r)(j) < 0 and lb < (1.0/R(r)(j))*(R(r)(len(R(r))-1) - sumprod(R(r)(0:(len(R(r))-1)),xx) + R(r)(j)*xx(j))):
            lb = math.ceil((1.0/R(r)(j))*(R(r)(len(R(r))-1)- sumprod(R(r)(0:(len(R(r))-1)),xx) + R(r)(j)*xx(j)))
        elif(R(r)(j) > 0 and ub > (1.0/R(r)(j))*(R(r)(len(R(r))-1)- sumprod(R(r)(0:(len(R(r))-1)),xx) + R(r)(j)*xx(j))):
```

```
            ub = math.floor((1.0/R(r)(j))*(R(r)(len(R(r))-1)- sumprod(R(r)(0:(len(R(r))-1)),xx) + R(r)(j)*xx(j)))
    return (lb,ub)
```

7.5.2 Fitness Sharing

```
import math

def fs(XX,G,S,n,lnn,AC):
    SS =
    for  i in range(0,len(XX)):
        SS.append((S(i) - (math.pow(G(i),2)/n(i)))/(n(i)-1))
    fs = range(0,len(XX))
    fsS = range(0,len(XX))
    rr =r(lnn(0),XX)
    for i in lnn(0):
        if G(i) >= 0:
            fs(i) = G(i) * (len(AC(i))+1) / n(i)
            fsS(i) = SS(i) * (len(AC(i))+1)
            for j in AC(i):
                fs(j) = G(j) * (len(AC(i))+1) /n(j)
                fsS(j) = SS(j) * (len(AC(i))+1)
        else:
            fs(i) = G(i)  / (n(i)* (len(AC(i))+1))
            fsS(i) = SS(i) / (len(AC(i))+1)
            for j in AC(i):
                fs(j) = G(j)  /(n(j)* (len(AC(i))+1))
                fsS(j) = SS(j) / (len(AC(i))+1)
    for i in lnn(1):
        m=0
        for x in XX:
            dij =
            for k in range(0,len(x)):
                dij.append(XX(i)(k) - x(k))
            Dij = sumprod(dij,dij)
            if Dij < rr:
                m+= 1- (Dij/rr)
        if fs(i) >= 0:
            fs(i) = G(i) * m
            fsS(i) = SS(i) * m
        else:
            fs(i) = G(i) / m
            fsS(i) = SS(i) / m
    return (fs,fsS)


def r(L,XX):
    r="a"
    for i in L:
        for j in L:
            if i != j:
                d =
                for k in range(0,len(XX(i))):
                    d.append(XX(i)(k)-XX(j)(k))
```

```
                dij = math.sqrt(sumprod(d,d))
                if r > dij:
                    r=dij
    return r

def sumprod(X,Y):
    Z=0.0
    for i in range(0,len(X)):
        Z+= X(i)*Y(i)
    return Z
```

7.5.3 Get Selection Probabilities

```

def gselprob(mG,eta,Gg):
    #where eta is a constant between 1 and 2
    s=
    m=
    order =
    for i in range(0,mG):
        s.append((eta - 2.0*(eta-1)*(i/(mG-1.0)))/mG)
    for g in range(0,len(Gg)):
        M=0
        for j in range(0,len(Gg(g))):
            M+= s(Gg(g)(j))
        for j in range(0,len(Gg(g))):
            m.append(M/len(Gg(g)))
            order.append(Gg(g)(j))
    quicksort2(order,0,len(order)-1,(m),1)
    return m


def partition2(A,left,right,M,minimize):
    i=left
    j=right
    r= random.randint(left,right)
    pivot=A(r)*minimize
    while i <= j:
        while A(i)*minimize < pivot:
            i+=1
        while A(j)*minimize > pivot:
            j-=1
        if i<=j:
            aux  = A(i)
            A(i) = A(j)
            A(j) = aux
            for k in range(0,len(M)):
                auxk = M(k)(i)
                M(k)(i) = M(k)(j)
                M(k)(j) = auxk
            i+=1
            j-=1
    return i

```

```
def quicksort2(A, left, right, M=, minimize=1):
    index=partition2(A, left, right, M, minimize)
    if left < index-1 :
        quicksort2(A, left, index-1, M, minimize)
    if index < right:
        quicksort2(A, index, right, M, minimize)
```

7.5.4 Stochastic Universal Sampling (SUS)

```
def sus(m):
    rn=random.random
    prob =
    sel =
    for i in range(0,(len(m)+1)/2):
        if rn + i*2.0/len(m) <= 1:
            prob.append(rn + i*2.0/len(m))
        else:
            prob.append(rn + i*2.0/len(m) - 1)
    prob.sort
    P=0
    i=0
    q = prob.pop(0)
    for p in m:
        P+=p
        if P>q:
            sel.append(i)
            if len(prob)==0: break
            q=prob.pop(0)
        i+=1
    return sel
```

7.5.5 Mating Scheme

```
def mating(sel, ac, xx, m, lnn):
    mate =
    for s in sel:
        #1. Uniformly randomly sample m individuals from the population.
        sample=
        for n in range(0,m):
            sample.append(random.randint(0,len(xx)-1))
        sample.sort
        #2. From those individuals, select the best one that is within
        #the same niche group as x i . If there is no such an
        #individual, select the one that is closest to x i .
        bol=True
        while bol:
            last=sample(0)
            for j in range(1,len(sample)):
                if last == sample(j):
                    sample(j)=random.randint(0,len(xx)-1)
                    sample.sort
```

```
                    bol=True
                    break
                last = sample(j)
                bol = False
        if s in lnn(0):
            group = ac(s)
        elif s in lnn(1):
            group = lnn(1)
        else:
            for local in lnn(0):
                if s in ac(local):
                    group = ac(local)
        for S in sample:
            if S in group and s!=S:
                mate.append(S)
                bol = True
                break
        if not bol:
            d="inf"
            for S in sample:
                h =
                for j in range(0,len(xx(s))):
                    h.append(xx(s)(j)-xx(S)(j))
                f=sumprod(h,h)
                if d > f and s!=S:
                    d=f
                    mate.append(S)
    return mate
```

7.5.6 Crossover

```
import numpy as np

def crossover(xx,sel,mate,r,types):
    barx=np.array(xx)
    barr=np.array(r)
    X=
    t=
    if len(types)==1:
        ere=
        for i in range(len(xx)):
            ere.append()
            for j in range(len(r)):
                ere(len(ere)-1).append(0.0)
        if types(0)(0):
            return crossoverbin(np.array(xx),sel,mate,np.array(r),np.array(ere
                ))
        else:
            return crossoverint(np.array(xx),sel,mate,np.array(r),np.array(ere
                ))

    for p in types:
        t.append()
        for q in p:
```

```
            t(len(t)-1).append(q)
    for i in range(len(sel)):
        X.append((0.0)*len(xx(0)))
        X.append((0.0)*len(xx(0)))
#         for xi in x:
#             X(len(X)-1).append(xi)
    while t!=:
        T=t.pop(random.randint(0,len(t)-1))
        xX=  barx(:,T(1):T(2))
        rR=  barr(:,T(1):T(2))
        if T(1)==0:
            Xx=barx(:,T(2):)
            Rr=barr(:,T(2):)
        else:
            if T(2)==len(xx(0)):
                Xx=barx(:,:T(1))
            else:
                Xx=np.concatenate(( barx(:,:T(1)),barx(:,T(2):) ),axis=1)
            Rr=np.concatenate( (barr(:,:T(1)),barr(:,T(2):)),axis=1 )
        ere= np.transpose(Rr(:,:(len(Rr(0))-1)).dot( np.transpose(Xx)))
        if T(0):
            new=crossoverbin(xX,sel,mate, np.concatenate((rR,Rr(:,(len(np.transpose(Rr))-1):)),axis=1) ,ere)
        else:
            new=crossoverint(xX,sel,mate, np.concatenate((rR,Rr(:,(len(np.transpose(Rr))-1):)),axis=1) ,ere)
#         for i in range(len(new)):
#             k=f
#             for j in range(len(new(i))):
#                 X(i)(k)new(i)(j)
        for i in range(len(sel)):
            for j in range(T(1),T(2)):
                X(2*i)(j)=new(2*i)(j-T(1))
                X(2*i +1)(j)=new(2*i +1)(j-T(1))
    for i in range(len(sel)):
        if not feasibility(X(2*i), r ):
            X(2*i) = xx(sel(i))
        if not feasibility(X(2*i +1), r):
            X(2*i +1) = xx(mate(i))
    return X



#Michalewicz (1996) described several crossover operators for GAs that optimize
#real-valued variables. Our experiments suggested that the arithmetical opera-
#tor works best. Given two parents x i and x j , the arithmetical crossover operator
#produces two offspring x i and x j by generating a number beta from uniform (0,1)
#distribution, and letting x i = beta x i + (1 - beta)x j , and
#x j = (1 - beta)x i + beta x j . Since this is an integer-ordered
#optimization problem, we need to round x i , x j to integers.
def crossoverint(xx,sel,mate,r,ere):
    X =
    for i in range(len(sel)):
```

```
        beta = random.random
        x1 =
        x2 =
        for j in range(0,len(xx(sel(i)))):
            x1.append(round(xx(sel(i))(j)*beta + xx(mate(i))(j)*(1-beta)))
            x2.append(round(xx(sel(i))(j)*(1-beta) + xx(mate(i))(j)*beta))
        if not feasibility(x1, np.concatenate( ( r(:,0:(len(np.transpose(r))
            -1)) , r(:, (len(np.transpose(r))-1):len(np.transpose(r))) - np.
            transpose(ere( sel(i):(sel(i)+1) ))  ) ,axis=1) ):
            x1 = xx(sel(i))
        if not feasibility(x2, np.concatenate( ( r(:,0:(len(np.transpose(r))
            -1)) , r(:, (len(np.transpose(r))-1):len(np.transpose(r))) - np.
            transpose(ere( sel(i):(sel(i)+1) ))  ) ,axis=1) ):
            x2 = xx(mate(i))
        X.append(x1)
        X.append(x2)
    return X


def crossoverbin(xx,sel,mate,r,ere):
    X=
    for i in range(0,len(sel)):
        beta = random.random
        x1 =
        x2 =
        sumx=
        if beta >= 0.5:
            t1= True
        else:
            t1= False
        f1= False
        for j in range(0,len(xx(sel(i)))):
            sumx.append(xx(sel(i))(j) + xx(mate(i))(j))
        for sx in sumx:
            if sx==0:
                x1.append(0.0)
                x2.append(0.0)
            elif sx==2:
                x1.append(1.0)
                x2.append(1.0)
            else:
                if t1:
                    x1.append(1.0)
                    x2.append(0.0)
                    t1=False
                else:
                    x1.append(0.0)
                    x2.append(1.0)
                    t1=True
                if f1:
                    beta = random.random
                    if beta >= 0.5:
                        t1= True
                    else:
                        t1= False
                    f1=False
```

```
                    else:
                        f1=True
            if not feasibility(x1,np.concatenate( ( r(:,:(len(np.transpose(r))-1)) , r(:,(len(np.transpose(r))-1):) - np.transpose(ere(sel(i):(sel(i)+1))) ) ,axis=1)):
                x1 = xx(sel(i))
            if not feasibility(x2,np.concatenate( ( r(:,:(len(np.transpose(r))-1)) , r(:,(len(np.transpose(r))-1):) - np.transpose(ere(sel(i):(sel(i)+1))) ) ,axis=1)):
                x2 = xx(mate(i))
            X.append(x1)
            X.append(x2)
    return X

def feasibility(x,r):
    for R in r:
        if sumprod(R(0:(len(R)-1)),x) > R(len(R)-1):
            return False
    return True
```

7.5.7 Mutation

```

def nonuniformmutation(C,R,K,bb,be):
    for x in C:
        I=random.randint(0,len(x)-1)
        lbub=calculatelbubj(x,R,I)
        if random.random <= 0.5:
            deltakrix= (lbub(1)-x(I))*random.random*max((0.005),math.pow(max((0.0,1.0 - (bb*1.0/K))),be))(0)
            x(I)+= round(deltakrix)
        else:
            deltakrix= (x(I)-lbub(0))*random.random*max((0.005),math.pow(max((0.0,1.0 - (bb*1.0/K))),be))(0)
            x(I)-= round(deltakrix)
    return C

def mutation(nonuniform,C,R,K,bb,be,x):
    if nonuniform:
        mutation=nonuniformmutation(C,R,K,bb,be)
    else:
        for x  in C:
            mutation.append(sample(R,1,x,1)(0))
```

7.5.8 Dominance

```


def dominance(lnn,AC,G,S,n,alpha,minimize):
    beta= math.pow(1-alpha,1.0/(len(lnn(0))))
    H=
```

```
    ni=
    SSEi=
    GG=
    vi=
    for i in range(0,len(lnn(0))):
        ni.append(n(lnn(0)(i)))
        SSEi.append(S(lnn(0)(i)) - (math.pow(G(lnn(0)(i)),2)/n(lnn(0)(i))))
        GG.append(G(lnn(0)(i)))
        vi.append(n(lnn(0)(i))-1)
        for j in AC(lnn(0)(i)):
            ni(i)+=n(j)
            SSEi(i)+=S(j) - (math.pow(G(j),2)/n(j))
            GG(i)+=G(j)
            vi(i)+=n(j)-1
    for i in range(0,len(lnn(0))):
        bol=True
        for j in range(0,len(lnn(0))):
            if i!=j:
                ti2=math.pow(calculatetstudent(beta,vi(i)),2)
                tj2=math.pow(calculatetstudent(beta,vi(j)),2)
                wij= math.sqrt((ti2*SSEi(i)/(vi(i)*n(i)))+(tj2*SSEi(j)/(vi(j)*
                    n(j))))

                if minimize*GG(i)/ni(i) > minimize*GG(j)/ni(j) + wij:
                    bol=False
                    break
        if bol:
            H.append(lnn(0)(i))
    return H
```

bus	x	y
1	280	18
2	255	25
3	220	25
4	244	74
5	270	67
6	275	310
7	240	220
8	212	228
9	244	318
10	250	309
11	264	306
12	280	405
13	275	420
14	230	470

Table 7.19: Buses locations

Parameter	Value
$\frac{R^t}{\sum_{\mathrm{i} \in V} \mathrm{d}_{\mathrm{i}}^t}$	0.05
Transmission Lines Failure rate	0.05
Restoration times Generation units	10
Restoration times Substations	5
Restoration times Lines	8

Table 7.20: Simulation Parameters.

Damage state	$\mu_1^{\mathrm{d}}s$	$\sigma_1^{\mathrm{d}}s$	$\mu_2^{\mathrm{d}}s$	$\sigma_2^{\mathrm{d}}s$
1	0.10	0.60	0.15	0.60
2	0.20	0.50	0.25	0.50
3	0.30	0.40	0.35	0.40
4	0.50	0.40	0.70	0.40

Table 7.21: Fragility curve parameters, 2 is strengthened (note that μ is shifted to the left). Here every generation unit and bus in the network share the same fragility curve parameters.

Location	x	y	z
North	250	50	50
Center	250	300	50
South	250	440	50

Table 7.22: Earthquakes centers locations.

7.6 Case Parameters: 14 bus IEEE Case Study

7.7 Resiliency results

7.7.1 Budget 1

NL	SB	ADG	AENS	N	CV
((3, 12))			269.046	9623	0.513
	3		246.619	7629	0.499
((8, 12))			269.644	7454	0.514
		((13,10.0%))	270.096	6732	0.521
((8, 13))			270.361	7291	0.515
((1, 4))			271.285	5819	0.518
			272.567	7752	0.512
((7, 11))			274.355	6899	0.520
	3		248.052	8366	0.508
((3, 14))			268.594	8573	0.503
((5, 12))			268.621	10950	0.514
((2, 14))			270.159	8449	0.512
((5, 11))			273.006	8750	0.511
((9, 11))			274.387	8450	0.516
((4, 11))			274.558	8974	0.514
((10, 14))			276.122	7581	0.518
		((5,10.0%))	276.735	8766	0.514
((3, 14))			269.662	8286	0.506
	4		257.388	6542	0.513
((5, 13))			270.716	6454	0.508
((7, 13))			275.062	6031	0.514
	9		258.542	9056	0.531
((2, 12))			271.876	7809	0.514
((7, 11))			272.667	9435	0.517
((3, 5))			273.462	7355	0.519
		((2,10.0%))	261.985	5490	0.496
((4, 13))			271.402	7779	0.519
	11		271.475	7260	0.521
	2		271.598	7055	0.511
	3		247.582	8485	0.500

Table 7.23: Results of ten optimizations of FU-ISC $budget = 1$

NL	SB	ADG	AENS	N	CV
((3, 13))			276.902	207	0.006
		((5,10.0%))	284.195	228	0.008
	3		253.337	233	0.005
	9		298.730	413	0.004
	3		271.046	427	0.003
	3		269.959	102	0.013
	9		264.249	201	0.006
		((3,10.0%))	265.580	245	0.005
((2, 12))			269.735	162	0.007
	3		232.764	70	0.018
		((2,10.0%))	265.286	252	0.005
		((8,10.0%))	266.005	211	0.006
((6, 10))			271.775	158	0.008

Table 7.24: Results of ten optimizations of SAA-ISC $budget = 1$

7.7.2 Budget 4

NL	SB	ADG	AENS	N	CV
((4, 12), (6, 8))	3, 9		231.552	16	0.084
((1, 4))	4, 9	((2,10.0%))	232.004	20	0.050
((6, 10))	3, 6	((1,10.0%))	234.501	16	0.034
	6, 9	((12,20.0%))	228.758	16	0.065
((1, 14), (5, 10))	3, 4		224.260	24	0.051
	3, 9, 13	((5,10.0%))	225.272	20	0.053
((3, 12), (8, 10))	3, 9		215.295	28	0.048
((5, 13))	3, 9	((11,10.0%))	196.599	17	0.045
((2, 14))	3, 6	((12,10.0%))	208.567	24	0.076
((1, 6), (5, 11))	3	((2,10.0%))	211.200	22	0.052
	3	((2,20.0%), (10,10.0%))	235.798	12	0.065
((6, 10))	2, 3, 9		237.010	16	0.109
	3, 13, 14	((8,10.0%))	241.452	24	0.033
((8, 13), (12, 14))	3		267.388	15	0.130
((6, 8))	4, 6	((8,10.0%))	235.361	24	0.057

Table 7.30: Results of ten optimizations of SAA-ISC, $budget = 4$ instance

7.7.3 Budget 7

NL	SB	ADG	AENS	N	Std.Dev	5%Q	95%Q
	3,		246.969	6000	123.614	243.842	250.097
	3,		247.083	6000	121.889	243.999	250.168
	3,		249.277	6000	124.955	246.115	252.438
	4,		257.928	6000	133.170	254.558	261.297
	9,		258.645	6000	142.004	255.051	262.238
		(2,10.0%)	262.062	6000	130.792	258.753	265.372
(3,14)			265.843	6000	134.338	262.443	269.242
(8,12)			269.048	6000	136.615	265.591	272.505
(2,12)			269.330	6000	136.903	265.866	272.794
(3,12)			269.535	6000	137.458	266.057	273.013
(5,13)			269.970	6000	136.315	266.521	273.419
(2,14)			270.176	6000	138.492	266.672	273.681
(3,14)			270.739	6000	136.897	267.275	274.203
	2,		270.794	6000	138.065	267.301	274.288
(5,11)			271.540	6000	139.869	268.001	275.079
(7,13)			271.603	6000	137.371	268.127	275.079
(3,5)			272.084	6000	140.017	268.541	275.627
(10,14)			272.248	6000	141.910	268.657	275.839
(5,12)			272.370	6000	138.660	268.861	275.878
		(13,10.0%)	272.953	6000	143.218	269.329	276.577
(7,11)			273.060	6000	139.097	269.540	276.579
	11,		273.109	6000	142.129	269.513	276.706
(4,11)			273.111	6000	141.017	269.543	276.679
			273.233	6000	139.217	269.711	276.756
(9,11)			273.465	6000	138.987	269.949	276.982
(7,11)			273.481	6000	141.387	269.903	277.059
(1,4)			273.680	6000	142.844	270.066	277.294
(8,13)			274.991	6000	142.538	271.384	278.598
		(5,10.0%)	275.880	6000	142.792	272.267	279.493
(4,13)			276.103	6000	140.284	272.554	279.653

Table 7.25: Ranking of solutions of 10 FU ISC *budget* = 1 instance optimizations, and statistical confidence interval for each solution.

NL	SB	ADG	AENS	N	Std.Dev	5%Q	95%Q
	3,		245.343	6000	123.645	242.215	248.472
	3,		247.431	6000	123.429	244.308	250.555
	3,		248.517	6000	124.344	245.371	251.663
	3,		250.631	6000	124.238	247.487	253.774
	9,		258.560	6000	139.841	255.021	262.098
	9,		259.121	6000	139.706	255.586	262.656
		(2,10.0%)	263.395	6000	129.590	260.116	266.674
(3,13)			267.221	6000	135.358	263.796	270.646
		(3,10.0%)	267.943	6000	137.456	264.464	271.421
		(8,10.0%)	268.294	6000	139.221	264.771	271.817
(6,10)			272.083	6000	140.053	268.539	275.627
		(5,10.0%)	272.680	6000	142.496	269.075	276.286
(2,12)			273.612	6000	139.271	270.088	277.136

Table 7.26: Ranking of solutions of 10 SAA ISC *budget* = 1 instance optimizations, and statistical confidence interval for each solution.

Run	T. NGA	T. COMPASS	T. R&S	Total T.	N NGA	N COMPASS	N R&S	Total N
0	3.273	10.003	0.002	13.277	7500	24 291	0	31 791
1	3.432	40.819	5.742	49.993	7800	98 944	14 867	121 611
2	3.295	37.219	11.088	51.601	7600	89 394	29 201	126 195
3	3.536	91.623	0.002	95.161	8000	224 797	0	232 797
4	3.560	95.287	19.430	118.277	8400	238 354	50 840	297 594
5	3.296	19.551	0.000	22.848	7500	45 790	0	53 290
6	3.215	33.394	7.239	43.847	7400	82 018	18 718	108 136
7	3.517	37.536	9.388	50.441	8100	91 562	24 283	123 945
8	3.142	37.774	10.254	51.169	7200	93 049	27 128	127 377
9	3.070	67.611	0.009	70.690	7000	163 147	0	170 147
Average	3.333	47.082	6.315	56.730	7650	115 134.600	16 503.700	139 288.300

Table 7.27: Performance of FU optimizations for instance $budget = 1$

Run	T. NGA	T. COMPASS	T. R&S	Total T.	N NGA	N COMPASS	N R&S	Total N
0	3.048	7.985	1.720	12.753	967	223	0	1190
1	2.160	1.979	0.000	4.139	244	0	0	244
2	3.129	17.586	0.026	20.741	2165	0	0	2165
3	1.471	3.483	0.000	4.954	437	0	0	437
4	2.802	2.165	0.766	5.732	263	93	0	356
5	2.892	5.558	0.001	8.450	705	0	0	705
6	3.318	34.562	1.976	39.856	3932	236	0	4168
7	3.122	7.426	0.497	11.045	890	61	0	951
8	2.100	2.127	0.000	4.227	263	0	0	263
9	3.099	10.709	1.227	15.034	1293	149	0	1442
Average	2.714	9.358	0.621	12.693	1115.900	76.200	0	1192.100

Table 7.28: Performance of SAA optimizations for instance $budget = 1$

NL	SB	ADG	AENS	N	CV
((3, 9))	4, 9		236.848	376	0.535
((7, 11), (8, 14))	4		243.513	396	0.507
((5, 14), (8, 14), (11, 13))	4		247.328	443	0.532
	6	((3,10.0%) ,(12,10.0%) ,(14,10.0%))	247.963	448	0.528
((4, 13))		((3,10.0%), (12,10.0%), (14,10.0%))	256.409	463	0.519
((1, 9), (1, 10), (2, 8))	12		257.792	457	0.544
((12, 14))	4, 9	((2,10.0%))	212.989	249	0.510
((5, 9))	3, 4, 6		223.395	275	0.507
((2, 14), (5, 10))	3		235.953	185	0.453
((6, 7), (9, 11))	9, 12		244.450	283	0.501
((2, 6), (8, 13))	3	((11,10.0%))	226.496	269	0.490
((1, 13), (2, 7))	3		238.265	369	0.487
((8, 11))	3, 12	((4,10.0%))	238.692	408	0.480
((3, 5), (4, 13))	3		240.962	230	0.442
((2, 13), (7, 10))	2, 3		224.448	622	0.491
((10, 13))	2, 3, 9		212.494	391	0.508
((1, 10), (2, 10))	3, 9		224.420	286	0.508
((1, 11), (4, 11))	3, 6		230.083	311	0.542
((6, 9), (6, 10), (9, 13))	3		233.143	369	0.503
((1, 3), (11, 14))	2, 3		235.742	315	0.502
((11, 12))	3, 12		246.456	429	0.540
((1, 10), (3, 5), (3, 14))	10		249.675	193	0.498
((4, 8), (8, 11), (10, 14), (11, 12))			276.103	277	0.534
	1, 3, 10		236.133	80	0.491
((3, 8), (3, 13), (8, 11), (10, 12))			239.947	88	0.500

Table 7.29: Results of ten optimizations of FU-ISC, $budget = 4$ instance

NL	SB	ADG	AENS	N	Std.Dev	5%Q	95%Q
(5,9)	3, 4, 6,		224.559	6000	112.832	221.704	227.414
(10,13)	2, 3, 9,		227.319	6000	116.853	224.362	230.276
(1,10) (2,10)	3, 9,		230.434	6000	120.070	227.396	233.472
(12,14)	4, 9,	(2,10.0%)	233.405	6000	118.873	230.397	236.412
(2,13) (7,10)	2, 3,		235.404	6000	114.093	232.517	238.291
(1,3) (11,14)	2, 3,		237.382	6000	117.983	234.397	240.368
(1,11) (4,11)	3, 6,		241.383	6000	122.713	238.278	244.488
(2,6) (8,13)	3,	(11,10.0%)	243.353	6000	120.700	240.299	246.407
(3,5) (4,13)	3,		243.508	6000	120.090	240.469	246.547
(8,11)	3, 12,	(4,10.0%)	243.654	6000	122.569	240.553	246.755
(3,9)	4, 9,		244.559	6000	128.110	241.318	247.801
(11,12)	3, 12,		244.674	6000	125.757	241.492	247.856
(1,13) (2,7)	3,		244.989	6000	118.713	241.986	247.993
(2,14) (5,10)	3,		245.413	6000	121.833	242.330	248.495
	1, 3, 10,		245.635	6000	123.495	242.510	248.760
(6,9) (6,10) (9,13)	3,		246.205	6000	120.305	243.160	249.249
(5,14) (8,14) (11,13)	4,		253.192	6000	127.926	249.956	256.429
(7,11) (8,14)	4,		254.050	6000	130.478	250.749	257.352
(6,7) (9,11)	9, 12,		257.387	6000	140.634	253.828	260.945
	6,	(3,10.0%) (12,10.0%) (14,10.0%)	257.907	6000	132.959	254.543	261.271
(4,13)		(3,10.0%) (12,10.0%) (14,10.0%)	263.972	6000	135.160	260.552	267.392
(3,8) (3,13) (8,11) (10,12)			264.219	6000	135.756	260.784	267.654
(1,10) (3,5) (3,14)	10,		267.149	6000	136.460	263.697	270.602
(4,8) (8,11) (10,14) (11,12)			271.288	6000	139.097	267.769	274.808
(1,9) (1,10) (2,8)	12,		273.511	6000	140.558	269.954	277.068

Table 7.31: Ranking of solutions of 10 FU ISC *budget* = 4 instance optimizations, and statistical confidence interval for each solution.

NL	SB	ADG	AENS	N	Std.Dev	5%Q	95%Q
(6,10)	2, 3, 9,		225.460	6000	115.763	222.531	228.389
	3, 9, 13,	(5,10.0%)	225.883	6000	122.401	222.786	228.981
(1,14) (5,10)	3, 4,		227.497	6000	110.513	224.701	230.294
(5,13)	3, 9,	(11,10.0%)	229.575	6000	117.726	226.596	232.554
(4,12) (6,8)	3, 9,		230.241	6000	117.654	227.264	233.218
	3,	(2,20.0%) (10,10.0%)	231.201	6000	105.760	228.525	233.877
(2,14)	3, 6,	(12,10.0%)	234.193	6000	119.322	231.174	237.212
(3,12) (8,10)	3, 9,		234.326	6000	117.331	231.357	237.295
	3, 13, 14,	(8,10.0%)	235.158	6000	125.756	231.976	238.340
(1,4)	4, 9,	(2,10.0%)	235.708	6000	119.580	232.682	238.733
(1,6) (5,11)	3,	(2,10.0%)	235.993	6000	114.909	233.085	238.900
(6,10)	3, 6,	(1,10.0%)	237.155	6000	119.687	234.127	240.184
(6,8)	4, 6,	(8,10.0%)	243.732	6000	128.502	240.480	246.983
(8,13) (12,14)	3,		246.464	6000	121.472	243.391	249.538
	6, 9,	(12,20.0%)	248.379	6000	137.268	244.906	251.852

Table 7.32: Ranking of solutions of 10 SAA ISC *budget* = 4 instance optimizations, and statistical confidence interval for each solution.

Run	T. NGA	T. COMPASS	T. R&S	Total T.	N NGA	N COMPASS	N R&S	Total N
0	10.206	17.498	0.940	28.645	20 560	28 550	2020	51 130
1	13.403	26.433	0.158	39.994	29 040	39 926	292	69 258
2	9.138	6.611	0.117	15.867	20 640	11 606	216	32 462
3	8.723	42.257	0.427	51.408	20 160	66 620	908	87 688
4	8.966	14.867	0.000	23.833	20 560	25 014	0	45 574
5	12.520	26.728	0.000	39.249	28 960	44 217	0	73 177
6	8.841	31.946	0.531	41.319	20 320	52 661	1163	74 144
7	9.810	3.285	0.199	13.294	20 560	5195	270	26 025
8	9.096	1.260	0.000	10.356	20 000	2057	0	22 057
9	9.655	3.004	0.000	12.658	20 080	4980	0	25 060
Average	10.036	17.389	0.237	27.662	22 088	28 082.600	486.900	50 657.500

Table 7.33: Performance of FU optimizations for instance $budget = 4$

Run	T. NGA	T. COMPASS	T. R&S	Total T.	N NGA	N COMPASS	N R&S	Total N
0	9.421	39.280	0.016	48.717	1020	3588	0	4608
1	8.719	21.957	0.028	30.704	1008	2049	0	3057
2	9.032	11.398	0.023	20.453	1024	1120	0	2144
3	4.607	1.736	0.000	6.343	520	184	0	704
4	10.006	22.785	0.023	32.814	1012	2206	0	3218
5	9.024	23.063	0.026	32.113	1020	2014	0	3034
6	9.722	28.200	0.025	37.947	1024	2536	0	3560
7	9.373	1.437	0.000	10.810	1028	160	0	1188
8	4.537	0.819	0.000	5.355	520	80	0	600
9	4.727	10.238	0.000	14.965	520	900	0	1420
Average	7.917	16.091	0.014	24.022	869.600	1483.700	0	2353.300

Table 7.34: Performance of SAA optimizations for instance $budget = 4$

NL	SB	ADG	AENS	N	CV
((2, 12), (5, 13), (7, 12), (11, 13))	3, 6		217.098	145	0.470
((2, 9), (8, 10), (8, 12), (11, 13))	2, 3, 5		229.797	157	0.506
((1, 11), (2, 14))	3, 4, 10, 11, 13		192.151	152	0.514
((1, 10), (7, 14))	3, 4	((2,10.0%), (9,20.0%))	196.243	146	0.458
((10, 12))	2, 3, 5, 6, 9	((2,10.0%))	209.426	192	0.544
((1, 11), (1, 14))	2, 3, 4, 9	((8,10.0%))	200.106	283	0.515
((2, 14), (7, 13), (8, 10))	3, 9, 14	((2,10.0%))	202.310	206	0.433
((8, 13), (9, 13))	3, 4, 5, 14	((3,10.0%))	205.096	275	0.531
((10, 12))	3, 4, 9, 12, 13	((8,10.0%))	205.363	332	0.522
((5, 13))	2, 3, 11	((2,10.0%), (4,10.0%))	211.812	260	0.475
((4, 6))	4, 9, 10, 12	((2,10.0%))	214.022	289	0.500
	3, 4, 5, 10	((1,10.0%), (3,10.0%))	200.821	205	0.463
((1, 11), (4, 12))	3, 9, 10, 13, 14		203.240	262	0.517
((2, 13))	3, 12	((2,10.0%), (4,20.0%), (14,10.0%))	206.045	259	0.436
((1, 4), (8, 9), (12, 14))	3, 4, 9, 10		209.445	285	0.508
((4, 10), (4, 14), (10, 13))	3, 9, 13	((8,10.0%))	215.987	357	0.550
((2, 11))	4, 9, 13	((4,10.0%), (6,10.0%))	221.835	412	0.554
((3, 9))	3, 9	((2,10.0%), (4,20.0%))	194.338	204	0.489
	2, 3, 4, 9	((2,10.0%), (4,10.0%))	187.794	187	0.440
((3, 10))	3, 4, 9, 13	((1,20.0%))	193.301	285	0.536
((1, 8), (3, 13), (6, 8))	3, 9, 12		216.537	185	0.501

Table 7.35: Results of ten optimizations of FU-ISC, $budget = 7$ instance

NL	SB	ADG	AENS	N	CV
((2, 12))	3, 4, 9, 14	((2,10.0%), (4,10.0%))	155.184	28	0.069
((7, 14))	3, 4, 6, 9	((2,20.0%))	156.353	36	0.068
((1, 12))	3, 4, 6, 10	((2,10.0%), (4,10.0%))	219.462	36	0.066
((1, 14))	2, 3, 4	((4,10.0%), (9,20.0%))	237.338	28	0.071
	4, 6, 9, 11	((2,20.0%), (5,10.0%))	197.501	24	0.074
((6, 10))	3, 4, 6, 9, 10	((3,10.0%))	195.651	36	0.082
	3, 4, 9, 14	((2,20.0%), (14,10.0%))	198.553	28	0.087
((9, 13))	3, 6, 9	((2,20.0%), (4,10.0%))	212.595	118	0.064
	3, 4, 14	((1,10.0%), (3,10.0%), (4,10.0%),(13,10.0%))	213.081	52	0.068
((2, 9), (4, 14))	3, 4, 9, 14	((3,10.0%))	187.547	28	0.068
	3, 4, 9	((2,30.0%), (4,10.0%))	182.354	24	0.064
((10, 14))	3, 4, 9, 14	((2,20.0%))	172.507	50	0.066

Table 7.36: Results of ten optimizations of SAA-ISC, $budget = 7$ instance

NL	SB	ADG	AENS	N	Std.Dev	5%Q	95%Q
	2, 3, 4, 9,	(2,10.0%) (4,10.0%)	202.530	6000	98.223	200.045	205.015
(1,11) (1,14)	2, 3, 4, 9,	(8,10.0%)	202.798	6000	104.563	200.152	205.444
(10,12)	3, 4, 9, 12, 13,	(8,10.0%)	203.465	6000	110.043	200.681	206.250
(3,10)	3, 4, 9, 13,	(1,20.0%)	208.517	6000	109.106	205.757	211.278
(10,12)	2, 3, 5, 6, 9,	(2,10.0%)	209.193	6000	103.027	206.586	211.799
(1,4) (8,9) (12,14)	3, 4, 9, 10,		215.184	6000	110.616	212.385	217.983
	3, 4, 5, 10,	(1,10.0%) (3,10.0%)	215.861	6000	106.237	213.173	218.549
(8,13) (9,13)	3, 4, 5, 14,	(3,10.0%)	217.028	6000	107.097	214.318	219.738
(2,14) (7,13) (8,10)	3, 9, 14,	(2,10.0%)	217.402	6000	109.455	214.632	220.171
(1,11) (2,14)	3, 4, 10, 11, 13,		217.416	6000	111.172	214.603	220.229
(1,10) (7,14)	3, 4,	(2,10.0%) (9,20.0%)	217.494	6000	104.333	214.854	220.134
(1,11) (4,12)	3, 9, 10, 13, 14,		218.613	6000	121.551	215.538	221.689
(3,9)	3, 9,	(2,10.0%) (4,20.0%)	221.824	6000	106.628	219.126	224.522
(4,10) (4,14) (10,13)	3, 9, 13,	(8,10.0%)	223.052	6000	119.477	220.029	226.075
(5,13)	2, 3, 11,	(2,10.0%) (4,10.0%)	227.164	6000	108.204	224.426	229.902
(1,8) (3,13) (6,8)	3, 9, 12,		227.960	6000	116.008	225.025	230.896
(4,6)	4, 9, 10, 12,	(2,10.0%)	230.146	6000	120.952	227.086	233.206
(2,13)	3, 12,	(2,10.0%) (4,20.0%) (14,10.0%)	232.312	6000	111.689	229.486	235.139
(2,9) (8,10) (8,12) (11,13)	2, 3, 5,		236.031	6000	114.922	233.123	238.939
(2,11)	4, 9, 13,	(4,10.0%) (6,10.0%)	237.998	6000	127.878	234.762	241.233
(2,12) (5,13) (7,12) (11,13)	3, 6,		238.203	6000	118.843	235.196	241.210

Table 7.37: Ranking of solutions of 10 FU ISC *budget* = 7 instance optimizations, and statistical confidence interval for each solution.

NL	SB	ADG	AENS	N	Std.Dev	5%Q	95%Q
	3, 4, 9,	(2,30.0%) (4,10.0%)	194.316	6000	87.451	192.104	196.529
(7,14)	3, 4, 6, 9,	(2,20.0%)	194.627	6000	90.613	192.334	196.920
(10,14)	3, 4, 9, 14,	(2,20.0%)	194.875	6000	94.731	192.478	197.272
	3, 4, 9, 14,	(2,20.0%) (14,10.0%)	197.329	6000	96.170	194.896	199.762
(2,12)	3, 4, 9, 14,	(2,10.0%) (4,10.0%)	201.704	6000	102.057	199.122	204.287
(2,9) (4,14)	3, 4, 9, 14,	(3,10.0%)	205.013	6000	104.786	202.362	207.664
(6,10)	3, 4, 6, 9, 10,	(3,10.0%)	206.072	6000	107.328	203.357	208.788
	4, 6, 9, 11,	(2,20.0%) (5,10.0%)	209.655	6000	105.372	206.989	212.321
(9,13)	3, 6, 9,	(2,20.0%) (4,10.0%)	210.177	6000	102.208	207.591	212.763
(1,12)	3, 4, 6, 10,	(2,10.0%) (4,10.0%)	210.539	6000	101.635	207.967	213.110
(1,14)	2, 3, 4,	(4,10.0%) (9,20.0%)	215.264	6000	103.148	212.654	217.874
	3, 4, 14,	(1,10.0%) (3,10.0%) (4,10.0%) (13,10.0%)	218.165	6000	109.675	215.390	220.940

Table 7.38: Ranking of solutions of 10 SAA ISC *budget* = 7 instance optimizations, and statistical confidence interval for each solution.

Run	T. NGA	T. COMPASS	T. R&S	Total T.	N NGA	N COMPASS	N R&S	Total N
0	9.429	4.932	0.044	14.405	20 560	8803	61	29 424
1	8.993	53.685	0.000	62.678	20 480	60 978	0	81 458
2	9.132	45.770	0.003	54.905	20 560	62 235	0	82 795
3	8.939	19.915	0.062	28.916	20 320	30 818	109	51 247
4	9.754	132.560	0.579	142.893	20 640	188 710	959	210 309
5	10.155	9.056	0.000	19.211	20 640	13 978	0	34 618
6	9.999	120.160	0.518	130.677	20 080	171 621	1083	192 784
7	9.652	53.912	0.057	63.621	20 480	78 311	83	98 874
8	8.942	131.925	0.102	140.970	20 320	185 570	172	206 062
9	5.292	3.283	0.000	8.575	10 400	5093	0	15 493
Average	9.029	57.520	0.137	66.685	19 448	80 611.700	246.700	100 306.400

Table 7.39: Performance of FU optimizations for instance $budget = 7$

Run	T. NGA	T. COMPASS	T. R&S	Total T.	N NGA	N COMPASS	N R&S	Total N
0	13.121	208.869	0.028	222.019	1344	14 980	0	16 324
1	9.546	88.709	0.028	98.282	1016	7055	0	8071
2	9.590	43.105	0.030	52.726	1020	3187	0	4207
3	4.624	18.898	0.001	23.523	520	1565	0	2085
4	4.927	22.222	0.000	27.149	520	1795	0	2315
5	9.457	18.762	0.000	28.219	1036	1511	0	2547
6	14.451	98.057	0.002	112.510	1008	6990	0	7998
7	4.507	13.288	0.000	17.795	520	1186	0	1706
8	10.018	197.347	0.026	207.391	1040	13 384	0	14 424
9	9.137	30.890	0.030	40.057	1004	2584	0	3588
Average	8.938	74.015	0.014	82.967	902.800	5423.700	0	6326.500

Table 7.40: Performance of SAA optimizations for instance $budget = 7$