PREVENTIVE SECURITY CONSTRAINED OPTIMAL POWER FLOW USING AN ENSEMBLED METHOD: CONSTRAINTS RELAXATION WITH ANALITYCAL OPTIMIZATION COMBINED WITH A HEURISTIC METHOD

GERSUSA

1. Team Information

Award No.	DE-AR0001090
Prime recipient	Gers U.S.A. LLC
Awarded period	From November 1st, 2108 to June 12th, 2020
Date of the report	January 16, 2020
Project title	PREVENTIVE SECURITY CONSTRAINED OPTIMAL POWER FLOW USING AN ENSEMBLED METHOD: CONSTRAINTS RELAXATION WITH ANALITYCAL OPTIMIZATION COMBINED WITH A HEURISTIC METHOD
Project director	J. Gers
Principal Investigators	D. Rodriguez
Project Team Members	T. Valencia, C. Acosta, D. Agudelo, D. Arango
Advisory Board	S. Rivera

2. Executive Summary

This report presents a decomposition methodology using constraint handling to solve the security-constrained optimal power flow (SCOPF) problem proposed in GO Competition - Challenge 1 (GOC-C1). The methodology consisted on the decomposition of the SCOPF problem into a base case problem and contingency sub-problems, using constraint handling rules to solve the complete problem in an iterative fashion. The first stage involved computing an OPF problem using the base case. The second stage dealt with the modification of the initial base case by updating some of the constraints limits according to the evaluation of potentially relevant contingencies. Along with parallel computing techniques and Interior Point Method, this methodology leads to achieve feasible solutions in a real-time framework (10 minutes) and near-optimal solutions in an off-line framework (40 minutes) for dataset provided in GOC-C1. The methodology successfully solved the proposed networks, satisfying active and reactive power re-dispatch in post-contingency scenarios. The following paragraphs describe the state of the art of different methodologies used to solve SCOPF problem. Finally, other contributions from the development of this research are listed.

A set of strategies have been proposed in literature to make the SCOPF optimization problem more tractable from a computational point of view. Some of these techniques, used in both OPF and SCOPF problems, use *linearization* and *convexification* of the optimization problem. One kind of *linearization* is solving the SCOPF problem through

the DC-OPF approximation [1]–[4]. However, the linear approximation may be unaccurate when using reactive power control variables (shunt reactance, voltage at generator buses) or under highly loaded conditions [5]. Other linearization techniques act directly on the objective function [3], and others include the Successive Linear Programming (SLP) method [6].

Both the OPF and SCOPF are not convex problems. Therefore, it is not possible to ensure a global minimum through mathematical programming [7]. Different attempts have been made to reach solutions close to the global minimum through genetic (GA), metaheuristic (MA) and machine learning algorithms. Strategies such as earthworm optimization algorithm, firefly algorithm tuned through fuzzy logic and approaches based on historical data have been presented in [8]–[10] to solve OPF. However, the size of the networks analyzed exceed no more than 300 buses and the number of iterations to reach the solution through this strategies cannot be ensured.

Decomposition strategies have been proposed that allow large problems to be divided into subproblems to solve them with parallel computing. The most commonly used algorithms include: Augmented Lagrangian Method (ALM), Alternating Direction Multipliers Method (ADMM), and Benders Decomposition (BD).

ALM in [11] was used to solve a distributed OPF, while in [12] it was used to solve the reactive OPF from network splitting. However, its application to the SCOPF has not been implemented yet. ADMM has been widely used because it allows the total problem to be divided and makes it parallelizable and easy to implement. This method was used in [3], [11], [13] to solve an OPF. The ADMM was also implemented in [14] to solve a C-SCOPF by testing a set of networks of up to 3,012 buses. However, only four contingencies were evaluated and it took 3,582 seconds to reach the solution.

BD was presented in [15] as an appropriate methodology to divide the C-SCOPF problem, but it was applied only for a 6-bus network. In [16], [17], a 118-bus and 2,351-bus systems were validated, but using a DC model of power flow equations. In [14] BD was also used in a 3,012-bus network but it considered only 4 contingencies and took 1,165 seconds to reach the solution.
Several works have tried to solve non-convex problems through a hybrid optimization strategy. For example, the OPF problem in [7] was solved by using GA to group the chromosomes in a search space close to the absolute minimum, and then a continuous Newton-Rhapson method was used to mathematically reach the global minimum. However, overload constraints were not considered. In [18], ALM and ADMM were used to solve the SCOPF in DC. In [19] BD was used to solve PC-SCOPF and EA to select the relevant contingencies; however, only a 118-bus system was tested.

Another strategy widely used to reduce the size of SCOPF is contingency filtering [5], [20]. For example, in [21] the umbrella contingencies method was used to select the most relevant contingencies taking into account the magnitude of the lagrangian multipliers associated with the post-contingency balance constraints. The disadvantage of this approach is that it is necessary to solve the SCOPF first, which makes it infeasible for real time. In [22], a method for contingency evaluation in real time was proposed using weighted digraphs and the central eigenvector of the Laplacian matrix. However, the Laplacian matrix was filled based on the number of overloads caused by the outaged lines, which took too long for a real time approach. EA was used in [19] for contingencies filtering, but it demands a full iteration to identify insecure contingencies. In [6], vulnerability and critically measures were used to select the relevant contingencies in the problem. However, it is necessary to evaluate a power flow for each contingency to carry out this selection, so it makes it infeasible for real time.

After trying different strategies, the decomposition of SCOPF problem and the update of constraints limits in an iterative way was selected as the final strategy to solve the networks proposed in GOC-C1. Other contributions derived from the application of this strategy are listed below:

- The methodology solves a power flow problem in each contingency, therefore , it might be faster than other

decomposition approaches that demands solving a optimization problem.

- A novel algorithm to active and reactive power re-dispatch is employed to compute a post contingency state.
- An algorithm is proposed to solve iteratively the SCOPF by modifying the constraint limits of the base case.

3. Project Description

The SCOPF problem was addressed in this research through a novel combination of problem decomposition and constraint handling rules in an iterative approach. The following subsections describe the approach originally proposed and the set of improvements included in the final version of the algorithm.

3.1. Original hypothesis

The technical approach originally proposed for the solution of SCOPF problem involved two methods: (i) an analytical optimization approach applied to the problem, and (ii) a heuristic algorithm applied to the solution obtained in the first stage using the original constraints. The flowchart proposed in Figure 1 aimed to solve the complexities in the formulation of SCOPF by using algorithms usually found as solutions to less complicated problems. That is to say, the original optimization problem was decomposed in two subproblems addressed iteratively.

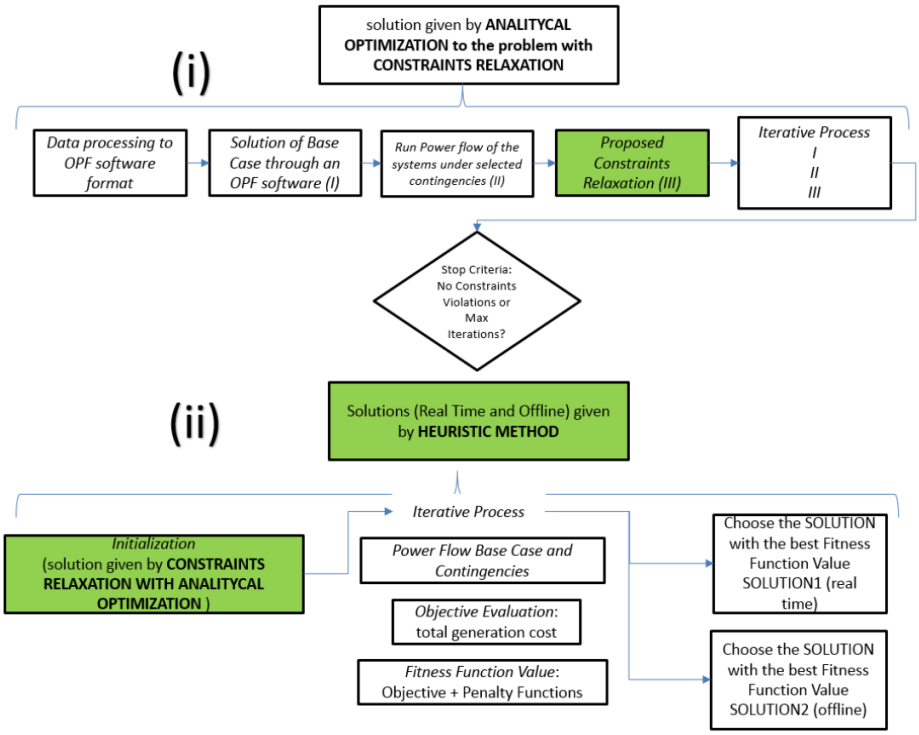

Figure 1 – Originally proposed approach.

Figure 2 shows the iterative processes mentioned in Figure 1. The first subproblem (A) was the solution of a traditional OPF (with modified constraints after the first iteration) and the second one (B) was the solution of a traditional power flow for selected contingencies. The results obtained in B were expected to define how to modify the constraints in problem A's next iteration. A and B were used in method (i) and B in method (ii).

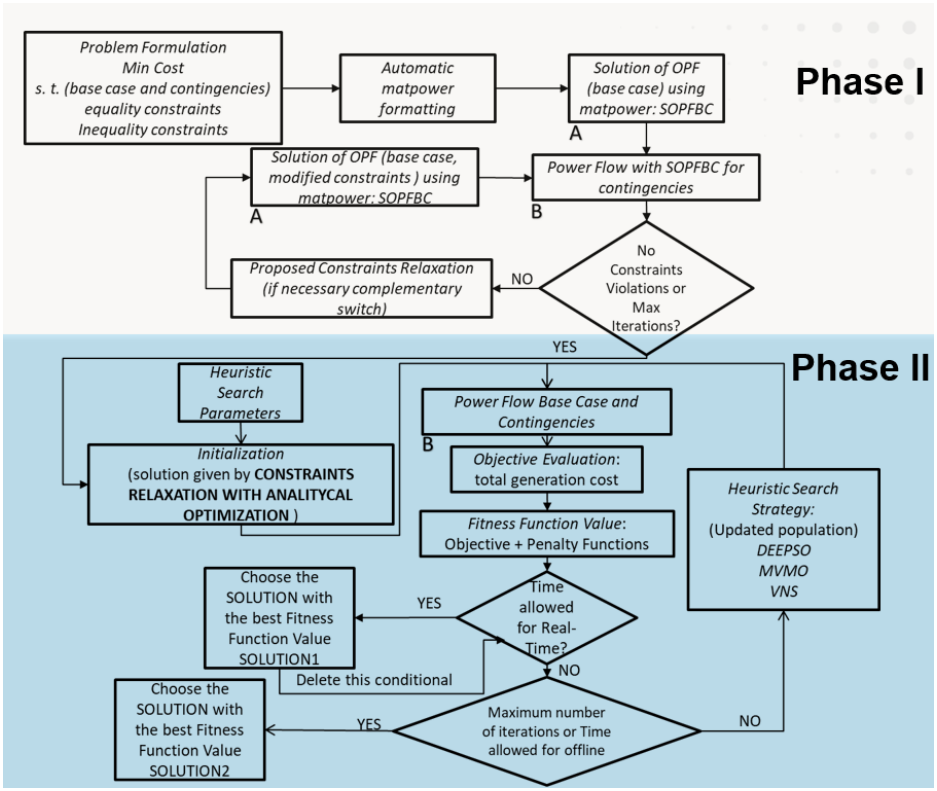

Figure 2 – Detailed flow chart for originally proposed approach.

The main purpose of phase (i) was to get the initial state of the network assessed and to achieve a feasible (or nearly feasible) solution. The general structure of this approach consisted of three basic steps: running an OPF algorithm, evaluating contingencies and storing eventual constraints violations, and including results as new or modified constraints. The constraints of active and reactive power were addressed in an iterative fashion (not directly built into the OPF model). Once a first solution was delivered (*solution1.txt*), the algorithm tried to improve the obtained solution (i.e. find a solution closer to the optimal) by means of heuristic algorithms.
This iterative approach allowed the algorithm to have good performance in small scale networks (with less than 1000 buses). Nevertheless, this algorithm presented a set of performance and feasibility issues when it was tested with large scale networks and an extensive list of contingencies:

- Unsolved voltage violations and non-Convergences under contingency evaluation.
- Execution time of heuristic search algorithms did not comply with time constraints given by GOC-C1 competition.

3.2. Final algorithm

Due to the problems founded with the methodologies summarized in Figures 1 and 2, the latest version of the algorithm included the changes listed below:

- Switched shunts as reactive generators: The inclusion of switched shunts as reactive compensators made it possible to solve most of the voltage violations in the networks assessed. They were modeled as generators with null active power limits.
- Area spin reserve constraint: This section added a linear constraint to the optimization problem to guarantee power reserve in case of contingency of the generator with maximum capacity.
- Contingencies screening: The complete list of contingencies usually took too long to be evaluated within GOC-C1 time constraints. Selection of contingencies was therefore implemented in medium and large scale networks (above 1000 buses) in order to cope with time constraints of GOC-C1 competition.
- Finite State Machine algorithm (FSM): Based on results obtained from the evaluation of the selected contingencies the FMS stage made it possible to restrict the base case in order to solve overloads, voltage violations, and non-convergences in a systematical and iterative way.
- Parallel execution of scripts: **Load Data**, **Run OPF**, **Select Contingencies**, and **Run selected Contingencies** blocks in Figure 3 were sets of scripts, each one executed in parallel.

An algorithm based on the Matpower toolbox [23] and the Interior Point Optimizer ([24], [25]) was proposed to solve the SCOPF problem [26], [27]. The algorithm consisted on *Pre- and Post-processing* stages (input and output data) and other stages classified into three groups: *Parallel OPF*, *Contingencies evaluation*, and *Constraints Handling*. Figure 3 shows the approach used for updating the constraint limits and solve the SCOPF problem in a iterative way. The bucle is stopped by two criteria: when the penalty cost is lower than a determined percentage of the objective function (e.g 5%) or when a number of iterations have been reached (e.g 40) .

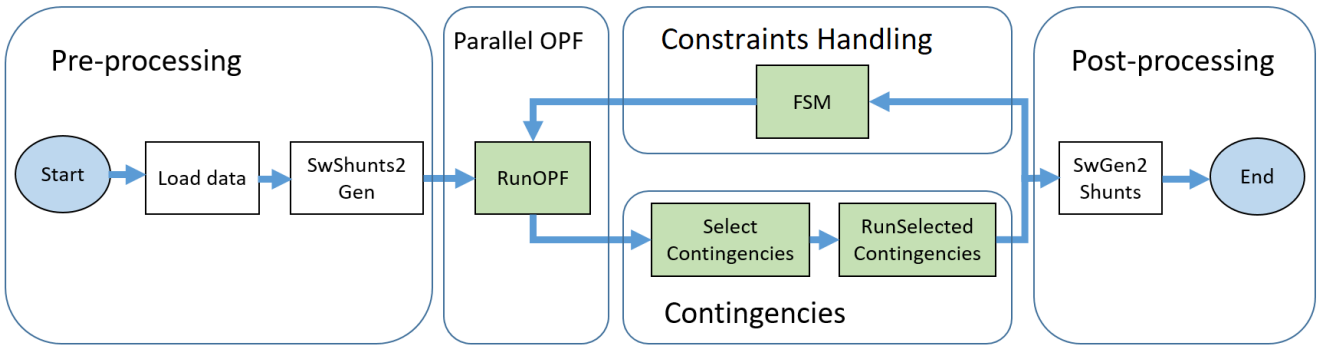

Figure 3 – Overview of main solution algorithm.

The **RunOPF** block was executed in parallel and is summarized in Figure 4. The *Load parallel seeds* function set different configuration parameters for IPOPT solver. The initial conditions of optimization variables, the linear solver used (like MUMPS, MA57, or MA86), and tolerance levels were combined in multiples workers due to their influence in the convergence time.

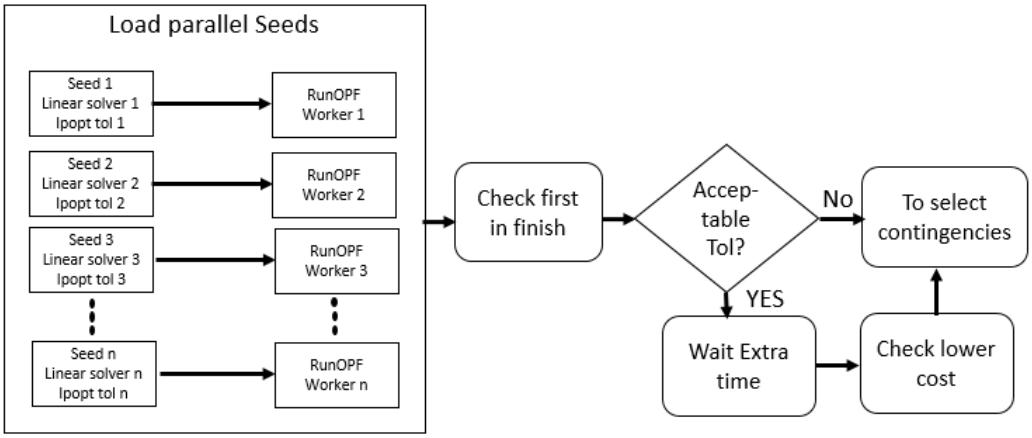

Figure 4 – Parallel optimal power flow.

The different seeds were selected considering random values according to different strategies. Some workers used seeds according to uniform distributions varying between upper and lower variables limits. Others used the variables values of last iteration or half point between upper and lower boundaries values.
On the other hand, the IPOPT tolerances were modified to make the solution sub-optimal in terms of cost (acceptable configuration), but feasible, with the advantage of lower convergence time in relation to another parameters configuration.
If an acceptable configuration was the first to finish, extra time was given to find better solutions. After that extra time had ended, the solution with the lowest cost was chosen to continue towards the *select contingencies* stage

The contingencies screening was used as a speed-up strategy for medium and large scale networks (i.e. more than 1000 buses). The methodology for such selection consisted on a set of sorted lists that reflected the likeliness of each contingency to develop overloads, voltage violations and non-convergences. The strategy was used separately for branches and contingencies, as depicted in Figure 5.

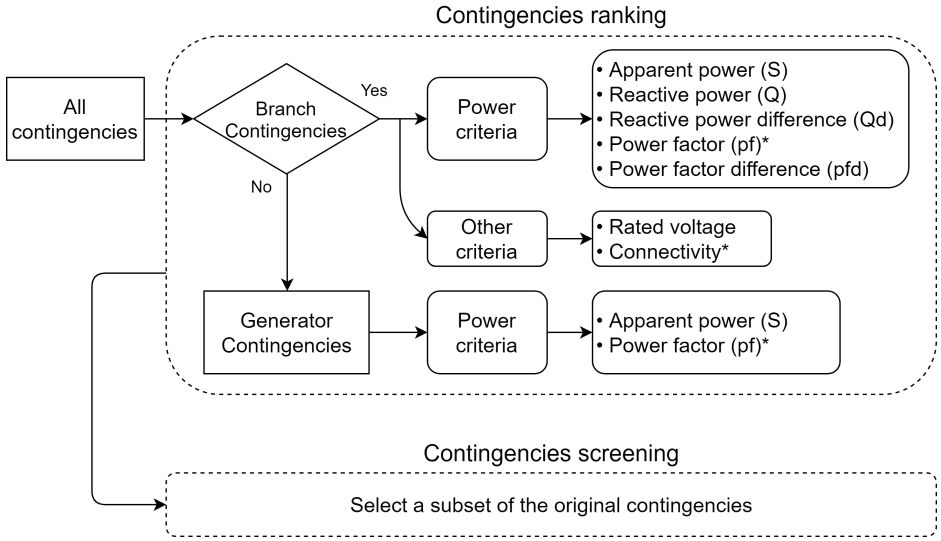

Figure 5 – Contingencies selection flowchart.

The results from the **RunOPF** block was the main indicator for such selection in both branch and generator contingencies. Screening for branch contingencies required, however, the use of complementary indicators such as the rated voltage and the connectivity (amount of branches connected to the same bus). Once the subset of contingencies was estimated, the evaluation stage continued.

Different workers simultaneously executed the steps summarized in Figure 6 through parallel computation. First, the contingency element is outaged and conventional power flow is executed with the same load and generated active power of the base case. In this step a slack bus is chosen by taking into account the affected area and the generator with maximum spin reserve.

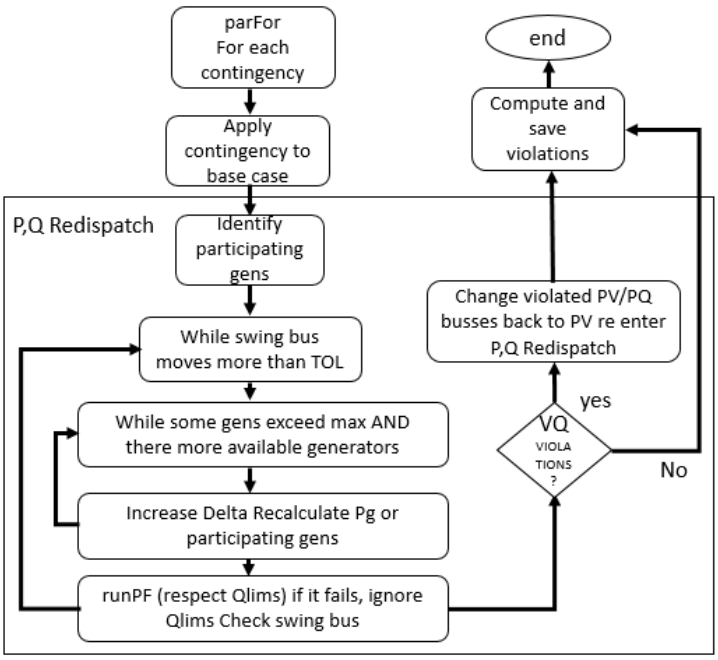

Figure 6 – Parallel optimal power flow.

The difference between total active power generated in the base case and the result of power flow is a first ΔP (Delta) estimation. According to this value, the set points of active power in post contingency are adjusted taking into account the participation factors. If a generator tried to exceed its limit, it is saturated and the ΔP value is increased through a normalized participating factor neglecting saturated generators.

The differences of active power between the programmed value and power flow result are due to QV violations in some generators. To fix the reactive power violations in PV buses these are converted to PQ to satisfy the re-dispatch of reactive power. Nevertheless, this lead to changes in voltage magnitude in PQ buses and leads power losses system . According to slack deviation, the delta value is recalculated and the power flow is executed again until reach a minimal deviation.

Finally the Finite State Machine (FSM) stage is shown in Figure 7, it was created to solve the overloads, voltage violations and the non-convergences issues identified by the *Contingencies* stage. After the set of selected contingencies is evaluated and the summary of largest violations is established (overloads, voltage violations, non-convergences), the FSM updated the boundaries of constraints for the next iteration. By doing so, the OPF of next iteration used a base case with updated constraints. The constraints handling is dependent of present state in the FMS.

Table 1 shows a summary of the main activities performed during the project.

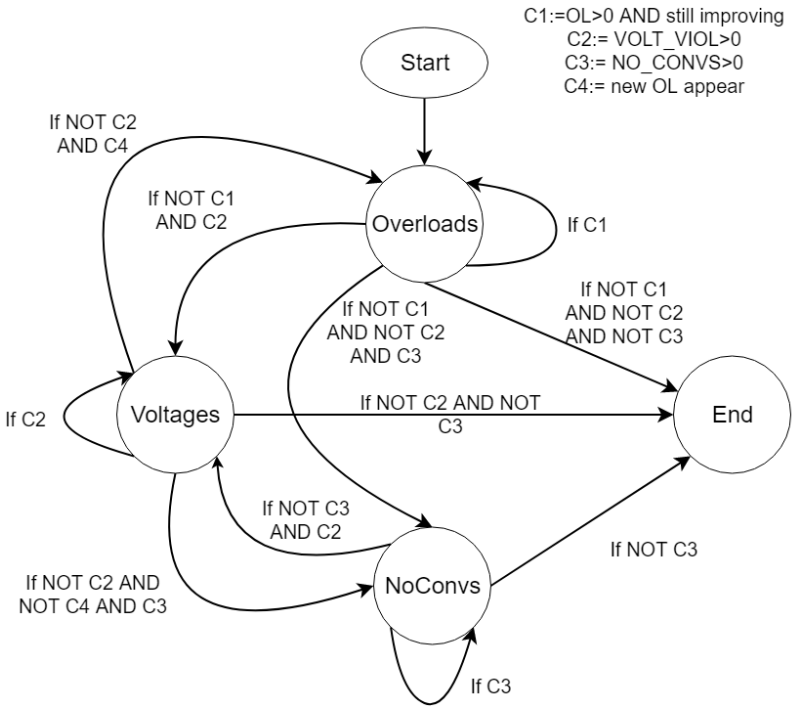

Figure 7 – FSM transitions.

Key activities	Results
1. Adaptation of data formats	The *.con, .rop, .raw* and *.inl* files were adapted to Matpower syntax.
2. Test the competition platform	All versions of scripts were successfully adapted to the GOC platform hardware during the whole project.
3. Extension of formulation of OPF	The standard OPF formulation in Matpower did not include the GOC formulation for the base case. Soft-limits, commutable shunt constraints and area spin reserve equations was added.
4. Solvers (PF and OPF) selection	IPOPT was selected as the default solver for the optimization problem. Different linear solvers were compiled to test the best performance in terms of time.
5. Algorithm constraints redefinition	A Finite State Machine was coded for solving overloads, voltage violations, and non-convergences issues in the networks assessed. This FSM redefined initial constraints of network base case in order to achieve a feasible solution, including n-1 contingencies.
6. Tuning of heuristic parameters	Heuristic algorithms were not further developed after Trial 1 due to GOC-C1 time constraints. Instead, the performance of phase (i) algorithm was improved to obtain feasible solutions through iterative execution.
7. Software acceleration	Improvement of scripts performance were successfully achieved by using mainly three strategies: contingency screening, parallel execution of scripts, and reduction in the overall data transfer.
8. Tuning of finite state machine	The constraints update by means of empirical rules proved to be faster and more efficient with respect to heuristic algorithms. Each state of FSM was tuned separately, several tests validated the effectiveness of the proposed configuration.

Table 1 – Summary of main activities developed during the GOC-C1 competition.

3.3. Problems encountered in final algorithm

Regarding the proposed methodology, problems were found in *Run Selected Contingencies* stage (see Figure 3) when evaluating post-contingency power flows. The idea that N workers allow to reduce the calculation time by the same factor is not true when the time of sending and receiving information between client and the servers is comparable with the power flows time.
This problem becomes more evident when an architecture with a single server and multiple clients is used, because the server has all the burden of sending and receiving the data and therefore the flow of information slows down. This limitation made the calculation of contingencies faster in practice when using 72 workers with respect to when 144 were used (for *solution1* approach). Even to create solution 2 it was better to use 24 workers when the networks exceeded 20,000 busses.

4. Products Identified

- Publications:

  T. Valencia, D. Agudelo-Martinez, D. Arango, C. Acosta, S. Rivera, D. Rodriguez, and J. Gers, "A fast decomposition for solving a Security-Constrained Optimal Power Flow (SCOPF) problem through constraint handling", 2019. Under revision.

- Techniques:

  – Contingencies selection according to seven different criteria
  – Constraints handling through a Finite State Machine.

5. Computer modeling

5.1. Performance criteria

The performance criterion of the algorithm was focused on minimizing the number of violations of the system which implies a reduction in the cost of the objective function. Each stage of the algorithm has it own performance criteria. For the *parallel OPF* stage, the performance criterion to choose the best solution was the convergence time. In case of having a solution coming from tolerance extended, the algorithm wait more solutions for a limited time to compare and choose the solution with the lowest cost.
In the *select contingencies* stage, the criteria for determining the percentage of selection was the remaining time of the algorithm, that is, a certain amount of contingencies was evaluated estimating that it was possible at least to perform another execution of the OPF.
In the finite state machine, several tests showed that for small scale networks it was better to start with **overloads** as the initial state. On the other hand for networks with more buses it was better to start with **voltages**. The performance criteria to choose this initial states was the reduction in penalization cost in function of the iterations of the algorithm.

5.2. Mathematical model

5.2.1 Problem formulation

The SCOPF problem [26] is focused on minimizing the total cost C_{tot}:

$$min(C_{tot}) = min\left[(\sum_{g\varepsilon G} c_g) + \delta c^{\sigma} + \frac{1-\delta}{|k|}\sum_{k\varepsilon K} c_k^{\sigma}\right] \tag{1}$$

Where G is the set of generators, c_g is the generation cost of generator g, c^{σ} is the total constraint violation penalty in base case and c_k^{σ} is the total constraint violation penalty in contingency k. K is the set of all contingencies. δ is a weight assigned to the penalty cost in the base case.
For all constraints, sc denote a particular scenario of the set SC:

$$SC = \{0,1,2,3,...,k-1,k\} \tag{2}$$

where $sc = 0$ means the base case and $sc = i$ with $i > 0$ denote the i^{th} scenario, which corresponds when the i^{th} contingency is applied.
The constraints associated to objective function are:

$$\underline{v_i} \leq v_i \leq \overline{v_i} \quad \forall i \; \varepsilon \; I^{sc} \; \wedge \forall sc \; \varepsilon \; SC \tag{3}$$

where I^{sc} is the set of active buses in the scenario sc. v_i is the voltage magnitude on bus i.

$$\underline{p_g} \leq p_g \leq \overline{p_g} \quad \forall g \; \varepsilon \; G^{sc} \; \wedge \forall sc \; \varepsilon \; SC \tag{4}$$

$$\underline{q_g} \leq q_g \leq \overline{q_g} \quad \forall g \; \varepsilon \; G^{sc} \; \wedge \forall sc \; \varepsilon \; SC \tag{5}$$

where G^{sc} is the set of active generators in the scenario sc. p_g and q_g are the active and reactive power of generator g.
The transmission lines overload constraints considered were:

$$\sqrt{(p_e^o)^2 + (q_e^o)^2} \leq \overline{R}_e v_{i_e}^o + \sigma_e^{sc,s} \quad \forall e \; \varepsilon \; E^{sc} \; \wedge \forall sc \; \varepsilon \; SC \tag{6}$$

$$\sqrt{(p_e^d)^2 + (q_e^d)^2} \leq \overline{R}_e v_{i_e}^d + \sigma_e^{sc,s} \quad \forall e \; \varepsilon \; E^{sc} \; \wedge \forall sc \; \varepsilon \; SC \tag{7}$$

$$\sigma_e^{sc,s} \geq 0 \; \forall sc \; \varepsilon \; SC \tag{8}$$

where in p_e^o , q_e^o , p_e^d and q_e^d are the active (p) and reactive (q) power in the e line extremes. The upper index means the origin (o) or destiny (d) bus of the line e. E^{sc} is the set of active transmission lines in the scenario sc. $v_{i_e}^o$ and $v_{i_e}^d$ are the voltage magnitudes in the origin and destiny buses respectively. $\overline{R}_e$ is the line e maximal apparent current. $\sigma_e^{sc,s}$ is the upper bound penalty for line current rating violation in the origin and destiny buses in the scenario sc.
The transformers overload constraints considered were the following:

$$\sqrt{(p_f^o)^2 + (q_f^o)^2} \leq \overline{s}_f + \sigma_f^{sc,s} \quad \forall f \; \varepsilon \; F^{sc} \; \wedge \forall sc \; \varepsilon \; SC \tag{9}$$

$$\sqrt{(p_f^d)^2 + (q_f^d)^2} \leq \overline{s}_f + \sigma_f^{sc,s} \quad \forall f \; \varepsilon \; F^{sc} \; \wedge \forall sc \; \varepsilon \; SC \tag{10}$$

$$\sigma_e^{sc,s} \geq 0 \; \forall sc \; \varepsilon \; SC \tag{11}$$

$$\sigma_f^{sc,s} \geq 0 \; \forall sc \; \varepsilon \; SC \tag{12}$$

where in p_f^o , q_f^o , p_f^d and q_f^d are the active (p) and reactive (q) power in the f transformer extremes. The upper index means the origin (o) or destiny (d) bus of the transformer f. F^{sc} is the set of transformers in the scenario sc . The apparent power rating is defined as $\overline{S}_f$ for the f transformer. $\sigma_f^{sc,s}$ is the upper bound penalty for the f transformer in the scenario sc.
The commutable shunts were modeled as generators with null real power. The reactive power constraints were considered as:

$$\underline{b_i^{cs}} v_i^2 \leq q_i^{cs} \leq \overline{b_i}^{cs} v_i^2 \quad \forall i \; \varepsilon \; I^{sc} \; \forall sc \; \varepsilon \; SC \tag{13}$$

where q_i^{cs} and b_i^{cs} are the reactive power and the susceptance value of commutable shunt on bus i respectively (if it

has one). v_i is the voltage magnitude on the same bus.
The power balance constraints considered were:

$$\sum_{g \varepsilon G_i^{sc}} p_g - p_{L_i}^{sc} - g_{fs_i}^{sc} v_i^2 - \sum_{e \varepsilon E_i^{sc,o}} p_e^o - \sum_{e \varepsilon E_i^{sc,d}} p_e^d - \sum_{f \varepsilon F_i^{sc,o}} p_f^o - \sum_{f \varepsilon F_i^{sc,d}} p_f^d = \sigma_i^{sc,P+} - \sigma_i^{sc,P-} \; \forall sc \; \varepsilon \; SC \tag{14}$$

$$\sigma^{sc,P+} \geq 0 \; \forall sc \; \varepsilon \; SC \tag{15}$$

$$\sigma^{sc,P-} \geq 0 \; \forall sc \; \varepsilon \; SC \tag{16}$$

for active power, and

$$\sum_{g \varepsilon G_i^{sc}} q_g - q_{L_i}^{sc} - (-b_{fs_i}^{sc} - b_{cs_i}^{sc}) v_i^2 - \sum_{e \varepsilon E_i^{sc,o}} q_e^o - \sum_{e \varepsilon E_i^{sc,d}} q_e^d - \sum_{f \varepsilon F_i^{sc,o}} q_f^o - \sum_{f \varepsilon F_i^{sc,d}} q_f^d = \sigma_i^{sc,Q+} - \sigma_i^{sc,Q-} \; \forall sc \; \varepsilon \; SC \tag{17}$$

$$\sigma_i^{sc,Q+} \geq 0 \; \forall sc \; \varepsilon \; SC \tag{18}$$

$$\sigma_i^{sc,Q-} \geq 0 \; \forall sc \; \varepsilon \; SC \tag{19}$$

for reactive power.
where $p_{L_i}^{sc}$ and $q_{L_i}^{sc}$ are the active and reactive load power on bus i in the scenario sc. $g_{fs_i}^{sc}$ and $b_{fs_i}^{sc}$ are the conductance and susceptance of the fixed shunts on bus i in the scenario sc. $E^{sc,o}$ and $E^{sc,d}$ are the set of active lines in the scenario sc in the origen and destiny buses. $F^{sc,o}$ and $F^{sc,d}$ are the set of active transformers in the scenario sc in the origen and destiny buses.
$\sigma_i^{sc,P+}$ and $\sigma_i^{sc,P-}$ ($\sigma_i^{sc,Q+}$ and $\sigma_i^{sc,Q-}$) are the positive and negative parts of the violation of active (reactive) power balance for bus i in the scenario sc.
An area spin reserve constraint was considered to prevent the lack of power when generation contingencies occur.

$$\sum_{g \varepsilon A} (\overline{P}_g - P_g) + \sigma_{A,sc} \geq max_{g \varepsilon (\chi_{sc} \bigcap A)} \overline{P}_g \tag{20}$$

where χ_{sc} is the set of generators that appear in a contingency of scenario sc and $\sigma_{A,sc}$ is an area spin reserve slack variable for each affected area A in the sc contingency.
The penalization cost of 1 for the scenario sc is computed according to slack variables as follows:

$$C_{sc}^{\sigma} = \sum_{i \varepsilon I} \alpha_p (\sigma_i^{sc,P+} - \sigma_i^{sc,P-}) + \sum_{i \varepsilon I} \alpha_q (\sigma_i^{sc,Q+} - \sigma_i^{sc,Q-}) + \sum_{f \varepsilon F} \alpha_{ef} (\sigma_f^{sc,s}) + \sum_{e \varepsilon E} \alpha_{ef} (\sigma_e^{sc,s}) + \alpha_{\sigma} \sigma_{A,sc}$$

According to secondary frequency regulation in multi-area systems, each control area should be able to recover the frequency according to participation factors predefined by network operator [28]. After the contingency has occurred, in steady state the active power generated must satisfy :

$$P_g^k = \begin{cases} \overline{P_g} & \text{if } P_g^0 + a_{pf} \Delta P \leq \overline{P_g} \\ \overline{P_g^0} + a_{pf} \Delta P & \text{if } \overline{P_g} < P_g^0 + a_{pf} \Delta P < \overline{P_g} \\ \overline{P_g} & \text{if } \overline{P_g^0} + a_{pf} \Delta P \geq \overline{P_g} \end{cases} \tag{21}$$

where P_g^k is the vector of active power generated in the affected area to the the k contingency. a_{pf} is the vector of participation factors. P_g^0 is the vector of active power in the base case, ΔP is the difference of real generation between the base case and the contingency.
The reactive power redispatch is performed according to voltage stability criteria. In principle the voltage control

tries to maintain the pre-fault voltage magnitude unless reactive limits are violated [29]:

$$\begin{cases} V_g^k = V_g^0 & \text{if } \underline{Q_g} \leq Q_g^k \leq \overline{Q_g} \\ \underline{V_g} < V_g^k < V_g^0 & \text{if } \overline{Q_g^k} = \overline{Q_g} \\ V_g^0 < V_g^k < \overline{V_g} & \text{if } Q_g^k = \underline{Q_g} \end{cases} \tag{22}$$

where Q_g^k is the reactive power generated vector in the contingency k, V_g^0 and V_g^k are voltage magnitude vectors in the generators in contingency k and in the base case, respectively.

5.2.2 Proposed approach

Considering the set of constraints corresponding to each contingency is a computationally demanding problem when a huge amount of contingencies are to be evaluated.
In order to simplify the complete SCOPF problem, the methodology applied pretended to cover the post-contingency restrictions by means of constraints handling of the base case. That to say, this methodology only employs optimization for the base case subject to constraints only considering sc=0, that is, without considering c_k^σ terms in Equation 1, neither constraints with $sc > 0$.
According to the latter reasoning, the objective function and constraints equations for the base case in the m iteration is given by minimize :

$$C = (\sum_{g \varepsilon G} C_g + \delta C^\sigma) \tag{23}$$

The lower and upper bounds of the variable x for the iteration m is denoted respectively as $\underline{x}(m)$ and $\overline{x}(m)$. The constraints considered were:

$$\underline{v_i}(m) \leq v_i \leq \overline{v_i}(m) \quad \forall i \; \varepsilon \; I \tag{24}$$

$$\underline{p_g}(m) \leq p_g \leq \overline{p_g}(m) \quad \forall g \; \varepsilon \; G \tag{25}$$

$$\underline{q_g}(m) \leq q_g \leq \overline{q_g}(m) \quad \forall g \; \varepsilon \; G \tag{26}$$

$$\sqrt{(p_e^o)^2 + (q_e^o)^2} \leq \overline{R}_e(m) v_{i_e}^o + \sigma_e^s \quad \forall e \; \varepsilon \; E \tag{27}$$

$$\sqrt{(p_e^d)^2 + (q_e^d)^2} \leq \overline{R}_e(m) v_{i_e}^d + \sigma_e^s \quad \forall e \; \varepsilon \; E \tag{28}$$

$$\sqrt{(p_f^o)^2 + (q_f^o)^2} \leq \overline{s}_f(m) + \sigma_f^s \quad \forall f \; \varepsilon \; F \tag{29}$$

$$\sqrt{(p_f^d)^2 + (q_f^d)^2} \leq \overline{s}_f(m) + \sigma_f^s \quad \forall f \; \varepsilon \; F \tag{30}$$

$$\underline{b_i^{cs}} v_i^2 \leq q_i^{cs} \leq \overline{b_i}^{cs} v_i^2 \quad \forall i \; \varepsilon \; I \tag{31}$$

$$\sum_{g \varepsilon G_i} p_g - p_i^L - g_i^{FS} v_i^2 - \sum_{e \varepsilon E_i^0} p_e^o - \sum_{e \varepsilon E_i^d} p_e^d - \sum_{f \varepsilon F_i^o} p_f^o - \sum_{f \varepsilon F_i^d} p_f^d = \sigma_i^{P+} - \sigma_i^{P-} \tag{32}$$

$$\sum_{g \varepsilon G_i} q_g - q_i^L - (-b_i^{FS} - b_i^{CS}) v_i^2 - \sum_{e \varepsilon E_i^0} q_e^o - \sum_{e \varepsilon E_i^d} q_e^d - \sum_{f \varepsilon F_i^o} q_f^o - \sum_{f \varepsilon F_i^d} q_f^d = \sigma_i^{Q+} - \sigma_i^{Q-} \tag{33}$$

$$\sum_{g \varepsilon A} (\overline{P}_g(m) - P_g) + \sigma_A \geq max_{g \varepsilon (\chi \cap A)} \overline{P}_g \tag{34}$$

The penalization cost for the base case is computed according to slack variables as follows:

$$C^{\sigma} = \sum_{i \varepsilon I} \alpha_p(\sigma_i^{P+} - \sigma_i^{P-}) + \sum_{i \varepsilon I} \alpha_q(\sigma_i^{Q+} - \sigma_i^{Q-}) + \sum_{f \varepsilon F} \alpha_{ef}(\sigma_f^s) + \sum_{e \varepsilon E} \alpha_{ef}(\sigma_e^s) + \alpha_{\sigma}\sigma_A \tag{35}$$

Where α_p, α_q, α_{ef} and α_{σ} are penalization functions. Each of these penalization functions satisfies that:

$$\alpha(x) = \begin{cases} k_1|x| & \text{if } 0 \leq |x| \leq x_1 \\ k_2|x| & \text{if } x_1 < |x| < x_2 \\ k_3|x| & \text{if } |x| \geq x_2 \end{cases} \tag{36}$$

The last optimization problem can be summarized as:

$$\min f(x) \tag{37}$$

subjet to:

$$\mathbf{g(x)} = 0 \tag{38}$$

$$\mathbf{h(x)} \leq B(m) \tag{39}$$

The interior point solver allows remove the inequality constraint using the barrier function, the problem is:

$$minf(x) - \gamma \sum_{m \varepsilon N_{ineq}} log(Z_m) = 0 \tag{40}$$

$$\mathbf{g(x)} = 0 \tag{41}$$

$$\mathbf{h(x)\text{-}B(m)} + Z = 0 \tag{42}$$

$$Z \geq 0 \tag{43}$$

So the Lagrangian function is given by:

$$L(x,Z,\lambda,\mu) = f(x) - \gamma(\sum_{m \varepsilon N_{ineq}} Log(Z_m)) + \lambda^T g(x) + \mu^T(h(x) - B(m) + Z) \tag{44}$$

The base case solution is given by the solution to:

$$\nabla L(x,Z,\lambda,\mu) = 0 \tag{45}$$

The ipopt solver uses Newton Rhapson's method to solve the previous equation, therefore it is necessary to calculate the derivative of each term in the previous equation, which implies calculating the Hessian of objective function and each constraint.

5.3. Methodology

The update rules was tuned in function of FSM present state (see Fig 7). These rules affect the $B(m)$ term in 44. If the present state is **overloads**, the rules to update limits is shown in the figure 10. Branch is used to refer both transmission lines and transformers.

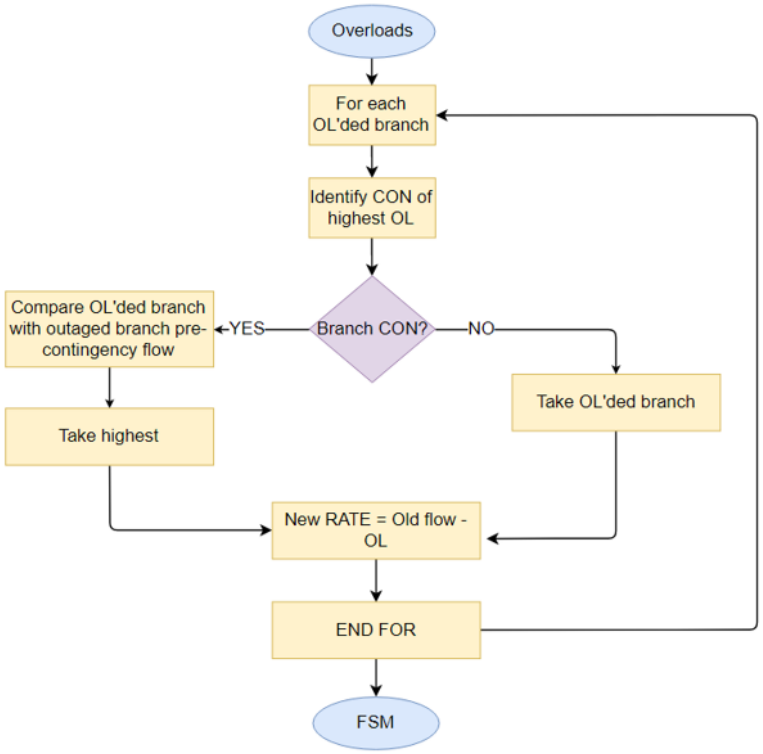

Figure 8 – Branch limits updating.

If the present state is **Voltages**, the limits update rules are shown in the figure 9.

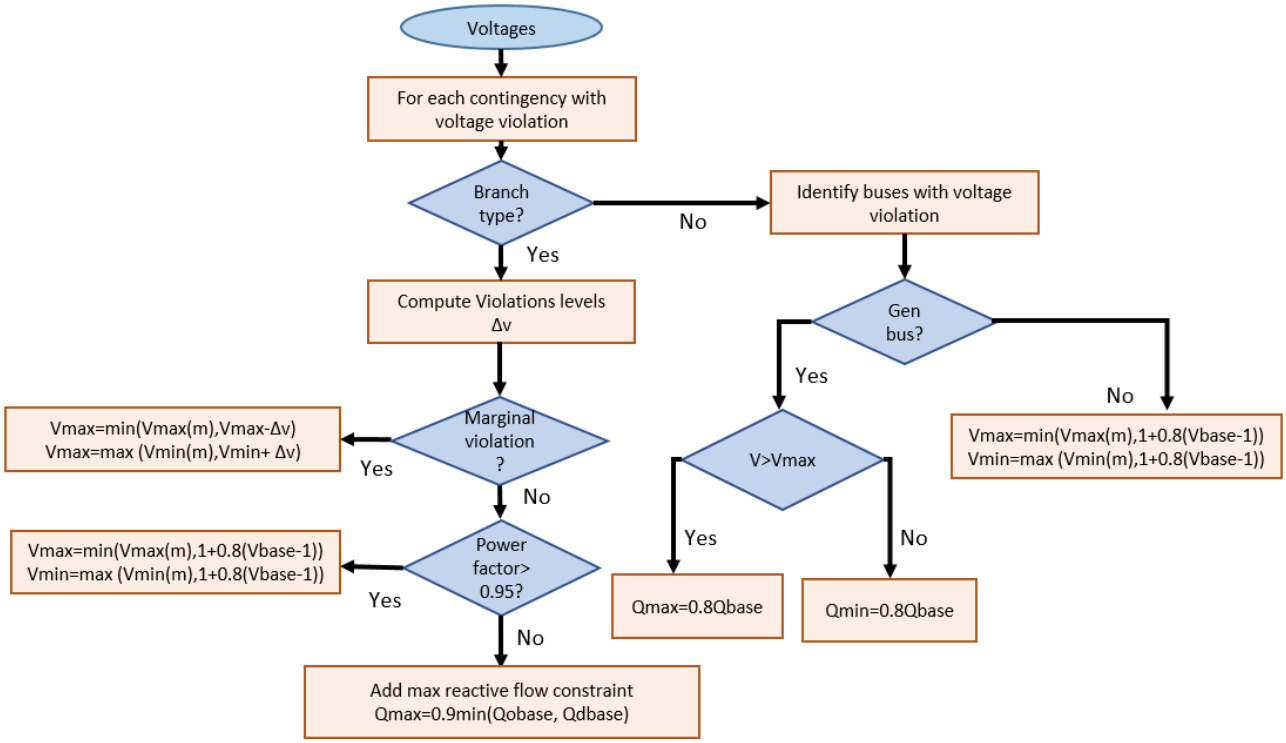

Figure 9 – Voltage limits updating.

In this case, marginal violation are defined as violations that satisfied:

$$((V_k - V_{min}) < 0.125)||(V_{max} - V_k) < 0.125)) \wedge |V_{base} - V_{lim}| < 0.01 \tag{46}$$

The power factor of complex power in the destination or origin extreme was validated as another criteria to update the voltage limit, in this case the voltage can be update similarly both undervoltages and overvoltages.
If the marginal voltage or power factor criteria is not satisfied, a new set of constraints is added in the case base:

$$q_f^d < q_{f_{max}} \tag{47}$$

$$q_f^o < q_{f_{max}} \tag{48}$$

If is a transformer contingency, and if is a line contingency:

$$q_e^d < q_{e_{max}} \tag{49}$$

$$q_e^o < q_{e_{max}} \tag{50}$$

Finally, if the present state is **non convergence** , the strategy is reduce the upper real power limit when is a contingency of generation. On the other hand, when a line contingency cause a non convergence the solution is reduce $\overline{R}_e$ or $\overline{S}_f$ (rateA) in a determinated percentage.

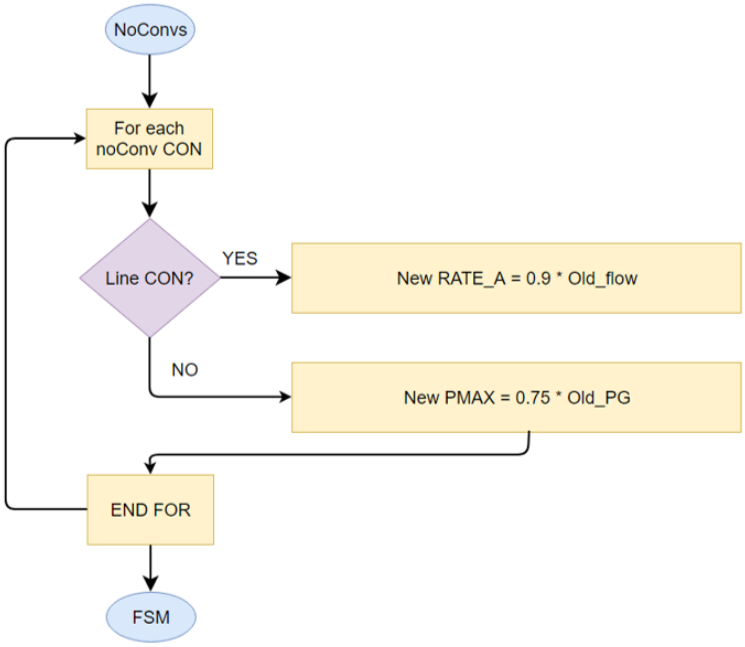

Figure 10 – Limits updating when non convergences occurs (rateA is $\bar{s}_f$ or $\bar{R}_e$ depending if is a transformer or line contingency).

5.4. Results

The algorithm proposed in this research was implemented through the Matpower toolbox [23] and IPOPT solver [25][24] to solve the SCOPF problem. For show the results of algorithm proposed, three networks had been chosen. These are described in Table 2. These datasets were taken of the Grid Optimization Competition - Challenge 1- Trial 2. Results from the evaluation of the networks listed in Table 2 are presented and discussed below. All the networks tested were run in a 64-bits Linux distribution of Matlab® 2019, Intel(R) Xeon(R) CPU E5-2680 @ 2.70GHz core i7, 128 GB RAM memory.

Table 2 – Description of networks tested.

Network	Buses	Generators	Loads	Branches	Transformers	Contingencies	Shunts	Areas
1	500	224	281	540	193	786	44	1
2	4,918	1,340	3,070	4,412	2,315	5,085	732	31
3	11,615	899	19,272	13,967	5,936	8,747	1,332	1

5.5. Parallel OPF (Optimal Power Flow)

A set of different initial points were used for the Optimal Power Flow computation. The solver used for the OPF problem was IPOPT [25]. Since the IPOPT methodology strongly depends of the initial point [30], a set of 16 different initial points were proposed to run the OPF algorithm in parallel. Each point was executed at a different CPU core.

Table 3 describes the different IPOPT parameters used to create the set of combinations. These combinations served as the initial points for IPOPT solver to perform the OPF algorithm. IPOPT executes the Interior Point Method (IPM) algorithms in an iterative fashion[24]. Parameter *Seed* was composed by two different options: Option *a* means the average of lower and upper bounds (average point) of constrained variables was selected as the initial point for all OPF iterations; option *b* (named as *Recursive warm start*) took the initial value of constrained variables from a base case network at first OPF iteration; for subsequent iterations, the values of OPF variables were updated using the data from the previous OPF iteration. On the other hand, Parameter *Initial voltage* used two options: option *a* took the values of voltages from a base case network, and option *b* set voltage at all buses in 1 p.u.

Table 3 describes the different IPOPT parameters used to create the set of combinations. These combinations served as the initial points for IPOPT solver to perform the OPF algorithm. IPOPT executes the Interior Point Method (IPM) algorithm in an iterative fashion [24]. On the one hand parameter *Seed* was composed of two different options. Option *a* refers to the average of lower and upper bounds (average point) of constrained variables and was selected as the initial point for all OPF iterations. Option *b* (named *Recursive warm start*) took the initial value of constrained variables from a base case network the first OPF iteration. For subsequent iterations, the values of the OPF variables were updated by using data from the previous OPF iteration. On the other hand, parameter *Initial voltage* used two options: option *a* took the values of voltages from a base case network; and option *b* sets voltage at all buses in 1 p.u.

Table 3 – Description of IPOPT parameters used in the set of combinations [25], [24].

	Description		
	a (*)	b	c
Linear solver	ma57	ma86	mumps
Seed	Average point	Recursive warm start	-
Initial voltage	Warm start	1 p.u.	-
Tolerances	IPOPT default settings	See Table 4	-

(*) :IPOPT default values

Table 4 – description of parameters with acceptable tolerance [25], ipopt

	options	
	a (*)	b
iterations	15	1
overall tolerance	1e-8	1e1
constrains violation	1e-2	1
dual infeasibility	1e10	1e2
complementarity infeasibility	1e-2	1e2
objective change	1e20	1.5e-1

(*) :ipopt default values

Table 5 – Parameters for combinations selected.

	Combinations															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Linear solver	a	a	a	a	a	b	b	b	b	b	c	c	c	c	c	c
Seed	a	a	b	b	b	a	a	b	b	b	a	a	b	b	b	a
Initial voltage	a	a	a	a	b	a	a	a	a	b	a	a	a	a	b	a
Tolerances	a	b	a	b	a	a	b	a	b	a	a	b	a	b	a	a

In Table 3 parameter *Tolerances* intended to relax the termination of the OPF algorithm for medium (network 2) and large (network 3) scale networks by setting six different parameters, listed in Table 4. These sets of relaxed tolerances were used to shorten the time in order to achieve a feasible solution to the OPF problem. Further information about the meaning and use of these parameters can be found in [24]. Finally, Table 5 shows the complete set of combinations used for the parallel execution of OPF algorithms.

The parallel approach to the OPF algorithm was implemented in Matpower and tested by using 7 different scenarios of each network listed in Table 2. Different scenarios of a network imply the same number of elements, but a different set of operating states, contingencies, and load profiles. Figures 11 to 13 show the maximum and average times (in p.u. respect to the maximal value), and the maximum and average costs (in p.u.) obtained from that parallel algorithm. These combinations were used as different initial points for the IPOPT solver to solve the OPF problem. In this way, the stage *Parallel OPF* shown in Figure 3 selected the fastest and feasible result from the OPF algorihtm as input for the *Contingencies* stage.

Table 6 – Summary of OPF results

Network	1	2	3
Max. cost [$]	3.93E+08	3.89E+07	2.21E+07
Max. time [s]	73.32	681.22	729.36

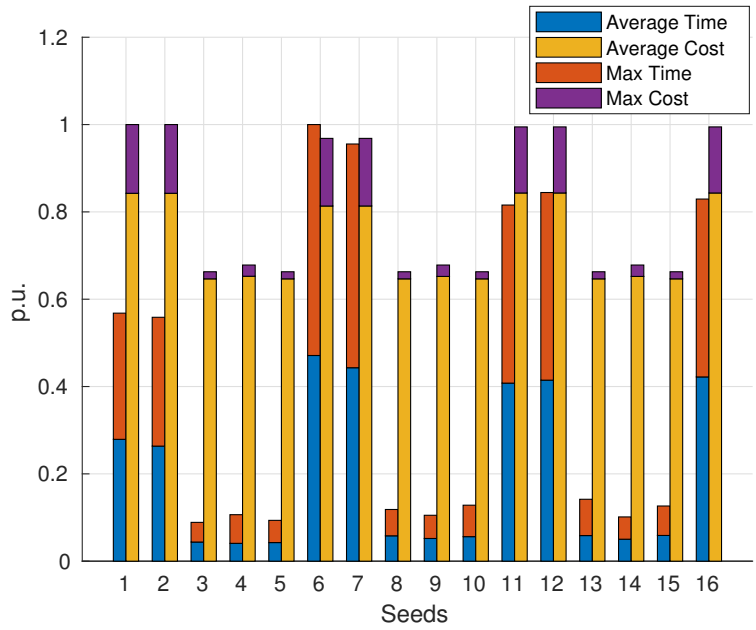

Figure 11 – Performance of Parallel OPF stage for network with 500 buses (Cost in logarithmic ordinate scale).

In general terms, the parallel execution of several and different initial points for the IPOPT solver made it possible to achieve an optimal (or sub-optimal) solution that is faster than in a serial execution approach. In the case of network 1 (Figure 11), the maximum and average costs for each combination were quite different, therefore, a logarithmic scale was used for comparison purposes:

$$Cost^* = \frac{log_{10}(C)}{log_{10}(C_{max})} \tag{51}$$

Indeed, the difference between the maximum cost from combinations 3 and 2 was about a thousandfold. This means that different initial points for the IPOPT may result in quite different results for network 1. It can be observed that combinations 3 to 5 produced in average a low mean time to achieve an optimal solution and, therefore, results from these combinations were selected as the input for the stage *Contingencies selection and evaluation*. The highest costs were obtained from combinations 1-2, 6-7, 11-12 and 16, which also corresponded to the highest maximum and mean time values.

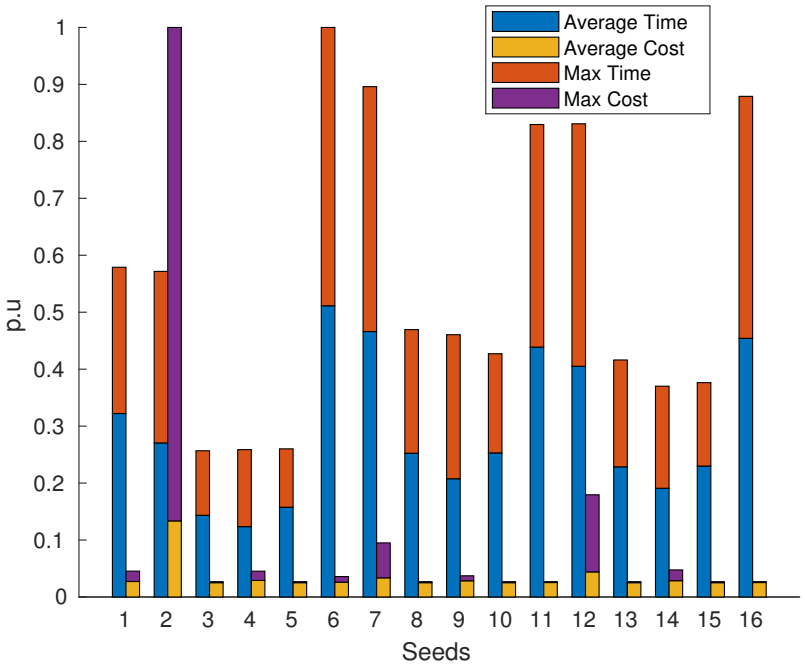

Figure 12 – Performance of Parallel OPF stage for a network of 4918 buses.

Most of the average costs were similar in the 16 combinations for network 2 (Figure 12). Combination 2 had the worst performance in this network thereby producing the highest operation cost. In terms of maximum and average times, however, the difference between different combinations was notorious. Therefore, the OPF problem was best solved in most of the scenarios considered of network 2 by using the initial conditions from combinations 3 and 4. Cheaper solutions for OPF problem can be achieved by using e.g. combinations 11 or 15. However, these combinations made the OPF algorithm expend more time in comparison to combinations 3 or 4. Since the approach used in this research aimed to obtain the fastest and feasible solution in a restricted time frame, OPF results using combinations 2 and 4 were the input to the next stages: *Contingencies selection and evaluation*.

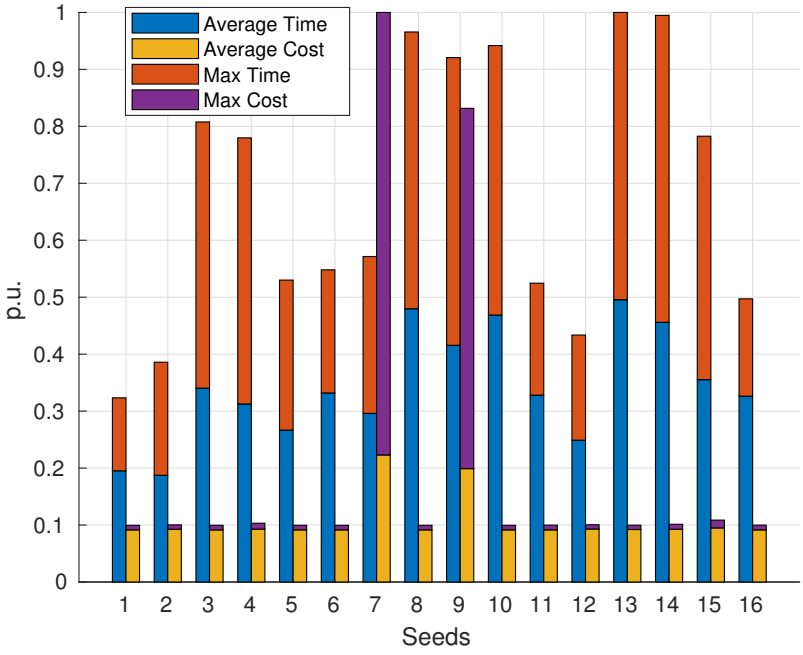

Figure 13 – Performance of Parallel OPF stage for network of 11615 buses.

Figure 13 shows the parallel OPF performance in 7 scenarios of network 3. All the combinations resulted in similar maximum and average costs, except for combinations 7 and 9. Maximum and mean time were notably different though. The fastest and feasible OPFs were obtained by mostly using combinations 1 and 2 in the scenarios of this network.

5.6. Contingencies selection and evaluation

To test the performance of contingencies selection (CS), a scenario was tested for each network of Table 2. CS was made for each iteration of the algorithm, leaving a fixed percentage of selection (25%, 50%, 75%, 100%) of the total number of sorted generation and branch contingencies.
In each iteration, an evaluation of 100% of the contingencies was carried out in order to determine the total number of violations (overvoltage, undervoltage, overloads) and non-convergences. In addition, the operating cost of the system was computed by taking into account the penalties for overload and power unbalance.
Voltage violations for the k contingency are translated into a power unbalance, according to 32 and 33 equations. The voltage magnitude v_i is the output of a forcing function $f(v)$, where for each v_i in the k contingency, the function operates as follows:

$$f(v_i) = \begin{cases} \underline{v_i} & \text{if } v_i \leq \underline{v_i} \\ v_i & \text{if } \underline{v_i} \leq v_i \leq \overline{v_i} \\ \overline{v_i} & \text{if } v_i \geq \overline{v_i} \end{cases} \tag{52}$$

To compute the penalization cost, equations 35 and 36 were used. The α function was tuned as follows:

$$\alpha(x) = \begin{cases} 1000|x| & \text{if } 0 \leq |x| \leq 2 \\ 5000|x| & \text{if } 2 < |x| < 50 \\ 1e6|x| & \text{if } |x| \geq 50 \end{cases} \tag{53}$$

where x is in p.u.
Figure 14 shows the transformed cost ($Cost^*$) and violations number in function of the algorithm iterations for the network 1. The transformed cost was computed because the unbalance penalization was higher with respect to the

minimum cost in the first three iterations, so it was computed as follows:

$$Cost^* = log_{10}(\frac{Cost}{C_{min}}) \tag{54}$$

In all cases, the algorithm ends with in 5 iterations. The minimal cost C_{min} was $2.63e5\$$. A 50% of CS is enough to achieve the minimal cost.
The computation time for 25% was 33.08 seconds, and for 100% was 42 seconds. Therefore just a 26% of computation time was increased when 75% more contingencies are evaluated.

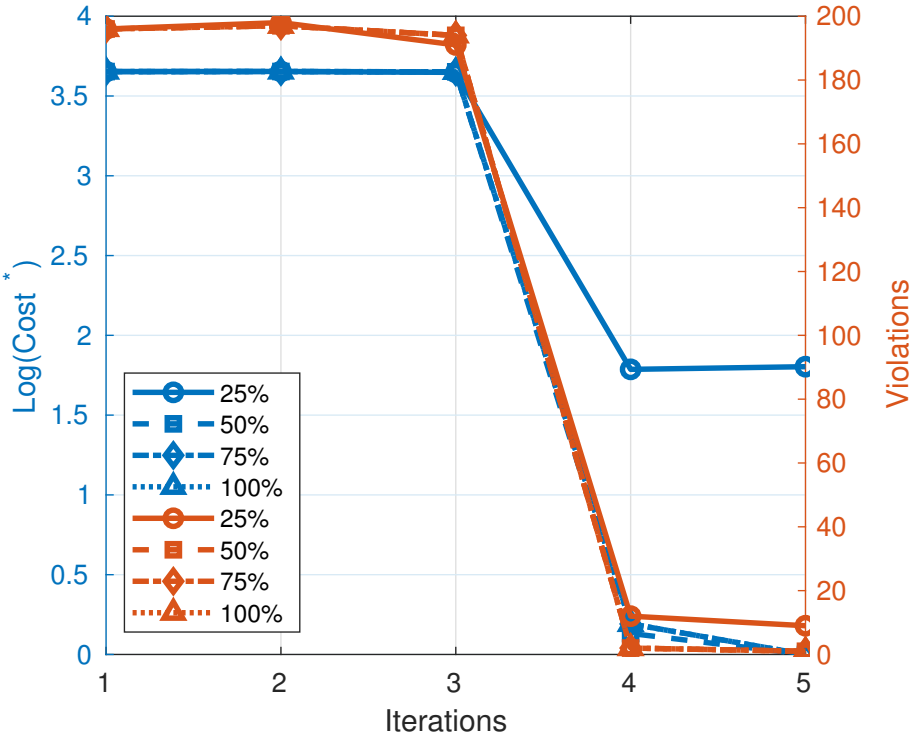

Figure 14 – Cost and violations in function of iterations for network 1.

Figure 15 depicts the costs and number of violations in function of the iteration of the algorithm for 4,918-buses network. In this case, the operation cost is minimal for 50% and highest for 75% of CS. For 100% the algorithm save the best iteration, so the minimal cost is reached in the iteration 5.
In terms of violations, with 50% of CS there were 33 violations and with 100% the algorithm ended with 23. This shows that it is not always cheaper to have fewer violations. In terms of time, for 25% of CS, the algorithm ended in the lowest time (682s) and final cost was lower than for 100%. With 75% of CS, the iterations were only 4, but the computation time was higher than with 25%. In general terms for a real time approach 25% of CS had an acceptable performance for this scenario.

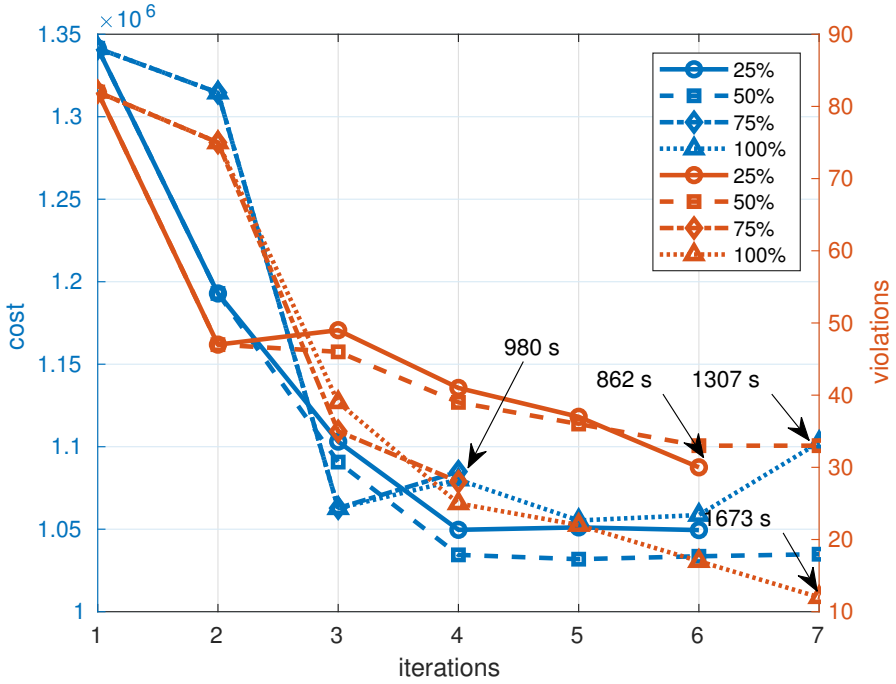

Figure 15 – Cost and violations in function of iterations for network 2.

In Figure 16, the transformed cost 54 is used again. In this case, a low cost is reached with 50%, 75%, and 100% of CS. With 25%, the penalization is approximately of 64 times higher than the minimal cost.
In Table 7 is summarized the results for the network 3. With 25% the algorithm ends with 22 violations of voltages. Although only 0.19% of the total number of the buses is violated the unbalance cost could be unacceptable. With 50%, 75%, and 100%, the penalty cost of violation could be acceptable. In terms of time, the best performance was reached with 75% of CS.

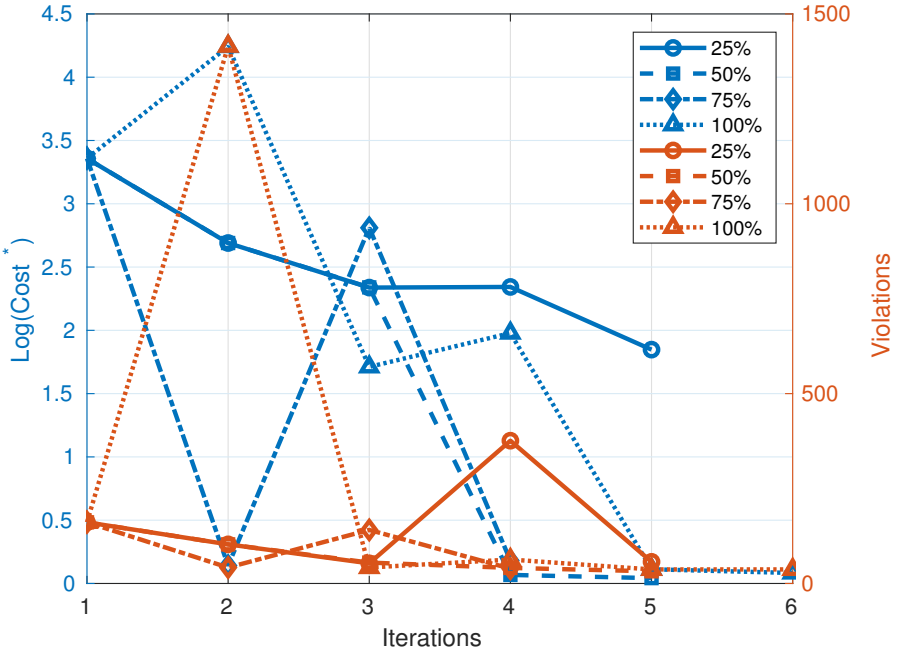

Figure 16 – Cost and violations in function of iterations for network 3.

Table 7 – Summary Results for different percentage of CS

Screening percentage	25	50	75	100
Convergence Time [s]	2200	2817	2542	5352
Cost (1e8) [$]	1.3467	0.0211	0.0264	0.0231
Violations [$]	22	2	0	2

Figure 16 shows the average time per contingency in function of the network size by using 16 and 72 workers. The time differences between them for a 11,615 network is 52 ms/cont, therefore, for 75% (6,560 contingencies) the time would decrease in around 341 s per iteration. For four iterations, the computation time would have been about 1,178 s (19.6 minutes) instead of 2,542s, reducing the calculation time by more than twice.

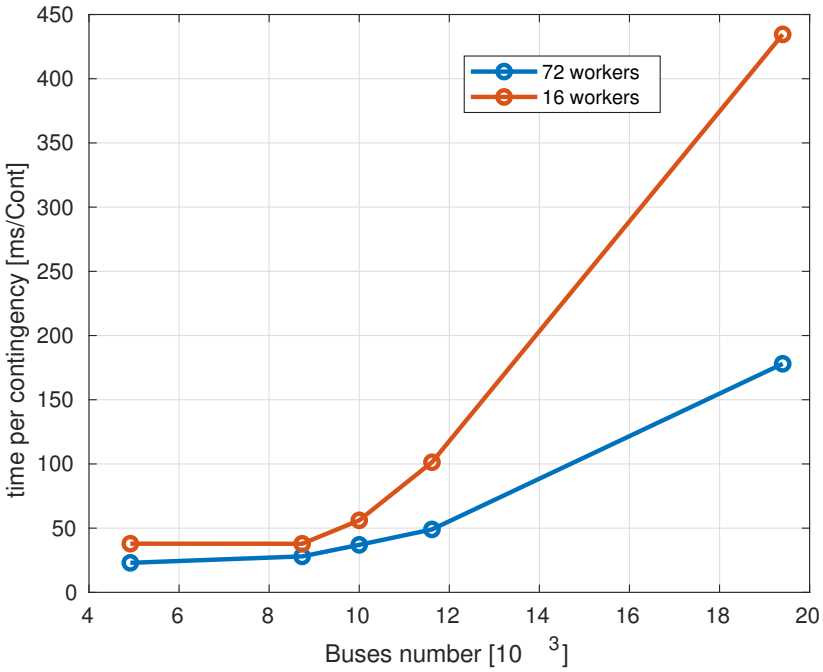

Figure 17 – Average time per contingency in function of number of buses.

6. Software deliverable Submission

To request the source code you must send an e-mail to the account **diego.rodriguez@gers.com.co**, with the subject **SCOPF-CODE REQUEST**. The applicant must have a Git-Hub account for the repository to be shared (`https://github.com/gersusa/FINAL_FINAL`).
The software developed works with minimum hardware requirements, however, the performance of the calculations improves as the PC has better resources of RAM memory, processor speeds and number of cores (workers).
Figure 17 shows the average evaluation time per contingency when these were evaluated doing use of different workers. It was observed that is more effective use 72 than 16 workers for networks with more than 10,000 buses. This observation allows to see the dependence of the software developed with the numbers of workers.
As fewer workers and RAM are available, the algorithm determines more expensive results for real-time approaches. If time is not a limitation, then the algorithm will take longer to arrive at an answer but the results may be satisfactory.
To execute the code developed, it is necessary to have installed:

- **MatLab** version (e.g 2016) with the **parallel computing Toolbox** active.

- *MatPower 7.0b*.
- *The repository files in the MatLab path*.
- *IPOPT* compilated (mexa file). The installation instructions can be found in `https://coin-or.github.io/Ipopt/` for linux or windows distributions .

The *README.md* file shows the instructions for using the tool. It also briefly describes the main functions of the code developed.

BIBLIOGRAPHY

[1] M. Al-Saffar and P. Musilek, "Distributed Optimal Power Flow for Electric Power Systems with High Penetration of Distributed Energy Resources", in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, IEEE, May 2019. DOI: 10.1109/ccece.2019.8861718. [Online]. Available: https://doi.org/10.1109/ccece.2019.8861718.

[2] V. Hinojosa and F. Gonzalez-Longatt, "Preventive Security-Constrained DCOPF Formulation Using Power Transmission Distribution Factors and Line Outage Distribution Factors", *Energies*, vol. 11, no. 6, p. 1497, Jun. 2018. DOI: 10.3390/en11061497. [Online]. Available: https://doi.org/10.3390/en11061497.

[3] M. Javadi, A. E. Nezhad, M. Gough, M. Lotfi, and J. P. Catalao, "Implementation of Consensus-ADMM Approach for Fast DC-OPF Studies", in *2019 International Conference on Smart Energy Systems and Technologies (SEST)*, IEEE, Sep. 2019. DOI: 10.1109/sest.2019.8848992. [Online]. Available: https://doi.org/10.1109/sest.2019.8848992.

[4] S. Lee, W. Kim, and B. H. Kim, "Performance Comparison of Optimal Power Flow Algorithms for LMP Calculations of the Full Scale Korean Power System", *Journal of Electrical Engineering and Technology*, vol. 10, no. 1, pp. 109–117, Jan. 2015. DOI: 10.5370/jeet.2015.10.1.109. [Online]. Available: http://dx.doi.org/10.5370/JEET.2015.10.1.109.

[5] F. Capitanescu, J. M. Ramos, P. Panciatici, D. Kirschen, A. M. Marcolini, L. Platbrood, and L. Wehenkel, "State-of-the-art, challenges, and future trends in security constrained optimal power flow", *Electric Power Systems Research*, vol. 81, no. 8, pp. 1731–1741, Aug. 2011. DOI: 10.1016/j.epsr.2011.04.003. [Online]. Available: https://doi.org/10.1016/j.epsr.2011.04.003.

[6] S. A. Sadat, D. Haralson, and M. Sahraei-Ardakani, "Security versus Computation Time in IV-ACOPF with SOCP Initialization", in *2018 IEEE International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, IEEE, Jun. 2018. DOI: 10.1109/pmaps.2018.8440287. [Online]. Available: https://doi.org/10.1109/pmaps.2018.8440287.

[7] S. Stankovic and L. Soder, "Optimal Power Flow Based on Genetic Algorithms and Clustering Techniques", in *2018 Power Systems Computation Conference (PSCC)*, IEEE, Jun. 2018. DOI: 10.23919/pscc.2018.8442583. [Online]. Available: http://dx.doi.org/10.23919/PSCC.2018.8442583.

[8] A. Zamzam and K. Baker, "Learning optimal solutions for extremely fast ac optimal power flow", *arXiv preprint arXiv:1910.01213*, 2019.

[9] I. Ghosh and P. K. Roy, "Application of Earthworm Optimization Algorithm for solution of Optimal Power Flow", in *2019 International Conference on Opto-Electronics and Applied Optics (Optronix)*, IEEE, Mar. 2019. DOI: 10.1109/optronix.2019.8862335. [Online]. Available: http://dx.doi.org/10.1109/OPTRONIX.2019.8862335.

[10] D. Lastomo, Widodo, and H. Setiadi, "Optimal Power Flow using Fuzzy-Firefly Algorithm", in *2018 5th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*, IEEE, Oct. 2018. DOI: 10.1109/eecsi.2018.8752903. [Online]. Available: http://dx.doi.org/10.1109/EECSI.2018.8752903.

[11] J. Guo, G. Hug, and O. K. Tonguz, "A Case for Nonconvex Distributed Optimization in Large-Scale Power Systems", *IEEE Transactions on Power Systems*, vol. 32, no. 5, pp. 3842–3851, Sep. 2017. DOI: 10.1109/tpwrs.2016.2636811. [Online]. Available: https://doi.org/10.1109/tpwrs.2016.2636811.

[12] M. Grabada Echeverri, M. J. Rider Flores, and J. R. S. Mantovani, "Dos técnicas de descomposición aplicadas al problema de flujo de potencia óptimo multi-areas", es, *DYNA*, vol. 77, pp. 303–312, Jun. 2010, ISSN: 0012-7353. [Online]. Available: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0012-73532010000200031&nrm=iso.

[13] J. Guo, G. Hug, and O. Tonguz, "Asynchronous ADMM for Distributed Non-Convex Optimization in Power Systems", *arXiv preprint arXiv:1710.08938*, 2017.

[14] D. Phan and J. Kalagnanam, "Some Efficient Optimization Methods for Solving the Security-Constrained Optimal Power Flow Problem", *IEEE Transactions on Power Systems*, vol. 29, no. 2, pp. 863–872, Mar. 2014. DOI: 10.1109/TPWRS.2013.2283175.

[15] Y. Li and J. McCalley, "Decomposed SCOPF for Improving Efficiency", *IEEE Transactions on Power Systems*, vol. 24, no. 1, pp. 494–495, Feb. 2009. DOI: 10.1109/tpwrs.2008.2002166. [Online]. Available: https://doi.org/10.1109/tpwrs.2008.2002166.

[16] J. Mohammadi, G. Hug, and S. Kar, "A benders decomposition approach to corrective security constrained OPF with power flow control devices", in *2013 IEEE Power & Energy Society General Meeting*, IEEE, 2013. DOI: 10.1109/pesmg.2013.6672684. [Online]. Available: https://doi.org/10.1109/pesmg.2013.6672684.

[17] Q. Wang, J. D. McCalley, T. Zheng, and E. Litvinov, "A Computational Strategy to Solve Preventive Risk-Based Security-Constrained OPF", *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 1666–1675, May 2013. DOI: 10.1109/tpwrs.2012.2219080. [Online]. Available: https://doi.org/10.1109/tpwrs.2012.2219080.

[18] D. T. Phan and X. A. Sun, "Minimal Impact Corrective Actions in Security-Constrained Optimal Power Flow Via Sparsity Regularization", *IEEE Transactions on Power Systems*, vol. 30, no. 4, pp. 1947–1956, Jul. 2015. DOI: 10.1109/tpwrs.2014.2357713. [Online]. Available: https://doi.org/10.1109/tpwrs.2014.2357713.

[19] Y. Xu, H. Yang, R. Zhang, Z. Dong, M. Lai, and K. Wong, "A contingency partitioning approach for preventive-corrective security-constrained optimal power flow computation", English, *Electric Power Systems Research*, vol. 132, pp. 132–140, 2016, ISSN: 0378-7796. DOI: 10.1016/j.epsr.2015.11.012.

[20] F. Capitanescu, "Critical review of recent advances and further developments needed in AC optimal power flow", *Electric Power Systems Research*, vol. 136, pp. 57–68, Jul. 2016. DOI: 10.1016/j.epsr.2016.02.008. [Online]. Available: https://doi.org/10.1016/j.epsr.2016.02.008.

[21] L. Yang, C. Zhang, and J. Jian, "A parallel method for solving the DC security constrained optimal power flow with demand uncertainties", *International Journal of Electrical Power & Energy Systems*, vol. 102, pp. 171–178, 2018, ISSN: 0142-0615. DOI: https://doi.org/10.1016/j.ijepes.2018.04.028. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0142061517333367.

[22] S. Eftekharnejad, "Selection of multiple credible contingencies for real time contingency analysis", in *2015 IEEE Power Energy Society General Meeting*, Jul. 2015, pp. 1–5. DOI: 10.1109/PESGM.2015.7286593.

[23] R. D. Zimmerman and C. E. Murillo-Sánchez, *MATPOWER*, en, 2019. DOI: 10.5281/ZENODO.3251119. [Online]. Available: https://zenodo.org/record/3251119.

[24] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming", *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, Apr. 2005. DOI: 10.1007/s10107-004-0559-y. [Online]. Available: https://doi.org/10.1007/s10107-004-0559-y.

[25] HSL, *A collection of Fortran codes for large scale scientific computation*, Availabel in: http://www.hsl.rl.ac.uk/, 2019 (accessed December 5th, 2019).

[26] ARPA-E, "SCOPF Problem Formulation: Challenge 1", Advanced Research Projects Agency–Energy), Tech. Rep., Apr. 2018.

[27] ARPA"-E, *Grid Optimization (GO) Competition*, Available in: https://gocompetition.energy.gov, 2019 (accessed December 5th, 2019).

[28] H. Bevrani, *Robust Power System Frequency Control*, ser. Power Electronics and Power Systems. Springer US, 2008, ISBN: 9780387848785. [Online]. Available: https://books.google.com.co/books?id=wVtGuzkrwiAC.

[29] J. Zhao, H.-D. Chiang, H. Li, and P. Ju, "On PV-PQ bus type switching logic in power flow computation", vol. 25, Jan. 2008.

[30] F. Capitanescu, M. Glavic, D. Ernst, and L. Wehenkel, "Applications of security-constrained optimal power flows", in *In Proceedings of Modern Electric Power Systems Symposium, MEPS06*, 2006.