Tactics for improving computational performance of criticality analysis in state estimation

Vinicius B. Braga Flôr | Milton B. Do Coutto Filho | Julio C. Stacchini de Souza | Andre A. Augusto

1 Institute of Computing, Fluminense Federal University, R. Passo da Pátria, Niterói, Brazil
2 Dept. of Electrical Engineering, Fluminense Federal University, R. Passo da Pátria, Niterói, Brazil

Correspondence
Milton B. Do Coutto Filho, Institute of Computing, Fluminense Federal University, R. Passo da Pátria, 156, Niterói, Brazil.

Abstract
Power system state estimation is a needful tool for providing reliable data on the system operating conditions. As such, state estimation requires adequate data redundancy levels, which can be expressed in terms of the degrees of criticalities associated with different groups of measurements. Criticality analysis allows the assessment of the potential risks that could impact state estimation results. The identification of criticalities is a computationally-intensive process owing to the combinatorial nature of the problem. This paper proposes some Branch-and-Bound-based tactics geared up for improving the computational efficiency of criticality analysis in state estimation. Throughout the paper, the authors sum up significant advances concerning their previous research, in terms of branching and bounding operations, choice of search strategies, data structures adopted, the proposition of a coefficient of criticality, uninformed and informed search schemes, different ways of objective function evaluation, and exploration of different objectives to be accomplished. Simulation results obtained on the IEEE 118-bus test system evince the performance of the proposed algorithms.

1 | INTRODUCTION

Since its commencement about 50 years ago, the power system state estimation (SE) has made great strides on the road towards acceptability [1]. Nowadays, SE is a computer-aided tool of energy management systems considered crucial for obtaining reliable data (not contaminated with gross errors) on the most likely system operating point—usually to assess adequacy. Within the ambit of SE, the operating condition is referred to as the static state, well-characterized by the complex bus voltages (phase angles and magnitudes) on the current grid [2].

For monitoring power grids, measurements (observations) should be taken and conveniently processed by SE. The ability of SE to observe the system state (to gain information) is translated into the observation capability of measuring systems. This aptitude is impacted by the existence of essential elements for SE, known as critical data. Therefore, the allocation of measuring units (MUs) in the network—devices used to measure a group of electrical quantities remotely—should be strategically planned so that SE is not unduly exposed to data unavailability, which would lead it to produce unreliable results. The term MU refers to the following measurement devices indistinctly: remote terminal unit (RTU), intelligent electronic device (IED), and phasor measurement unit (PMU). Usually, RTUs/IEDs obtain power flow/injection and voltage magnitude measurements, whereas PMUs gather voltage/current synchrophasors. Substations can be equipped with a combination of these devices, collectively named MU.

It is increasingly important to understand and manage the risks that complex modern systems are exposed to while performing the tasks for which they were conceived. In the realm of SE, one should recognize that the lack of data redundancy can pose the estimation process under threat of being unable to cover the network as a whole (risk of impending unobservability), as well as to deal with measurements containing gross errors (risk of providing unreliable state/measurement estimates). The term bad data (BD) usually refers to spurious measurements.
Identifying adverse conditions for both observability and BD debugging is performed by the criticality analysis (CA) of SE available measurements. In other words, CA deals with the identification of measurements, taken individually or arranged in groups, based on their essentiality to system observability, i.e. considering the risk that their unavailability may present to the SE process. CA is also valuable in optimizing measuring systems (planning or reinforcing them), taking into account alternatives of different reliability levels for the placement of MUs, established by the possible loss of measurements or outages of network branches. Preventive maintenance scheduling tasks performed on measuring systems devoted to SE can be based on CA to organize and prioritize the necessary activities, according to the risks involved.

One can define a critical tuple of order k or cardinality k (hereafter denoted by Ck) as a group of k elements such that the unavailability of them all together renders the system unobservable, and the unavailability of any combination formed with part of them does not result in unobservability [3]. Thus, a single critical element is designated as C1, critical pair as C2, critical triplet as C3, and so forth. One or more C2s, those of having a measurement in common, form BD groups or minimally dependent sets [4]. The general concept of criticality can be applied to both elements of a measuring system, i.e. individual measurements or their groups that integrate the MUs placed at each bus of the power grid.

After the Introduction, the manuscript is organized as follows. Section 2 surveys the existing works on CA, and Section 3 presents the paper’s motivation and contributions. Section 4 briefly reviews the WLS estimation process regarding the participation of matrices Gain and residual covariance in CA. Section 5 is devoted to the formulation of branch-and-bound (B&B) algorithms, computationally attractive, to deal with the identification of SE criticalities. Numerical results, carried out on the IEEE 118-bus test system to illustrate the performance of the proposed B&B algorithms, are included in Section 6. Finally, Section 7 presents the conclusions reached in the paper, and Section 8 lists the references.

2 | LITERATURE REVIEW

Despite SE be an active research field, reviewing the literature on the subject, one ascertains that the studies on CA still are scarce. Most of the studies have been dedicated to finding Cks of individual measurements with cardinality up to three [5–8]. In these studies, CA is not viewed as a combinatorial optimization problem. Reference [9] is of central importance to identifying Cks in a set of measurements and analysis of their influence on detecting and identifying multiple BD. In [10, 11], the aim is to identify a plan with w measurements required to guarantee the power grid observability if any k measurements become unavailable. More recently, the search for minimum cardinality Cks containing at least one arbitrarily chosen measurement is presented in [12]. Also, a measuring system’s state observation capability considering different criticalities levels is probabilistically evaluated in [13]. The first results towards the general solution of the hard-combinatorial optimization problem of identifying critical elements in SE are reported in [14], being considered so far a benchmark for CA in SE.

One should have in mind that the solution of an optimization problem, primarily those computationally-intensive, such as the CA, is influenced by many factors, namely: choice of the optimization method; formulation of the objective function and its evaluation; and crucially, the strategic way the optimization algorithm is implemented. The present research is concerned with all these factors, and the numerical results achieved here evince the performance of the proposed approach.

For most discrete optimization problems, those related to CA included, complete (or explicit) enumeration of candidate solutions exceeds reasonable limits; even for moderate dimension instances, the number of solution points computed by the Brute Force (BF) enumeration method is immense. For instance, considering the IEEE 118-bus system, supervised by 99 MUs, the number of combinations of these MUs in tuples of cardinality up to four approaches the four million mark. These 99 MUs gather up 176 pairs of measurements, making the system observed in full. Thus, such a naive approach is inappropriate to solve real-world problems since the number of feasible solutions usually grows exponentially with the instance’s size to be solved.

Among the exact classical methods for discrete optimization—since it was advocated by Land and Doig [15] in 1960—B&B has become the first choice for many problems, being implemented in different ways [16, 17] and considered the heart of numerous state-of-the-art applications for integer programming [18]. B&B is a general-purpose approach that uses simple principles to enumerate every solution point implicitly. It adopts strategies in the rooted search tree to establish when to grow (branching) or to stop growing (bounding) the tree [19, 20]. The first step towards applying the B&B paradigm to perform CA in SE has been given in [14]. Thus, it becomes natural to recognize that there is ground for further research efforts on the subject, and this paper aims at contributing to the development of novel computationally efficient B&B algorithms.

3 | MOTIVATION AND CONTRIBUTIONS

Although SE can be considered a mature function, one should permanently be searching for improving it. In this sense, the need to direct research efforts towards aspects that involve the SE process’s reliability has become increasingly evident. CA is one of these aspects, considered fundamental here. The primary motivation for presenting this paper is that, to date, no work on applying a combinatorial optimization method to carry out CA—in a reasonable computing time—not restricted to low-cardinality criticalities is found.

CA establishes different observability degrees, i.e. it quantifies the SE vulnerability to events involving the unavailability of measurements. Concerning these events, they may occur due to temporary malfunction of the communication system entailing
the loss of measurements, BD elimination, MUs out-of-service for maintenance (block unavailability of measurements), and unexpected network branch outages with loss of measurements. The cyber-security analysis of intelligent grids has recently raised the problem of SE vulnerability to an assemblage of malicious attacks, such as those that inflict BD intentionally—not by chance, as they usually occur. It is reasonable to believe that identifying how exposed the SE is to the measurement unavailability overrides the determination of how it may occur, by chance, or deterministically. An extensive CA provides answers to the problem posed, being the central aspect of the approach conveyed in this paper.

Concisely, CA can be enunciated as the following problem: given a power grid observed through a measuring system, determine the elements (considered individually or forming groups) critical/essential to the SE process. The problem is computationally tricky since many element combinations are checked concerning the state observability. Criticalities do not indicate only imminent unobservability but also SE weaknesses in processing BD effectively by the residual statistical analysis. Illustrating, gross errors in C1 measurements are undetectable, in C2-tuples are detectable but unidentifiable, and generalizing, (k−1) gross errors in Ck-tuples of measurements are not identified. CA gives answers in the interest of SE reliability results, identifying low metering coverage zones, i.e. weak spots (critical MUs-centred) in which metering reinforcements are suggested.

The paper addresses the challenging problem of developing an efficient computational way of finding critical MUs (in their various degrees) installed in each measuring system designed for SE. This work continues the research effort carried out in [14], in which a naïve B&B (also referred to as reference implementation) was used. Now, novel B&B algorithms, more intelligent and efficient computationally, are formulated in the paper. They combine tactics conceived to incorporate relevant aspects regarding:

1. Search strategies: They drive the algorithms towards optimal solutions by determining the strategic order in which unexplored subproblems in the B&B search tree (used to model the sequence of actions) are selected for exploration. The way these subproblems are arranged has the potential to affect the amount of computing time and memory required by the B&B algorithm. The performance of two usual search strategies is assessed in this study, namely: the depth-first search (DFS), in which the data structure is a stack, and the one with the opposite effect, the breadth-first search (BFS), with a queue data structure. Reference [14] has addressed only the BFS strategy.

2. Data structures: The data organization, management, and storage formatting, enabling efficient access and modification, also plays a significant role in the performance of searching algorithms. In DFS, the data is stored as a stack, whereas BFS uses a queue data structure. There is a repertoire of current search engines and to choose among them represents extra complexity to the problem at hand. The proposed strategies are assessed, taking into account the use of data structures such as the priority queue and hash table. In the paper, priorities are assigned according to a new metric defining the proximity to unobservability.

3. Uninformed and informed search algorithms: There are two fundamental types of search, considering whether they utilize further problem-specific information on the goal pursued. Only the uninformed search has been adopted in [14]. Instead, informed search algorithms, handy for ample search space (S), use the knowledge on how far from goal nodes the candidate solution is. Both uninformed (blind) and informed search strategies are evaluated in this paper.

4. Specific objectives: The complete problem of searching for critical MUs can be considered intractable computationally for large-scale systems, owing to the exponentially increasing number of potential solutions. However, it may be of practical interest to solve the problem in a reduced S. This paper explores the flexibility of the B&B algorithmic framework, proposing, for instance, the task of searching for critical data within a radial area centred in a specified MU or else finding all the CkS containing a given MU with a high failure rate (unavailability).

5. Criticality evaluation: The classical observability analysis is performed to determine whether a measuring system provides enough measurements (varied and strategically dispersed in the grid) to make the SE process successful. The observability/criticality condition is confirmed when the weighted least squares (WLS) process gain matrix is invertible. Especially in CA, the residual covariance matrix [3] can be utilized since its elements represent the degree of interaction between measurements. In contrast to the study reported in [14], in which only the Gain matrix has been used, the performance of the B&B algorithms proposed here is evaluated, adopting both the Gain and Covariance matrices to evaluate criticalities.

6. Utilization: In favour of the proposed algorithms, one can remark that they do not require an already installed SE application, thus enabling simple testing of a measuring system, for instance, at the planning/commissioning stage. They also help in making maintenance schedules of MUs.
4 | WLS STATE ESTIMATION

Properly provided with varied, strategically allocated, and abundant measurements, SE becomes capable of facing the challenge of producing reliable results. This section presents a quick reference to the WLS state estimation process. The numerical approach adopted to address the network observability. The residual analysis and its connection with data criticalities are described. For a detailed description of each SE step, the readers can refer to [2, 3]. The notation used throughout the paper is the following: uppercase and lower-case boldface italic type letters indicate matrices and column vectors, respectively; (.)' denotes vector/matrix transpose.

CA in SE concerns the assessment of the reciprocal relation between interdependent state variables and their observations (measurements). This interrelation—created by the power grid configuration regardless of the measurement values—is quantified through distinct observability degrees. Given the CA structural nature, the methods adopted typically assume paired (P, Q) measurements and make use of the active power-angle linear model:

\[ \mathbf{r}_a = \mathbf{H}_a \hat{\theta} + \mathbf{v}_a \]  

where \( \hat{\theta} \) – bus voltage phase angle vector \((n \times 1)\); \( n \) – no. of phase angles (active state variables); \( \mathbf{r}_a \) – active measurement vector \((m \times 1)\); \( m \) – no. of measurements; \( \mathbf{v}_a \) – active measurement error vector \((m \times 1)\), supposed to have zero mean and covariance matrix identical to \( \mathbf{I} \) (identity); \( \mathbf{H}_a \) – active measurement Jacobian matrix \((m \times n)\). For the sake of simplicity, henceforth, subscript “a” (indicates \( P-\theta \) model) is omitted.

According to Equation (1) and with the WLS state estimation process, the estimated \( \hat{\theta} \) (denoted \( \hat{\theta} \)), is given by:

\[ \mathbf{G} \hat{\theta} = \mathbf{H}' \zeta \]  

where \( \mathbf{G} = \mathbf{H}' \mathbf{H} \) is the active Gain matrix.

The classical observability analysis occupies itself by checking whether there are enough measurements available to make SE possible. In mathematical terms, it means that the unique solution of Equation (2) can be achieved for the entire power grid. Matrix \( \mathbf{G} \) should be non-singular in observable networks, i.e. its rows/columns are linearly independent. One way to verify this condition is to apply the Cholesky factorization to the symmetric matrix \( \mathbf{G} \), and at the end of the process, no zero pivots should occur (i.e. \( \text{det} \mathbf{G} \neq 0 \)).

In the residual analysis, the residual vector \( \mathbf{r} \)–whose elements are the differences between \( \zeta \) and corresponding estimated values \( \hat{\zeta} \)–is normalized and submitted to the following statistical validation (\( r_N \)-test):

\[ \hat{\zeta} = \mathbf{H} \hat{\theta} \]  

\[ r_N (i) = \frac{|r (i)|}{\sigma_{\Omega} (i)} \]  

where \( \hat{\theta} \) is obtained from Equation (2); \( \Omega \) is the residual covariance matrix and \( \Omega(i, i) \) its element in the \( i \)th row and \( i \)th column; \( \sigma_{\Omega} (i) = \sqrt{\Omega(i, i)} \) is the standard deviation of the \( i \)th residual vector component. Threshold violations arouse suspicion that corrupted data are present.

The covariance matrix \( \Omega \) contains elements representing the degree of interaction between measurements, which can be explored in CA. A \( \mathcal{C}_k \) is formed by a group of \( k \) measurements whose corresponding rows/columns of the matrix \( \Omega \) are linearly dependent.

Illustrating with the simplest case, if the matrix \( \Omega \) has a row/column of zeros, then the corresponding measurement is a C1 (not correlated with any other in the set). Now, consider the situation in which no C1 is present and admit \( \Omega^{(k)} \) as a sub-matrix of \( \Omega \), for instance, composed of the rows/columns associated with a group of \( t \) measurements served by \( k \) MUs. If the unavailability of this \( k \)-tuple of MUs makes the grid unobservable (i.e. \( (t \times k) \) matrix \( \Omega^{(k)} \) has columns/rows linearly dependent, i.e.\( \text{det} \mathbf{\Omega}^{(k)} = 0 \)), then this is a candidate CA. The identification of \( \mathcal{C}_k \)s can be verified by the Cholesky decomposition process [3], applied for the triangulation of \( \Omega^{(k)} \).

5 | BRANCH-AND-BOUND TACTICS

This section starts describing the combinatorial optimization problem to be solved, known as CA. The problem is then formulated using the B&B algorithmic framework, and the aspects mentioned in Section 3 concerning its efficient computational implementation are applied here to solve the CA problem, namely: branching and bounding operations, search strategies, data structures, uninform and informed search schemes, objective function evaluation, and tactics for reducing the search space.

5.1 | Combinatorial optimization

Power system researchers face more and more situations of increasing complexity. CA is one of the challenging items on the SE agenda involving the field of discrete combinatorial optimization.

In the SE context, criticality is the level of contribution of an element (e.g. measurements, MUs, network branches) to the estimation process in maintaining observability. CA is the process of assessing the different criticality levels (expressed as the cardinality \( k \) of a tuple of elements) of a given measuring system. It has the purpose of protecting the SE residual analysis, which is vital to assure the reliability of the obtained estimated quantities. The scope of the analysis can be extended to cover cyber threats and vulnerabilities.
Consider the unavailability of MUs as the dominant event (most likely to occur) in CA [10]. The problem to be solved consists of listing \(k\)-tuples of MUs, a combinatorial process with an explosion of possibilities, followed by a decision problem that asks whether the system becomes unobservable when each \(k\)-tuple is unavailable. In classical observability analysis, observable networks are those for which the Gain matrix \(G\) of the estimation process is non-singular. The formation of \(k\)-tuples and how they are checked regarding observability define the objective to be pursued. This objective is characterized by a function whose values are evaluated in the search for optimal solutions to the combinatorial problem.

The B&B paradigm is customarily adopted in combinatorial optimization problems. It constitutes a way of enumerating (intelligently and implicitly) all candidate solutions, with a branching scheme that generates different states or nodes, giving rise to a search tree. Once this tree has been thoroughly explored, an optimal solution achieved by the B&B guided search is informed. The term branch refers to the fact that the method recursively divides \(\cal S\), thus determining when to stop the growth of the branching tree.

To determine how efficiently an algorithm solves a problem, usually in terms of computing time and memory, is a task concerned with the complexity analysis. B&B algorithms have a worst-case running time of \(O(M^b)\), where \(M\) is a bound on the length of time to explore a subproblem, \(b\) is the branching factor, and \(d\) is the search depth of the search tree [17]. Considering CA in SE, characterized in the next section by Equation (7), the time complexity to evaluate \(f(T^{(k)})\) is \(O(M^2k)\), where \(M\) is the time upper bound to verify network observability and \(k\) is the cardinality of the tuple \(T^{(k)}\) under analysis. Careful computation of the objective function \(f(\cdot)\) can reduce this time up to \(O(Mk)\). Since \(T^{(k)}\) influences \(M\)—note that the evaluation of the function \(f(\cdot)\) requires to check \(T^{(k)}\) for observability, as well as all its subtuples—the combination of different tactics in B&B implementation, intending to generate subproblems/solutions intelligently, can result in different worst-case running times. As commonly used in practice, in this paper, the efficiency of B&B algorithms is numerically demonstrated by the computational running time in conjunction with the number of nodes visited.

### 5.2 Problem formulation

Let the search space \(\cal S\) be the set of MUs associated with a given observable network and \(T^{(k)}\) a subset (\(k\)-tuple) of \(\cal S\)—associated with the index \(k\) that denotes an action of grouping elements from \(\cal S\), for \(k = 1, \ldots, n_j\) (no. of clusters)—to be tested in terms of its impact on the grid observability. Thus, consider the function \(g: \cal P(\cal S) \rightarrow \mathbb{N}\) given by:

\[
g(T^{(k)}) = \begin{cases} 0, & \text{if the grid is observable } \big(T^{(k)}\big) \text{ unavailable} \\ 1, & \text{if the grid is unobservable } \big(T^{(k)}\big) \text{ unavailable} \end{cases}
\]

where \(P\) is the power set of \(\cal S\)—i.e. the set whose members are all possible subsets (tuples) of \(\cal S\), including the empty set \(\emptyset\) and \(\cal S\) itself. Also, taking a subset \(T^{(k)}\) (i.e. a subtuple of the \(k\)-tuple \(T^{(k)}\)), one can state that:

a. if \(g(T^{(k)}) = 0\), then \(g(T^{(k)}) = 0\) for every \(T^{(k)} \subseteq T^{(k)}\);  
   b. if \(g(T^{(k)}) = 1\), then there is at least one \(T^{(k)} \subseteq T^{(k)}\) such that \(g(T^{(k)}) = 0\).

Condition (a) establishes that a \(T^{(k)}\) when unavailable does not make the grid unobservable. Neither does any of its subtuples. On the other hand, according to condition (b), the network is unobservable with an unavailable \(T^{(k)}\) because there is at least one subtuple \(T^{(k)}\) whose unavailability does not cause unobservability. Note that \(\emptyset\) (no action of grouping measurements from \(\emptyset\)) is a subset of any \(T^{(k)} \subseteq \emptyset\).

If a group of \(k\) MUs (\(T^{(k)}\)) when unavailable lead matrix \(G\) to become singular, it means that this group forms a \(C_k\) (see Section 4). Let \(G^{(k)}\) be the Gain matrix in this case. Thus, \(\det G^{(k)} = 0\), and for any \(T^{(k)} \subseteq T^{(k)}\), \(\det G^{(k)} \neq 0\); \(G^{(k)}\) is associated with a subtuple of \(T^{(k)}\); using Equation (6), \(g(T^{(k)}) = 0\), when \(\det G^{(k)} \neq 0\), and \(g(T^{(k)}) = 1\) when \(\det G^{(k)} = 0\). Instead of adopting \(G^{(k)}\), this paper will test the use of the matrix \(\Omega^{(k)}\) to check observability.

Therefore, the following combinatorial optimization problem can be formulated:

\[
\min f(T^{(k)}) = g(T^{(k)}) + \sum g(T^{(k)}), \quad \forall T^{(k)} \subseteq T^{(k)} \\
\text{subject to } g(T^{(k)}) = 1
\]

For a given network, all the \(C_k\)s of MUs considered in CA represents the solution of Equation (7). Note that the constraint \(g(T^{(k)}) = 1\) imposes the unobservability condition when \(T^{(k)}\) is unavailable; consequently, the lowest value for a feasible solution of Equation (7) corresponds to \(f(T^{(k)}) = 1\) indicating that there are no critical subtuples \(T^{(k)} \in T^{(k)}\). According to conditions (a) and (b), the optimal solution of (7) is attained only when \(g(T^{(k)}) = 0\), for any \(T^{(k)} \subseteq T^{(k)}\). It means the unavailability of any subset of \(T^{(k)}\) (except \(T^{(k)}\) itself) does not cause unobservability. Thus, the feasible solutions of Equation (7) represent the searched \(C_k\)s.

### 5.3 Branching operations

At the root node in the B&B search tree, the original problem \(\cal S\) is sequentially divided into subproblems, each of them
associated with a node. The branching operation is used to select which generated branches/nodes should be explored first, so it improves the efficiency of B&B indirectly by pruning nodes promptly. Each node represents a given MU.

According to the candidate solution quality, the branching operation subdivides \( P(S) \) recursively. Thus, consider \( T^{(k)} \) as one of the \( ns \) possible solutions of (7) to be submitted to the branching operation. Let \( S(T^{(k)}) = \{ T^{(\ell)} \in P(S) \mid T^{(\ell)} \subseteq T^{(k)} \} \) be a superset of \( T^{(k)} \) (that is, \( S(T^{(k)}) \supseteq T^{(k)} \)). The effect of using \( P(S) \) instead of \( S(T^{(k)}) \) on (7) is a reduction of \( S \) to the solutions that contain the elements of \( T^{(k)} \). Thus, the optimal solutions are the \( C_k \)s composed of all the elements present in \( T^{(k)} \). Let \( T^{(k)} \) be the current solution in the B&B process and \( T^{(k)} \) its complement set concerning \( S \). The branching operations consist of the following steps [14]:

1. If \( T^{(k)} = \emptyset \), then stop. Otherwise, set \( ns = 1 \);
2. Select the next unvisited element \( i \in T^{(k)} \);
3. Create the augmented set \( T_{ns}^{(k)} = T^{(k)} \cup \{ i \} \);
4. Mark element \( i \) as visited;
5. If there are unvisited elements in \( T^{(k)} \), set \( ns \leftarrow ns + 1 \), and go to step 2. Otherwise, go to the next step;
6. Return the new search subspaces \( S(T_1^{(k)}) \), \ldots, \( S(T_{ns}^{(k)}) \).

5.4 | Bounding operations

Based on Equation (7), the bounding process establishes the condition in which a candidate tuple is accepted as critical. The bounding operation prunes unpromising paths, i.e. those that do not lead to optimal solutions. Given a generic \( k \)-tuple \( T^{(k)} \), first is necessary to evaluate \( g(T^{(k)}) \). If \( g(T^{(k)}) = 1 \), this \( k \)-tuple is a candidate for being an optimal solution, confirmed only if \( f(T^{(k)}) = 1 \). Otherwise, if \( g(T^{(k)}) = 0 \) it indicates non-criticality, and this \( k \)-tuple is stored for posterior exploration (generation of children). Active tuples can be stored in a queue, stack, or priority queue. The data structure selected defines the way the solutions are searched. For a candidate \( C_k \) (i.e. when \( g(T^{(k)}) = 1 \)), one can define a lower (\( lb \)) and an upper bound (\( ub \)) to evaluate the quality of the solution. Based on Equation (7), the minimum value assumed by \( f(T^{(k)}) \) is 1, then \( lb = 1 \). To assure that a \( k \)-tuple is critical, it is necessary that \( \sum g(T^{(k)}_j) = 0 \), for all \( T^{(k)}_j \subseteq T^{(k)} \). Defining \( ub = f(T^{(k)}) \), then \( T^{(k)} \) can be checked. If \( lb = ub \) the candidate tuple is critical, i.e. it is a solution of Equation (7), and should be stored in a list of optimal solutions. Otherwise, if \( lb \leq ub \) the candidate is not a \( C_k \), there is a critical \( T^{(k)}_j \) in \( T^{(k)} \), and this node can be safely discarded. In both situations, the search tree should be pruned because no other optimal solution can be found in this branch.

5.5 | DFS and BFS strategies

One can find information more effectively/efficiently via adequate strategies (considering the requisites of consumption of time/memory), which suggests the question: in which order should the unexplored parts of \( S \) be searched?

DFS and BFS are regarded as classical key strategies when solving combinatorial optimization problems by B&B. Ref. [14] has addressed only the BFS strategy. Both strategies are evaluated in Section 6.1.

DFS always expands the deepest node at the edge of the search tree. The search continues to the deepest level of the search tree, which does not have any children yet (otherwise, it is not the deepest node). The node is expanded, and then the successor (or one of the successors) is expanded, and this process is continued until the goal node is reached or the node has no more successors. If the latter has occurred, the search back-ups to the previous node and explores its other successor if any of them is still unexplored. DFS is strongly influenced by the sequence in which the MUs (nodes) are branched, and the identification of \( C_k \)s does not occur following ascending degrees of cardinalities.

On the other hand, BFS is a search strategy in which the root node is visited first, followed by the expansion of all the successors of the root node (level 1), then their successors (level 2), and so on, until that the goal node is reached. The nodes at a given depth are expanded before any other node at the next depth level. Thus, the identification of \( C_k \)s will obey ascending levels of cardinalities. For this type of search, the sequence in which the MUs are explored is unimportant. BFS requires substantial memory to store the search tree nodes.

5.6 | Data structures

In the paper, stack, queue, hash table, and priority queue are considered [19]. Search-based algorithms require data structures to store and control the search tree, i.e. they establish the sequence in which the proposed solutions are visited. DFS handles data stored as a stack, in which its last element (at the top) is processed first and the first one last. This process is known as LIFO (last in, first out). Instead, in BFS, the opposite happens; the first element is processed first and the newest last (an approach known as FIFO, stands for first in, first out). The data structure that implements FIFO is the queue. Examples of data structures more specialized are the priority queue and hash table, both also adopted in this paper (see Section 6.2).

The priority queue elements are not processed by their insertion order but according to the priority assigned to each of them (the one with the highest priority is processed first). A priority queue can be implemented as a linked list but more efficiently as a heap. Details on the construction and basic operations of a heap can be found in [19]. In this paper, the priorities are assigned to MU tuples according to the coefficient of criticality (CoC) defined by:

\[
\text{CoC}_{ij}^{(k)} = \left| \det \Omega_{ij}^{(k)} \right|
\]  

where \( \Omega_{ij}^{(k)} \) is the submatrix of \( \Omega \) considering that the \( i \)th \( k \)-tuple of MUs is unavailable. The CoC is a metric related to the
proximity to unobservability. In a priority queue, the candidate tuples with low CoC values (≈ 0) are of a higher priority since their unavailability reveals imminent unobservability.

If the system turns unobservable with the unavailability of the $k$-th tuple, then $CoC^{(k)} = 0$.

Hash table is an efficient data structure for the implementation of dictionaries, i.e. it stores data (array format) in an associative manner, establishing an index (obtained from a key converted employing a hash function) associated with the desired element to be stored. A hash table compiles unordered unique key-value pairs. This data structure can be used to store the $Cs$ already identified during the critical analysis. In doing so, the idea is to avoid the objective function re-evaluation for subtuples of $Cs$, which is a costly computational task.

### 5.7 Uninformed and informed search schemes

The DFS and BFS are blind search strategies since they do not consider additional information on the distance from the current tree node to that one established as a goal. Differently, the informed search scheme aims at reducing the process of search by selecting the nodes for expansion diligently. Thus, this type of search needs a way of evaluating the chance that a given node (the most promising) is on the solution path, which can be done customarily using a heuristic function. The cost of nodes is stored in a priority queue. Greedy Best-First and A* (A star) are examples of informed searches [21].

A scheme that considers a priority queue data structure, built according to CoC values, is tested in Section 6.2, to illustrate the use of an informed search.

### 5.8 Pruning rules

Aiming at reducing $S$, an intelligent tactic to be explored is to incorporate the information on the $Cs$ previously identified in the search process. When present in the measuring system, the identified $Cs$ are the easiest to be used in an uninformed search as a pruning rule [14]. As stated in Section 4, critical tuples of one element do not involve combinations (they are entirely uncorrelated), being identified in advance by rows/columns of zeros of the matrix $\Omega$. The use of previous information on the existing $Cs$ is tested in Section 6.3.

### 5.9 Criticality evaluation

Usually, the critical condition of elements that take part in the WLS estimation process is determined by the evaluation of the determinant of matrix $G$ ($n$-dimensional), as stated in Section 4. If $\det G = 0$, then the criticality is confirmed. However, instead of $G$, the residual covariance matrix $\Omega$,—composed of the rows/columns associated with a group of $t$ elements whose criticality is under analysis—can be adopted in Equation (7). Note that the critical conditions to be tested are straightforwardly represented in the covariance matrix. In addition, $\det \Omega$ ($m$-dimensional) does not need to be computed; only the determinant of the reduced matrix $\Omega_2$ ($i$-dimensional, $i < m$) is calculated. In CA, the performance of the B&B algorithms to identify $Cs$ through both the Gain and covariance matrices—i.e. $f(G)$ and $f(\Omega)$—is evaluated in Section 6.4.

### 5.10 Objectives of specific interest

The execution time is usually considered the cornerstone of CA, leading B&B tactics to be orientated towards this objective. A line of reasoning is to combine techniques evaluated as viable. In this sense, for example, one can build the following strategy: A DFS with a hash table as the data structure containing complementary information on criticalities, plus the previous knowledge on the presence of $Cs$, and the $f(\Omega)$ formulation. The effective use of these ideas is found in Section 6.5.

In some cases, the division of a large power network into areas or zones may be of practical interest, being defined in application-specific ways or by individual company decisions. CA becomes facilitated in such cases since a reduced $S$ is formed. For instance, CA is beneficial when the interest falls on the reinforcement of a measuring system in an area poorly observed by SE, in which measurements may be added to assure a desirable level of reliability against loss of data and capability to process corrupted measurements. Defining zonal study areas facilitates more efficient and effective operational/investment decisions. The task of searching for critical MUs within a radial area centred on a given bus is performed in Section 6.6.

The proposed approach is flexible enough to accommodate other specific interests; for instance, when one can be interested in identifying all the $Cs$ containing one or more priority MUs (see the test of Section 6.7).

### 6 RESULTS

This section presents the performance of the proposed B&B tactics for identifying $Cs$. Extensive simulations are carried out on a 3.4 GHz desktop, with 16 Gb RAM, Intel Core i7. The proposed B&B tactics are implemented on an application written by the authors in C++, and no optimization software package was adopted. Tests are accomplished on the IEEE118-bus system, observed in full by 99 MUs [8], gathering 176 power flow measurements (pairs active/reactive). There are no MUs at the following buses: 10, 36, 57, 58, 67, 72, 73, 87, 97, 102, 104, 107, 111, 112, 113, 115, 116, 117, and 118. The redundancy of the available measurements is $\approx 1.5$. Although only conventional measurements form the data set considered first in the simulations, the proposed methodology can deal with synchrophasors provided by PMUs, as illustrated later in Section 6.8. Bus voltage phase angle and real part of complex branch current measurements available by PMUs can be directly incorporated into the
TABLE 1 BFS and DFS—Test 1

| Type of strategy | Execution time | No. of visited solutions |
|------------------|----------------|--------------------------|
| BFS              | 2 h 48 min     | 2,157,955                 |
| DFS              | 2 h 37 min     | 2,157,955                 |

TABLE 2 No. of Cks identified (up to \( k = 4 \))—Test 1

| Cardinality | \( k = 1 \) | \( k = 2 \) | \( k = 3 \) | \( k = 4 \) |
|-------------|-------------|-------------|-------------|-------------|
| No. of Cks  | 16          | 85          | 34          | 78          |

\( P-G \) model used in CA [8]. Without loss of generality, in the simulations, the search for Cks is limited to \( k = 4 \), for visiting a tractable number of candidate solutions. To better assess this choice facing the rapid growth of the complexity of CA, consider that if BF enumeration were adopted (exploration of the entire \( S \)), the candidate Cks to be visited would reach an impressive number: almost four million. Numerical results—selected to be shown here through eight tests—are related to the following topics: DFS and BFS performance evaluation; DFS versus informed search strategy with CoC priority queue; an efficient data structure (hash table) containing complementary information on criticalities; inclusion of previous knowledge on the presence of C1s; use of Gain or covariance matrices in the criticality evaluation; a combination of strategies; given attention to specific objectives that reduce \( S \). These computational implements are first considered in separate—to evaluate their influence in the performance of CA—to be then combined in hybrid strategies. It is often difficult to envisage with reasonable certainty the efficiency of a method before its preparation and consequent application to a given problem.

There is a current tendency in favour of hybrid methods, which endeavour to benefit from the specific advantages of different approaches by combining them.

6.1 Test 1—BFS vs DFS

Initially, the performance of BFS and DFS strategies are evaluated in performing CA. The results achieved in this test with the BFS search are the same found in [14], reproduced here for the sake of completeness. They are used as a reference for comparisons throughout the present study. Table 1 establishes the performance of BFS and DFS. Note that the number of visited solutions with BF enumeration, considering the 99 MUs and Cks of cardinality up to 4, is given by the following binomial calculation:

\[
\binom{99}{1} + \binom{99}{2} + \binom{99}{3} + \binom{99}{4} = 3,926,175
\]

The number of Cks identified by both BFS and DFS are shown in Table 2. No Ck (up to \( k = 4 \)) is missing or was incorrectly identified (BF method confirms these results).

As can be seen, B&B has visited 45% fewer candidate solutions than BF, which serves to corroborate the effectiveness of the proposed approach. Also, DFS consumes 6.5% less time to be executed than BFS, which can be justified by the additional time destined to access the higher amount of memory required by BFS. Although the execution time is not a requisite to be taken individually, with the results obtained in Test 1, it is reasonable to adopt the DFS strategy over the BFS (adopted in [14]) in the next tests.

6.2 Test 2—Informed search

6.2.1 Priority queue

The tactic of using an informed search is put into practice. The priority queue data structure, formed according to the CoC values computed by Equation (8), is considered, and Table 3 shows the results obtained. One can observe that the execution time is significantly reduced.

Another aspect of the criticality problem that arouses interest concerns knowing the distribution of Cks throughout evaluating the candidate solutions. Figure 1 shows the number of visited solutions (percent values) versus the number of identified Cks (percent of the total Cks) obtained by DFS and informed search (with priority queues).

The simulations reveal that the informed search explores \( S \) intelligently, i.e. it obtains the criticalities in a well-distributed way considering their cardinalities. This strategy leads to reduced losses, even for a limited number of visits (e.g. set accordingly with the dimensionality of \( S \)), which can be crucial when computing time is necessary to interrupt the search process.
The contrary happens with DFS. As expected, the losses are expressive since it is dedicated primarily to the identification of $C_k$s of highest cardinality (a strategy that performs the search in deep first).

### 6.2.2 Hash table

The DFS with a Hash table is a strategy that uses a data structure containing further information (on previously identified criticalities) to avoid the unnecessary evaluation of the objective function. Thus, this strategy cannot be formally considered as an informed search; such information is not directly used to abbreviate (through intelligent choices) the search for selected nodes for expanding the search tree. Here, the hash table containing the already identified $C_k$s is implemented in the DFS.

Table 4 presents the results obtained using the information stored in a hash table. There is a reduction of $\approx 24.4\%$ in the DFS execution time as compared with the BFS, which is a promising result, considering that more sophisticated hash table schemes can be formulated.

### 6.3 Test 3—DFS with C1s

One adopts here the DFS strategy, in which it is known a priori the existing C1s among the available MUs. Table 5 contains the results obtained with this strategy versus the BFS. There is a marked reduction ($\approx 34\%$) of the execution time.

### 6.4 Test 4—DFS with $f(\Omega)$

Conventionally, criticality conditions are evaluated by checking whether $G$ is non-singular as in [14]. Alternatively, the matrix $\Omega$ can be used, replacing $G$ in Equation (7), where the optimization problem handled in the paper is formalized. In this test, the performance of both formulations is evaluated, adopting $f(G)$ and $f(\Omega)$, as shown in Table 6. The execution time of the DFS with $f(\Omega)$ strategy is very favourable. Hence, this strategy is adopted in the next tests.

### 6.5 Test 5—DFS with a hybrid strategy

At this point, it is natural to try a combination of strategies, aiming at better performances of B&B. Therefore, it is proposed here the combined use of three topics: Hash table data structure, previously identified C1s, and $f(\Omega)$ formulation. Table 7 shows the results obtained with this hybrid strategy in DFS. An additional reduction of $\approx 23\%$ is observed in the execution time of the hybrid strategy concerning DFS, together with $f(\Omega)$.

### 6.6 Test 6—CA in a region of interest

Power networks are commonly divided into study areas (preserved with full detail) for analysis and control, especially when the system scale is very large. The B&B tactics described so far are flexible enough to accommodate their application in a region of interest of a large-scale power grid. In this line, consider that a region of interest is selected in the network of the IEEE 118-bus test system, centred at bus #49, chosen for being the most connected. The limits of this region are defined by adjacency, being extended to the 3rd neighbourhood of bus #49. The buses in the 1st neighbourhood refer to those connected directly to the central bus. The 2nd neighbourhood comprises 1st neighbourhood buses and those connected to them, and 3rd neighbourhood incorporates the buses adjacent to the 2nd neighbourhood buses. Figure 2 depicts a network region of 46 buses surrounding bus #49 and the power flow measurements (represented by the symbol ●).

The 1st neighbourhood—delimited by a dashed curve that defines its border—is centred at bus #49, with 9 buses around in, bounded by the 2nd (+18 buses) and 3rd neighbourhoods (+18 buses).

Table 8 presents the results obtained with the application of B&B, DFS using $f(\Omega)$, to perform CA in the region described in the test. Considering the 1st neighbourhood, the execution...
time to perform this analysis is about 59 s. If the region is extended to the 2nd and 3rd neighbourhood, the results are also encouraging; the DFS algorithm requires a little more than 2 and 3 min, respectively, to be executed.

Concerning the practical impact of CA in the reliability of SE results, consider, for instance, the 1st neighbourhood of bus #49. No C1 is identified there, which means that, in terms of observability, this part of the measuring system is \((m-1)\) secure. Besides, 8 pairs, 15 trios, and 6 quartets of MUs are pointed out as critical. Take, for example, the C2 (49,51) of MUs.

Additionally, \((P_{49,51}, P_{51,52}, P_{51,58})\) form a C3 of measurements. If a problem occurs with the MU at bus #51 (e.g. RTU failure), such that \(P_{51,52}\) and \(P_{51,58}\) measurements are corrupted, SE is incapable of identifying which measurement(s) of the critical trio is(are) spurious. An even more difficult situation appears in the 2nd neighbourhood, involving the MU at bus #68. If \(P_{68,116}\) is corrupted, then this BD will not be detected, since the MU at bus #68 is critical (C1). The CA results can be explored to help the allocation of new measurements targeting the elimination of these identified criticalities.

6.7 Test 7—CA involving selected MUs

Suppose now that, instead of a region of the network, MUs of specific interest (e.g. those having a higher failure rate) are selected. For the sake of illustration, consider situations in which one, two, or three MUs are of concern, corresponding to the following sets arbitrarily chosen: Set #1 = \{50\}; Set #2 = \{50, 69\}; Set #3 = \{49, 50, 69\}.

Table 9 summarizes the results obtained, adopting a search strategy guided by a priority queue of MUs. According to these results, when the search is orientated towards a few specific MUs, the B&B execution time falls substantially—it is reduced to the level of a few minutes. This is an encouraging result since it evinces that the B&B framework can admit the parameterisation of the problem of searching for \(C_k\)s with specified MUs.

CA indicates that the MU of Set #1 is present in two C2s—\((50, 49)\) and \((50, 56)\). When considering Set #2, twelve C3s also appear, with MU #69 taking part in all of them. In the case of the MUs of Set #3, two more C2s are identified—i.e. \((49, 51)\) and \((49, 56)\)—together with seven more C3s and eighteen C4s, all of them containing MU #49.

6.8 Test 8—CA involving PMUs

Consider again the IEEE 118-bus system previously adopted, in which are now aggregated 10 PMUs (plus the existing 99 UTRs), placed at buses 2, 5, 9, 11, 12, 17, 21, 25, 28, and 114. Each PMU provides line current and bus voltage synchrophasors, as in [22].

Next, Tables 10 and 11 present the CA results (\(C_k\)s up to \(k = 4\)) with the application of B&B with the hybrid strategy on the measuring system with UTRs and PMUs. When compared with the results of Test 5, one can conclude that the execution time obtained is slightly greater (>3 min), compatible with the additional effort of combining 45 extra measurements (synchrophasors) with the 176 conventional ones (branch power flows).
Additional remarks

CA is regarded as an inherently difficult problem since its solution requires significant resources, whatever the algorithm used. The time required (or the memory space, or any other measure of complexity) to solve the CA problem is related to the size of the instance, being dependent upon: the number of network buses and its topology (i.e. the number of interconnections and how they are configured); measuring system (number, type, location of measurements); and criticality level of interest (cardinality of the tuples) which characterizes the observation degree of the system.

So far, global search algorithms of the B&B variety are the most efficient known methods for solving NP-hard problems. Roughly speaking, this means that the effort required to perform CA grows exponentially with the problem’s size. The computational complexity of B&B can be evaluated by its worst-case behaviour, representing an upper bound on its performance. However, such a measure adds little information on the implemented B&B algorithm since it can have widely different behaviour over many CA instances, considering that they are influenced by the factors mentioned above. Instead, from a practical viewpoint, this paper describes a research effort for improving the efficiency of the proposed algorithm implementation by assessing the effect of varying one of the B&B parameters at a time (e.g. DFS vs BFS strategies, different data structures, informed vs uninformed search schemes), in a well-known instance (IEEE 118-bus system [14]). After that, the paper proposes a successful hybrid strategy that combines these parameters. As a result, the time reduction to perform CA simulated on the IEEE 118-bus system was significant—from 2 h 48 min to the 10 min—which is a way of demonstrating in practice (by a numerical experiment) the algorithm efficiency. The test system adopted in the paper is amply used in SE practice (by a numerical experiment) the algorithm efficiency.

To date, only the study reported in [14] and the present work have dealt with the general combinatorial optimization problem of searching for all the critical \(k\)-tuples of MUs serving SE. A closely related work can be found in [12], which formulates the CA problem as a mixed-integer linear programming (MILP) for the computation of critical tuples of measurements of higher cardinalities (\(k > 3\)). However, that reference has addressed a particular problem: to find the sparsest (i.e. minimum cardinality) critical \(k\)-tuples involving at least one arbitrarily specified measurement.

The approach proposed in [12] is restricted since not all the existent \(C\)s are identified, notably, when the measuring system offers low redundancy, a situation in which one should expect quality solutions from CA, not provided by that approach in this case. Besides, [12] is concerned with counting \(C\)s of measurements, giving little attention to other important aspects, e.g. which measurements are present in a given \(C\), the network location a \(C\) occurs, and the chance of losing a \(C\). All these aspects are considered in the approach proposed here, as can be ascertained from the illustrative example given next.

Consider that CA analysis has been performed on the entire network of the IEEE 118-bus system, observed as a whole by 99 MUs, providing 176 power flow measurements (the same measuring system used in the tests 1–7). Owing to space limitation, the results selected to be shown here refers to the surroundings of bus #100, depicted in Figure 3. CA has identified six C2s and two C3s involving the arbitrarily specified measurement \(P_{100-106}\), as can be seen in Table 12. Although this measurement is present in six critical pairs, the most important fact is its presence in the critical trio \((P_{100-106}, P_{100-103}, P_{100-104})\). Note that if the critical MU #100 becomes unavailable—an event more likely to occur than the unavailability of C2s involving measurements from several MUs in the surroundings of bus #100—the system observability is lost. On the other hand, if a malfunction of MU #100 occurs, leading to the corruption of at least two of the critical trio \((P_{100-106}, P_{100-103}, P_{100-104})\)—also

![FIGURE 3 IEEE 118-bus system—around bus #100](image-url)
likely to occur—it would render BD identification impossible. In contrast to the approach presented in [12], all the sparsest \( C_k \)s were identified here by the proposed B&B algorithm.

Another point to mention is the family of algorithms, usually called metaheuristics (e.g. variable neighbourhood search (VNS), greedy randomized adaptive search procedures (GRASP), among others), that combine heuristic methods in higher-level frameworks, aimed at efficiently exploring the set of feasible solutions (search space) of a given combinatorial problem. These algorithms can find a reasonable number of suboptimal solutions. For this reason, they are considered out of the scope of this paper.

As mentioned earlier, the cyber-security analysis of power networks indicates that SE can be exposed to malicious data attacks, i.e. those perpetrated by a smart intruder who inflicts BD intentionally, not by chance as BD habitually occur [23]. The cyber-system intruder mainly attempts to maximize the impact of the attack and evade detection. In this scenario, \( C_k \)s of larger cardinalities (\( k \geq 3 \)) gain importance in BD processing studies since they can be considered as potential targets for sophisticated attacks.

Ending here with a brief comment related to the dimensionality of the CA problem seems to be opportune.

With the advent of networking, computing has evolved to distributed computing, with enormous value in solving the scalability problem, driving, for instance, the research and development of decentralized, efficient power network monitoring applications [24]. Large-scale power systems are typically composed of interconnected subsystems or control areas. Many studies on SE have been conducted exploring the local nature of the problem, e.g. hierarchical, decentralized, parallel and distributed SE. Power network typically has a sparse structure (buses with few pairwise physical connections) irrespective of the system size, i.e. the average number of branches incident to a bus remains practically constant, which contributes to the local nature of the SE problem [25]. When large-scale power networks are decomposed into areas of interest, it is known that the observability of one area is independent of the observability of other areas.

Similarly, segmented CA can be carried out in large-scale systems, as illustrated in the paper by Test 6, in which a region of interest is defined. The critical tuples identified in this region by CA performed locally (Table 8) are the same as those identified when the CA considers the system as a whole. Besides, Test 7 involving selected MUs (Table 9) and the simulation considering a specified measurement (Table 10) evince CA’s local nature. With these comments in mind, one admits that the test system adopted in the paper (amply used in SE studies) can be considered a sub-network of interest of a large-scale system in which CA is performed. Distributed SE, for instance, is accomplished following this line of reasoning.

It is hoped that the aspects involving criticality and observability presented in the paper can be easily assimilated and that SE researchers develop other efficient forms of computing to be applied to quantify observability.

7 | CONCLUSIONS

This paper provided tactics for attaining advances towards a better knowledge of the combinatorial optimization problem characterized by the criticality analysis to be performed routinely in power system state estimation.

The proposed B&B algorithms proved to be computationally adequate to guide the search for critical elements in an immense space of solutions, whose bounds are defined not only by the size of the power network (number of buses and how they are connected) but also by the characteristics of the measuring system and the observation degree specified. As an exact optimization method, once established the criticality level of interest, all the critical measuring devices are identified by the proposed technique, which can be confirmed, for instance, by the brute force method.

The use of specialized data structures, particularly the priority queue sorted by the proposed coefficient of criticality, reduced the execution time of the B&B algorithm radically. The simulations revealed that the informed search explored the solution space judiciously, obtaining the critical tuples in a well-distributed way, which allowed to interrupt the search process early (when computing time needs to be limited) without significant losses in the criticality identification process.

The best performance of B&B achieved in the simulations was with the tactic of combining three points (hybrid strategy): The use of the Hash table data structure, previously identified single critical elements, and evaluation of the objective function using the residual covariance matrix. The computation time reduction—obtained with this hybrid strategy to perform the criticality analysis (completed up to critical quadruples), simulated on the IEEE 118-bus system—was significant as compared with the time consumption in a naive B&B algorithm found in the literature.

Finally, it was shown that the proposed approach is very flexible, capable of being applied to sub-networks of a large-size system or selected measuring units.

Nomenclature

\[ \mathbf{\theta} \quad (n \times 1) \text{ vector of bus voltage phase angles} \]

\[ \mathbf{\Omega} \quad (m \times m) \text{ residual covariance matrix} \]

Acronyms

B&B branch-and-bound
BD bad data
BF brute force
BFS breadth-first search
CA criticality analysis
\( C_k \) critical tuple of cardinality \( k \)
CoC coefficient of criticality
DFS depth-first search
\( F \) objective function

For the reader’s easy reference, this section presents the notation and acronyms used herein. When necessary, other symbols will be introduced in the text.

\[ \mathbf{G} \quad (n \times n) \text{ Gain matrix} \]
H \((m \times n)\) Jacobian matrix

IED intelligent electronic device

M no. of measurements

MU measuring unit

N no. of state variables in the power-angle (P-\(\theta\)) model

PMU phasor measurement unit

R \((m \times 1)\) measurement residual vector

RTU remote terminal unit

\(S\) search space

SE power system state estimation

T\((k)\) \(k\)-tuple of \(S\)

V \((m \times 1)\) vector of measurement errors

WLS weighted least squares

\(z\) \((m \times 1)\) vector of measurements

REFERENCES

1. Filho, M.B.D.C., et al.: Roots, achievements, and prospects of power system state estimation: A review on handling corrupted measurements. Int. Trans. Electr. Energy Syst. 29(4), e2779 (2019)

2. Monticelli, A.: Power system state estimation: A generalized approach. Kluwer, Amsterdam (1999)

3. Abur, A., Gómez-Expósito, A: Power System State Estimation: Theory and Implementation. Marcel Decker, New York (2004)

4. Korres, G.N., Contaxis, G.C.: Identification and updating of minimally dependent sets of measurements in state estimation. IEEE Trans. Power Syst. 6(3), 999–1005 (1991)

5. S. Costa, et al.: Qualitative methods to solve qualitative problems in power system state estimation. IEEE Trans. Power Syst. 5(3), 941–949 (1990)

6. London, J.B.A., et al.: Analysis of measurement-set qualitative characteristics for state-estimation purposes. IET Proc. Gen. Transm. Distrib. 1(1), 39–45 (2007)

7. Augusto, A.A., et al.: Low-cardinality critical \(k\)-tuples in measurement sets for state estimation. In: Proceedings of the 10th IEEE PES PowerTech Conference. Grenoble, France. pp. 1–6 (2013)

8. Filho, M.B.D.C., et al.: Quantifying observability in state estimation. IEEE Trans. Power Syst. 28(3), 2987–2996 (2013)

9. Clements, K.A., Davis, PW: Multiple bad data detectability and identifiability: A geometric approach. IEEE Trans. Power Delivery. PWRD-1(3), 355–360 (1986)

10. Crainic, E.D., et al.: Power network observability: The assessment of the measurement system strength. IEEE Trans. Power App. Syst. 5(4), 1267–1285 (1990)

11. Castillo, E., et al.: \(m-k\) robust observability in state estimation. IEEE Trans. Power Syst. 23(2), 296–305 (2008)

12. Sou, K.C., et al.: Computing critical \(k\)-tuples in power networks. IEEE Trans. Power Syst. 27(3), 1511–1520 (2012)

13. Augusto, A.A., et al.: Probabilistic assessment of state estimation capabilities for grid observation. IET Proc. Gener. Transm. Distrib. 10(12), 2933–2941 (2016)

14. Augusto, A.A., et al.: Branch-and-Bound guided search for critical elements in state estimation. IEEE Trans. Power Syst. 34(3), 2292–2301 (2019)

15. Land, A.H., Doig, A.G.: An automatic method of solving discrete programming problems. Econometrica 28(3), 497–520 (1960)

16. Brusco, M.J., Stahl, S.: Branch-and-Bound Applications in Combinatorial Data Analysis. Springer, Berlin (2010)

17. Morrison, D.R., et al.: Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. Discrete Optimization 19, 79–102 (2016)

18. Conforti, M., et al.: Integer Programming. Springer, Berlin (2014)

19. Cormen, T.H., et al.: Introduction to Algorithms. MIT Press, Cambridge, MA. 3rd ed. (2009)

20. Neapolitan, R.E.: Foundations of Algorithms. Jones & Bartlett Learning, Sudbury, MA. 5th ed. (2015)

21. Russel, S, Norvig, P.: Artificial Intelligence: A Modern Approach Pearson, London. 3rd ed. (2010)

22. Filho, M.B.C.F., et al.: Enhanced bad data processing by phasor-aided state estimation. IEEE Trans. Power Syst. 29(5), 2200–2209 (2014)

23. Kosut, O., et al.: Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures. In Proceedings of the First IEEE International Conference on Smart Grid Communications (SmartGridComm), Gaithersburg, MD (2010)

24. Kouveliotis-Lysikatos, I., et al.: Towards an internet-like power grid. Journal of Modern Power Syst. and Clean Energy (2020), https://doi.org/10.35833/MPCE.2020.000154

25. Lin, Y., Abur, A.: A highly efficient bad data identification approach for very large scale power systems. IEEE Trans. Power Syst. 33(6), 5979–5989 (2018)

How to cite this article: Flôr, VBB, Filho, MBDC, de Souza, JCS., Augusto, AA. Tactics for improving computational performance of criticality analysis in state estimation. IET Gener Transm Distrib. 2021;15:1214–1226.

https://doi.org/10.1049/gtd2.12097