Journal of Physics: Conference Series

An Incomplete Observability-Constrained PMU Allocation Problem by Using Mathematical and Evolutionary Algorithms

To cite this article: Nikolaos P. Theodorakatos *et al* 2024 *J. Phys.: Conf. Ser.* **2701** 012013

Journal of Physics: Conference Series **2701** (2024) 012013 doi:10.1088/1742-6596/2701/1/012013

An Incomplete Observability-Constrained PMU Allocation Problem by Using Mathematical and Evolutionary Algorithms

Nikolaos P. Theodorakatos [1*], Rohit Babu[2] and Angelos P. Moschoudis[1, 3]

1 School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), Iroon Polytechniou 9, Zografou 15780, Athens, Greece
2 Department of Electrical and Electronics Engineering Alliance University, Anekal, Bengaluru, 562 106, India
3 Department of Electrical and Electronics Engineering, University of West Attica, 122 44 Athens, Greece
*Email: nikos.theo2772002@gmail.com, or nikos277@central.ntua.gr

Abstract: The purpose of this paper is to introduce several optimization algorithms that can be used to address optimization models in the power network, where the level of observability may be either complete or incomplete. These algorithms include discrete, continuous and metaheuristic methods. Initially, the optimization problem is approached by implementing a zero-one mixed integer linear program solved by several methods, including branch and bound revised simplex and primal dual-simplex in combination with interior point algorithms. To solve the problem of depth-one-unobservability (DoOU), a nonlinear program is proposed using Sequential Quadratic Programming (SQP), Interior-Point methods (IPMs) or YALMIP\s branch-and-bound algorithm. Additionally, the paper proposes the use of metaheuristic algorithms, such as Genetic Algorithms (GAs) and Binary Particle Swarm Optimization (BPSO), to solve optimization problems under incomplete observability. The proposed algorithms are tested using simulations on IEEE standard systems to illustrate their efficiency and reliability in solving the optimization problem under partial observability. Overall, the paper concludes that these algorithms can efficiently lead to the optimum point in a reasonable runtime. Hence, this work examines the problem of putting a restricted PMUs number to make the DoOU and to give a feedback to the state estimation routine accuracy.

1. Introduction

The transformation of conventional power grids into smart grids (SGs) is a process that must begin with the installation of sophisticated monitoring equipment capable of determining the current status of the power grid in actual time. At present, the existing power systems are changed completely into futuristic power grids incorporating smart devices such as Phasor Measurement Units (PMUs) [1].

A futuristic power grid called SGs, where synchronized measurements will observe the state of the grid, will replace the existing power grids soon [1]. PMU is an advanced electronic monitoring device that is utilized in SGs to collect data for grid data-driven implementations such as protection, state estimation and control in real–time [1]-[8]. PMUs are replacing traditional measuring devices in the existing power grids and thus transforming them into Smart Grids [1]. This replacement allows a Wide

Area Monitoring System being present and enables sufficient applications such as observation, control and protection.

PMUs are posed in power system substations in which power transmission lines, generators and loads are connected [1]. The installation of the equipment, that is the selection of the substation in which the PMU will be located, is an agreement between sensible constraints and restrictions [9].

By posing a PMU on a network node, the synchrophasor devices are able to calculate both the phasor voltage and the current phasor in power lines emanating from that node [1]. In real-life, the final installation of a PMU is the decision being derived by discussion aimed at agreement, to some degree an outcome of a minimization process, such as derived by PMU minimizing process [9].

These synchronized measurements are used for Energy Management System (EMS) applications and real-time observation of the grid state [2]-[8]. The utility industries are putting in place PMUs in the classical transmission grids for the purpose of achieving the system observation in real-time [1].

At the moment, the existing transmission grids aremonitored by a PMU limited number in conjunction with traditional SCADA measurements [11]-[13].

In the near future, the existing power grids are transformed into Smart Grids, a futuristic power grid where the synchronized measurements will observe the grid state [1]. By the point in time, following the logic of massive installation of PMUs in power grids, at each node, a vast amount of cost is required for getting the purpose of real-time power monitoring [10]-[12].

For the purpose of decreasing the cost of installation of PMUs in the case of massively placing at power grid nodes, a minimum number of those devices must be optimally selected on the inside of a power grid. Those optimal PMU numbers are adequate to perform real-time observation of the power grid state estimation [2]-[6]. Minimization algorithms result in those numbers of PMUs [11]-[13].

In evolutionary algorithms, the evaluation of each individual and its fitness function is performed in coordination with the prior run of the algorithmic model, which is executed in MATLAB [14].

In contrast to this, mathematical algorithms involving continuous and discrete optimization models make a fast evaluation of the optimization function being optimized [14]. The principal concept is to adopt the mathematical algorithm and begin using a different initial point each time the iterative process is starting to get the optimal point. Several mathematical techniques have been addressed for the solution of a large number of similar technical challeges [70]-[77].

To succeed with stochastic algorithms, some randomness is followed to solve the combinatorial optimization problem which is an iterative procedure [14]. An evolutionary algorithm uses a population in a double vector to get the optimum point [14].

Integer linear programming models with binary decision values construct an iterative process to estimate the PMUs number and their optimal sites for complete observability [15]-[18].

Ali Abur et al implemented and solved the optimal PMU arrangement problem with a binary integer programming-based procedure [15]. In [18], some indices such as Bus Observability Index (BOI) and System Observability Redundancy Index (SORI) have been suggested for monitoring of the state of a power network. In this work, we found those optima points with the highest SORI.

Heuristic algorithms are utilized in the PMU allocation problem in handling all inequality constraints of the binary integer program [14], [19]-[26]. After a comparative study with BBA's metrics, it can be inferred that evolutionary algorithms solve exactly the optimal PMU localization problem in getting a global optimum non-unique and constraint point. [19]-[26].

The minimization models are stated as a binary-integer-linear program [15]-[18] as well as nonlinear (polynomial) models, for which this study brings algorithms not wasteful for optimal solutions being developed [27]-[30]. Nonlinear algorithms such as sequential quadratic programming (SQP) and interior point methods (IPMs) are adopted in [27]-[30].

An optimal solution is found to satisfy the numerical observability using a Semi-definite programming approach [31]. A fault location observability based on synchronized and traditional measurements was presented in [32]-[33]. As it is well observed and studied, the full condition of observability signifies that all network buses are fully monitored by an optimal PMU configuration

[15]. On the other hand, $depth - of - n\ unobservability$ signifies that there exist not more than $\boldsymbol{n}$ power network buses remaining unobservant by putting a PMUs number on the inside of the power grid [34]-[41].

The concept of $depth - of - n\ unobservability$ signifies unobserved states being derived by a restricted but not sufficient PMUs number within the power grid [49], [34]-[40]. The circumstances of partial observability are firstly introduced in [34].

Afterward, Gou proposed a concept under the framework of the integer program in binary nature [17]. The incomplete observability is a basic context because it delivers a systematic approach of installation of synchronized sensors, arranging gradually, to such an extent the parts of an electric graph not being monitored to be decreased until the power network will be completely monitored. The concept of $depth - of - one\ unobservability$ (DoOU) has been given in [17], [39]. The approach employed In this work utilizes graph theory as a methodology to tackle and resolve the issue of incomplete observability at level one. At the initial stage, the minimization problem is proposed as a binary integer program (BIP) to gather some benchmark optimal solutions within a zero-gap tolerance.

This label tolerance allows us to characterize these solutions as globally optimal solutions. The BIP model is executed with MATLAB solvers either black-box routine embedded in MATLAB optimization library [61]-[62], commercial packages as Gurobi [64]-[66] or MOSEK [67] or open-source optimizer codes [63].

We also follow an evolutionary standpoint to attack the BIP model [14]. Then, we transform the model into a nonlinear program. This program can be solved either with SQP [28]-[30] or IPMs [27] towards locality or with SCIP optimizer getting the global solution [48]-[50], [58]-[60]. The models are given with a 7-bus system. The entire process optimization is executed using the classical power systems of the IEEE organization [56].

In this work, the DoOU problem is revisited and formulated, and solved in the following programming way; a 0/1 integer-linear program (ILP) [17] is utilized in the first phase, and in the second phase, a nonlinear (polynomial) problem is solved by nonlinear algorithms or a branch-and-bound tree [52]. Preferable nonlinear algorithms are the interior-point methods [26] and sequential quadratic programming [27]-[29] to solve nonlinear problems [30]-[48].

Those nonlinear algorithms are able to be aware of the existence of a locally optimal solution. Those optimum points can be characterized as non-sole globally solutions after an appropriate relative to other BBA's optimal metrics [49]-[52].

As is well known, BBAs efficiently solve the BIP discovering a globally optimal solution inside a zero-gap tolerance counting a specific number of nodes where linear problems are solved towards optimality [45]-[48]. Some infeasible regions are pruned during the iterative process, the first best integral and feasible solution is found and then a globally optimal solution is attained [49]-[52].

Also, evolutionary algorithms efficiently handle the inequality constraint of the BIP in the direction of getting a globally optimal solution [13]. Then, optimal results are derived by the utilization of mathematical and evolutionary algorithms to the partial observability optimization problem [34].

The optimum points are derived after testing the mathematical models on classical electrical graphs [57]. We use benchmark power systems such involving 14-, 30-, 57- and 118 bus systems [56]. Their design data lines and edge-vertex incidence matrix of a line-diagram power network and a square binary matrix can be found in MATPOWER software [56].

Hence, the square matrix as well as the branch-to-node incident matrix [17] is well constructed and utilized in the $0/1$integer program to get the global solution within a zero-gap tolerance [39].

The work [17] can be used as a case study related to the concept of depth-one-unobservability (DoOU). The incomplete observability with degree one is stated in binary or continuous minimization models. For the thing just mentioned this study presents effective algorithms for calculation solutions during the iterative process [45]-[52].

Our purpose is to give PMU placement sets under the DoOU using mathematical and evolutionary algorithms. Performing of the suggested algorithms is examined deeply via experimental results using standard IEEE power systems [57]. Branch-and-Bound algorithms [15]-[18] and evolutionary algorithms such as Genetic Algorithms (GAs) [19]-[22] and Binary Particle Swarm Optimization (BPSO) [22]-[26] handle with efficiency the 0/1ILP constraints towards optimality [14], [30]-[48].

Each population-based algorithm counts several function evaluations that reveal a global optimal solution at the final iteration [14]. An optimum objective function is assessed and a set of solution sites related to that objective function are derived [19]-[26].

Then, the 0/1ILP model is transformed into an equivalent nonlinear model giving the same PMU numbers under the concept of DoOU [17], [34]-[40]. In any case, the PMU placement sets are fervently dependent on the nature of the algorithm and always cover the incomplete observability with one-depth-observability [17]. This work helps the power engineers to keep in mind concepts such as complete and incomplete observability based on synchronized measurements [15]-[17], [34]-[40].

Thus, the wide-area measuring system can observe the network state in real time [2]-[8]. The final remarks conclude the impact of mathematical and evolutionary algorithms to the partial observability problem solving with distance equal to one-depth (DoOU) [17], [34]-[40].

Each programming code has the factor leading success in a manner that guarantees the least linear measure between buses not being observed and monitored power network nodes during installation in phases. Consequently, in order to completely see each voltage phasor's node, the unobserved portions of the power network are steadily reduced until they are eliminated [34]. In that case, an adequate number of sensors will be installed after installation in phases [17], [34]-[40].

In the following section, the concept of full and partial observability is studied, analyzed and results are confirmed for further validation of our modeling application. This study considers unobservable areas of a power network following an installation of a limited number of PMUs.

We present a novel contribution related to a lower bound of unobservability regarding the full condition of a power monitoring. The depth-of-one unobservability is presented, examined and solved in an integer as well as nonlinear frameworks.

We use for this situation the acronym DoOU. Initially, a binary integer program (BIP) is presented and solved by technology advanced optimizer functions in the integer programming domain [45], [49]-[50]. Then, we follow an evolutionary standpoint to analyze convergence to desired outcomes and finally we transform the BIP model into a nonlinear model to get the optimality [52]. The last model is solved by SQP [28]-[30], IPMs [27] or YALMIP's branch-and-bound algorithm [68]. The nonlinear problem is the result from a transformation of a binary integer program into a continuous format [52].

The objective function consists of a polynomial function either in continuous or binary decision domain [52]. The transformation is efficient and hence optimal solutions are achieved within absolute tolerance criteria [47]. To develop the nonlinear framework, the cost function is stated as either in quadratic or linear format. Both optimization functions are minimized to get the optimality under the same multiple-choice constraint function [47].

Hence, the whole task is deeply analyzed with suitable algorithms being presented In this work. We show the robustness of our models to detect optimality even for this situation with a restricted PMUs number being present within a power grid. Some experimental results are produced by mathematical and evolutionary algorithms to prove its robustness from different algorithmic schemes. Finally, we present and conclude this study's impact with a discussion and the final remarks.

2. Concept of Complete and Partial Observability

Using actual-time measurements through the utilization of synchronized PMUs permits us to accomplish the wide-area monitoring and the observation of the system state [1]. Although it is a necessity to install PMU at every node, it's impossible to do this due to economic reasons [10].

To stay away from this fact, scientists and planner engineers have developed strategies to pose a restricted PMUs number together with traditional supervisory control and data acquisition (SCADA) measurements in order to observe the state estimator [6]-[8].

In the power grids of the future, the operator collects measurements based on PMUs [1]. Therefore, branch-and-bound algorithm is mainly used to find a solution of the binary integer linear program to declare a restricted PMUs number and optimal in unison [15]-[18]. Full observability employs the topological-based scenario [15]-[18] where a suitable mathematical or evolutionary algorithm embedded in general purpose optimization functions find an optimal solution [13]; a desired outcome with an adequate PMU in numbers at selected power network nodes.

The whole concept is executed and simulated in the MATLAB platform [61] with specialized optimizer libraries [63]-[64]. For that purpose, benchmark power systems are used to give good enough PMU devices for cases such as an existing power system with traditional measurements performing, helping in a successful execution of a linear estimation [2]-[8]. As a future target, the technology advanced branch-and-bound algorithmic scheme solves the classical integer constraint program with binary values to find an optimal solution within an absolute gap [16]. It can give the optimal solution reflecting the smallest in amount PMU devices giving a total observation of voltage phasors of each power node [15]-[17].

Partial observability concerns the optimal PMU localization when the number and the PMU positioning sites are not adequate to declare the full state of the power system [17], [34]. Thus, the PMU number and their sites are not good enough to describe the overall collection of phasor bus voltage at every node in the power grid [34]. The incomplete observability is a calculation of the imprecise distance of an unobserved node from it's observed nearby power nodes [34].

In this work, the concept of depth-one-observability (DoOU) has been implemented. According to our rigorous definition of DoOU, these instances are when all nearby network nodes must be observably connected to the power node that cannot be observed [17], [34]-[41]. The DoOU is displayed in Fig.1 [39]. As can be seen, there is one unobservable bus which is linked together with observable buses directly monitored by PMUs installed at a specific power network [39].

We study the depth of unobservability with degree one which is a measure of the distance of an unobservant network node from a neighborhood including observable buses [39]. Partial observability is presented, analyzed and some experimental results are derived based on that concept. The placement result justifies the statement in which all adjacent buses are observable and only one node is unobservable as displayed in Fig.1.

DoOU topological scenario is determined as a state in which all the neighboring power network buses of any unobservable bus must be monitored. This definition declares that it is not possible for two unobservable buses to link together, which gives a higher depth-of-unobservability on that occasion. The entire picture is given in Fig.1. This methodology guarantees a minimum distance between an unobservable node and the adjacent local area consisting of observable network buses.

In Fig.1 bus B_4 is linked together with buses B_3 and B_5 both being observable directly by PMUs posed at network buses B_2 and B_6. Also, buses B_1 and B_7 are directly observable by PMUs posed at buses B_3 and B_6 respectively [39].

Hence, bus B_4 is the only unobservable bus in the entire local area within a power transmission grid. On one occasion as studied in this work, the PMU numbers are expected to be lesser than the PMUs number needed for complete state observability. The experimental results are studied in a comparison way with those found by using a BILP [17], [39].

DoOU is declared as a circumstance where any unobservant network node is connected with a neighborhood consisting of buses being observable [17], [34]-[40]. The concept is to decrease the PMU numbers related to the PMU number needed to succeed in fully observability conditions.

We desire to illustrate the optimal PMU numbers and their sites using mathematical and evolutionary algorithms [45]-[48]. In this work, computer programming algorithms are put into effect beneath the states of separate amounts of observability. An amount of observability means how many times each power network node can be reachable by a synchronized measurements set collected by a PMU number around the power grid [1]-[2]. Following the studies, we refer to the concept of DoOU as the status where there is one bus not observable linking to two or more than two buses with phasor voltages measurements [17], [34]-[40]. The least PMU numbers for the system complete observability or depth-one-unobservability (DoOU) can be detected using binary integer program (BIP) [17], [38].

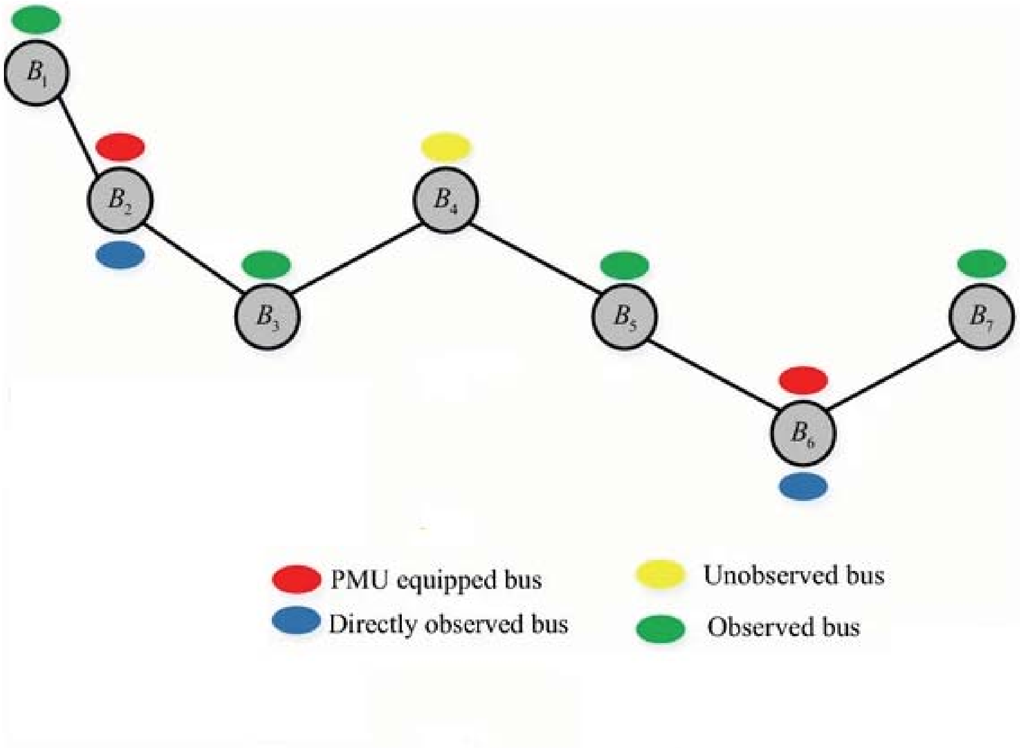

Figure 1. Incomplete observability with degree one [39]

Table 1 gives adequate PMU numbers the complete system observability and DoOU obtained by the BIP presented in [17]. The main issue of this study is that it gives only the PMUs number and not the placement sites. The simulation results displayed in the table are used as a benchmark standpoint for the utilization of mathematical and evolutionary algorithms employed to give placement sites.

As observed, a lesser PMUs number are detected than the CO [17]. It is a fact given that we state an $0/1$ integer program in which an unobservable node is adjacent with a neighborhood with observable nodes by synchronized or traditional measurements.

This mathematical based model is declared in two stages. Initially an $0/1$integer program is declared and then we transform it to a polynomial optimization model with continuous variables [47]-[48]. Hence, we use mathematical-based algorithmic schemes as well as evolutionary algorithms handling the $0/1$integer program to find PMU set solutions for the concept of DoOU [17], [39]. Each placement algorithmic scheme was tested on IEEE systems for depth-of-one unobservant region.

Table 1: The least PMU numbers required for the two observability states

IEEE Network	# Number of Branches	# PMU numbers for complete observability (CO)	# PMU numbers for DoOU
14-bus	20	4	2
30-bus	41	10	4
57-bus	80	17	11
118-bus	186	32	18

As it is proven, all the algorithms are fully convergent and give optimum points satisfying optimality conditions in discrete or continuous domains regarding to thedecision variables nature.
The leading algorithm once again for such kind optimization problems as the optimal sensor problem is the branch-and-bound algorithm to get the optimality.

This algorithm executes the 0/1 integer program under a zero-gap tolerance spending reasonably exploring nodes where relaxed problems is solved; infeasible regions are pruned and finally get a globally optimal solution at the final explored root-node of the BBA’s tree [45]-[50].

The partial observability is presented, examined, solved and its effect on the placement sites related to the resulting best objective value is evaluated. The concept of DoOU is the mainly studied topic and it is related to the distance between one unobserved bus from its observed incidence buses.

When the power network observability is diminished from full condition to unobservability with degree of one, the PMU devices are diminished in number with the trade-off of for power system reliability and security [41].

Even if there is a reduced degree of unobservability, it is still possible to optimize the process using the same method; however, the linear estimation will have a lower level of precision. This process optimization helps a lot with the incremental installation of those devices in phases within a power transmission network [40]. Note that the PMU set solutions derived by solving concepts of reduced depth of unobservant regions is outside of this study.

We examine the DoOU as a benchmark approach and a similar procedure can be used in a future work. Our impact on the PMU number and its placement related to partial observability with depth-of-one unobservant regions is summarized in the following stages. Initially, we utilize graph theory using a square matrix and an edge-vertex incidence matrix, multiplying them to introduce a suitable matrix for appropriate calculations to compute the PMU's sites throughout the execution of integer program's solvers [42], [49]-[52]. The integer model is tested on benchmark systems [56]-[57].

Genetic algorithms and binary particle swarm optimization are well performed in solving the integer-constraint-program with binary nature [13]. The performance of partial observability and its effect on the PMUs number is continued with equivalent nonlinear programming solved by sequential quadratic programming and interior-point methods or a branch-and-bound tree [45]-[49].
The evolutionary and nonlinear algorithms produce global solutions. This can be produced after a comparative study with the integer programming solver’s optimality metrics [45], [49]-[52].

Also, the nonlinear program can be solved with the SCIP optimizer function towards a globally optimum point [58]-[60]. The specific integer program’s solver linearizes the non-convexity nature, spending a reasonable number of nodes in which relaxed problems are solved. Some infeasible regions are pruned and finally a global optimum point can be found with a zero-gap tolerance [58]-[60].

We attack the nonlinear problems solved globally with precise possibility using a global nonlinear branch-and-bound (BB) algorithm included in YALMIP program [54]. It gives an optimal solution through calling general purpose optimizer functions to count the upper and lower bounds in BB tree. Finally, the cost value is the upper bound for the minimization problem studied in this paper [70].

3. Novelty, Contribution and Consideration related to implementation tasks

A different approach to the optimal PMU localization problem is the incomplete observability presented in [17], [34], and [39]. Incomplete observability means the power system state in which the PMU numbers and their locations are not enough to define the calculation of the voltage phasor at every network bus. The depth of unobservability is a quantity of the linear measure between a power network bus not observed from a local area consisting of observable buses [17], [34], and [39].

The context of power network observability can be stated as those situations where a power network is considered to be observable on occasion that the topology is given as well as measurements being obtainable [15]-[17]. At the moment, PMUs are optimally posed at selected substations and they work together with traditional measurements to achieve the wide-area data-driven monitoring [2]-[8].

Graph theory performs an important part in analytics methodologies due to its capability to present the power network topological arrangement of parts [42]-[45]. Hence, the graph theory is employed to revisit the partial observability with DoOU. We take into account the vertex-edge incidence matrix as well as a binary connectivity matrix [42]-[44]. Both matrices are based on the one-line diagram of the power system in order to build the inequality constraints of the BIP program for that purpose [50].

Each power network being implemented In this work is considered to be a multigraph without loops and multiple edges being present. A multigraph is a set of vertices and edges assigning to every edge two vertices consisting of its ends [42]. We use the multigraph to represent each IEEE power network, to form the binary connectivity matrix as well as the Branch to node incidence matrix for the implementation of the binary integer program [42]-[44].

It must be declared that multigraphs are defined them as graphs or simple graphs [42]. Incomplete observability state means the status where PMUs are installed in a restricted amount of number at selected nodes. The context of depth-one-unobservability declares the linear measure of a power network bus not being monitored from the closest bus being observable in an adjacent neighborhood of observable buses [17], [34]-[40].

The depth of partial observability is declared and studied at a number of concepts namely depth-of-one Unobservability, depth-of-two Unobservability and others [40]. More than one-depth-of-observability means that the accuracy is to a smaller degree when the PMUs number is posed in the state-estimator routine [2]-[7]. Hence, larger uncertainties could be calculated in getting the state of power monitoring at present including traditional measurements and measurements collected by a restricted PMUs number within a power grid [17], [34]-[40].

In this work, a graph-theory is used for PMU installation based on DoOU. For that purpose, mathematical and evolutionary algorithms are used. It utilizes the edge-vertex incidence matrix of a graph representing a standard IEEE power network as well as a square matrix with dimension equal to the number of power network nodes [42].

Each IEEE power network is a simple graph which means that the implementation is easy in the MATLAB [61]. The data line connections between the graph nodes can be found in MATPOWER [54]. Hence, the square binary connectivity matrix as well as the edge-vertex incidence matrix is formulated in the MATLAB platform to engage appropriate calculations [61].

Then, an integer program with binary decision can be executed giving a PMUs number sufficient of the partial observability scenario with depth-one [17], [34]-[40]. This PMUs number can be used as a benchmark solution for the purpose of the implementation of the evolutionary algorithms to our study [13]. Evolutionary algorithms handle efficiently the integer program’s constraints giving an alternative standpoint getting the optimum point [13].

Then, the entire integer program is transformed into nonlinear models seeking out locally optimal solutions which are also globally solutions [45]-[52]. Then, this minimization problem is transformed into an equivalent nonlinear programming model [30]-[48]. The nonlinear program is implemented by using Sequential Quadratic Programming and Interior-Point methods getting the optimality [30]-[48].

Also, metaheuristic algorithms are studied for partial observability with level one by proper handling of the constraint integer linear programming [13]. On the other side, we declare the concept of unobservability in a 0-1 polynomial problem being implemented in YALMIP platform [69].

The remarkable remark of this study is that GAs and BPSO succeed in finding a global optimal solution after a trial-and-error process optimization [13]. The concept of depth-of-one-observability (DoOU) state leaves a number of power network nodes unobserved meaning unobserved regions within the power grid [17], [34]-[40].

We utilize mathematical and derivative free algorithms as well as evolutionary algorithms to study each impact to the convergence to an optimum point; a global solution point [45]-[50]. The feasibility of our suggested algorithms has been validated via experimental tests with standard transmission grids as their connectivity data-lines are found in [54]-[57].

4. Application Tasks in Process Optimization Model

We strictly declare DoOU as those situations where all adjacent network nodes must be observably connected to the power node not being observable [17], [39]. DoOU formulation presents the most excellent estimate at this depth where the unobserved bus is linking to observed neighboring buses.

The depth of partial observability calculates the linear measure of a power network node not being observable from its adjacent observed nodes [17], [34]-[40]. Based on this concept, the depth-of-one unobservability (DoOU) is studied and some numerical results are derived towards optimality.

The decisive point is to permit us to estimate the buses not observed by PMUs underlying the partial-observability using synchronized measurements [17], [34]-[41]. The power network can bemonitored by a PMU limited number in conjunction with traditional measurements being derived by the SCADA [4]-[5]. This work studies the DoOU to give us a restricted PMUs number and the condition of buses not observable considering the neighboring observable buses [17], [34]-[41].

Greater than one-depth-of-observability leads to less accuracy in state-estimator process and larger uncertainties to compute the current power network state based on synchronized and traditional measurements [2], [34]-[40]. Also, we intend to do a concept to achieve where a restricted number of PMU numbers can be selected to be placed at network buses [17], [34]-[41]. Then a larger PMUs number can be added to attain complete observability for a power transmission grid [17], [34]-[41].

The PMUs are placed in power system substations to which transmission lines, loads, and generators are connected. Zero-injection (ZI) buses are not taken into account in this work [10].

Power network buses are mainly substations in actual life, and on the inside consumption of the substation is a fact that we cannot ignore it [10]. In the case of CO, we find optima by which we know all the voltage phasors of power network nodes [15]-[17], [19]-[31]. Hence, the CO condition is totally known for a successful run of the state estimator routine [2].

In the second stage, we find an optimum placement set in which an unobservant power node is adjacent with a neighborhood of fully observed nodes by synchronized measurements. Such an incomplete status, the figure 1 depicts the whole situation with an unobservable bus to be linked together with a neighborhood of observable buses [39]. In this work, we analyze the performance of mathematical and heuristic algorithms for process optimization in partial observability using PMUs.

Mathematical and population-based algorithms appear to be functional for the purpose of solving on/off optimization problems in this area of Smart Grids (SGs) monitoring state [1].

First, we examine the utilization of state-of-the-art binary integer program, which gives the exact PMUs number at specific power network nodes [50]. Then we proceed to implement the DoOU scenario [17], [39]. As it is expected, the PMUs number for depth-of-unobservability of one (DoOU) is lesser than the PMUs number for full observability (CO) [17], [39].

Also, Genetic Algorithms (GA) and a binary particle swarm optimizations manipulate the binary integer program's constraints to give the identical PMU nubmers but at not alike locations [14], [49], [52]. Hence, an evolutionary standpoint is given for the implementation of this optimization PMU localization task. Then, we proceed with a mathematical standpoint to give an acceptable and workable optimum point at the same.

The optimization models are declared either in binary or continuous domain, for which this work adopts algorithms with efficiency for getting optimum points with adequate computational tests. To achieve this task, we suggest two frameworks, that is, a nonlinear problem with continuous decision domain and a binary polynomial model.

The first model is solved using with sequential quadratic programming (SQP) [28]-[30] and interior-point methods (IPMs) [27] to solve the nolinear problem. Both algorithms find a solution pool of local optima points over the feasible set of the nonlinear programming model. Those local optima can be characterized as globally found by using multi-start procedure [45]-[49].

On the other hand, the second model is solved using a global branch-and-bound algorithm included in YALMIP calling external technology advanced nonlinear and integer programming solvers [68].

Then, we test the applicability of alternative formulations to solve the DoOU which give an acceptable accuracy whereas the state estimation tool is executed [2]-[7]. In all cases, the solution is fully functional and agrees to those found by previous work [17], [39]. The process optimization is performed and a global optimal solution is derived for each power network [56]-[57]. In all case studies, we found a proper number of PMU that is lesser than the one delivered for CO scenario.

5. Optimum design model under the concept of complete observability and unobservability

Some well-known local algorithms in the continuous domain are adopted to solve the nonlinear problem. These algorithms are strongly convergent to non-unique local optima points; equivalent with global optima derived by solving the binary integer linear program. The optimization problem is within an interest where an objective function is minimized under a box-bound on the decision variables.

We solve it with SQP methods in the direction of getting an optimal solution [28]-[30]. Also, we can utilize IPMs to solve the model with the absence of bounds defined on the decision variables [27].

The minimization model is applied to benchmark power systems and some experimental results are presented [56]-[57]. As an innovative approach, we also use the Solve Constraint Integer Program (SCIP) to solve the nonlinear problem consisting of a linear cost function subject to a polynomial constraint function and a box-constraint defined on the decision variables [56]-[57].

SCIP linearizes each polynomial constraint whereas the integrality is relaxed to a continuous domain [58]. Afterward, an optimal solution is found within a zero-gap tolerance [58]-[60]. This gap is succeeded because the Primal and Dual Bounds are evaluated to be equal [58]-[60].

Additionally, the polynomial optimization problem can be programmed in a symbolic format using the YALMIP program [69]-[70]. YALMIP global BBA finds a global solution for such a kind polynomial problem with binary symbolic decision variables [68]-[70]. The target of this work is to deliver the value of each optimization algorithm related to its capability to find the exact optimal solution.

The optimum point is the best possible solution and the well-known optimum point as well. Hence, the global solution is achieved within a zero-gap [45]. Hence, technologies advanced solvers are adopted to solve the incomplete observability either in continuous or binary domains [61]-[67]. The optimal solutions are non-unique global optima for such a case optimization study [61]-[67].

To complete the whole picture for incomplete observability, we give a standpoint of complete observability using Gurobi [64]-[66]. A Gurobi optimizer engine gathers a solution pool either for full observability or maximum observability set solutions at a single run [64]-[66].

a. Full System Observability

For a complete picture of incomplete observability with less accuracy in state estimation, and installation in phases of PMUs if it is possible, we present again the concept of the complete observability in two stages [54]-[55]. Initially, an objective function with one criterion is declared in a $0-1$ constraint integer program to gather an OPTIMA point for full observability [15]-[17].

In a second stage, we derive the objective function with two criteria optimized by Gurobi to gather a solution pool of optimal points [64]-[66]. This tactic is also followed for accomplish the partial observability with degree one.

The power system is fully observable if the voltage phasor is calculated for each node directly or indirectly [39]. The prevailing design formulation is a constraint binary integer programming related to the optimal PMU localization problem. For the n-bus power grid, the model formulation is as follows [15]-[17]:

$$min \sum_{i=1}^{n} w_i x_i = \sum_{i=1}^{n} x_i, (w_i = 1, i = 1 \dots n) \quad (1)$$

$$s.t. \quad A \cdot \vec{x} \geq \hat{1} \quad (2)$$

Where $\vec{x} = (x_1, x_{2,}, \ldots, x_n)^T$ is a binary design variable vector, whose aspects are declared in the current ***yes or no*** decision investment [47].

$$x_i = \begin{cases} 1 & \text{if a PMU is installed at bus i} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

Let us define the connectivity matrix of the power network with binay elements as follows [15]-[17]:

$$a_{ij} = \begin{cases} 1 & \text{if } i = j \text{ or } j \in \mathcal{A}_i \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

The PMU arrangement problem is declared based on two statements being minimized under the concept of full observability [54]-[55]. The achievement of maximum observation of synchronized measurements is also studied. The declaration of this objective function is as follows [54]-[55].

$$min\left(w^T - \frac{1}{n}e^T \cdot A\right) \cdot \vec{x} \tag{5}$$

$$A \cdot \vec{x} \geq \vec{1} \tag{6}$$

$$\vec{x} \in \{0,1\} \tag{7}$$

Where $\vec{w} = (w_1, w_{2,}, \ldots, w_n)_{n*1}$ w_i is the price of the PMU installed at bus i; the optimal model can be implemented by Gurobi' integer solver. The two optimization models are implemented by using the Gurobi solver towards finding a solution pool for both case studies.

Gurobi ILP solver figures out a global optimum point within a zero-gap tolerance whereas the iterative process is terminated with a satisfied relative gap [64]-[65]. The mixed integer solver returns with an optimal solution when the gap between the upper and lower bounds on the cost function is less than by the default setting equal with 1e-4. The relative gap is declared as follows [64]-[66]:

$$\text{relative gap} = 100 \times |\text{Zp} - \text{Zd}|/\text{Zp} \tag{8}$$

Zp is the primal bound on the objective, that is, the incumbent objective value, which is the upper bound for minimization problem while Zd is the dual bound that is, the lower bound for minimization models [64]. If the gap closes to zero, a global optimal solution is achieved [45].

b. Incomplete System Observability

This study suggests the context of partial observability to get a restricted PMUs number. The DoOU is stated, studied and solved using a mathematical-based optimization model. We visit the partial observability with purpose to have an acceptable accuracy whereas the state estimation routine runs [2].

Our impact on the PMU number and its placement related to partial observability with depth-of-one unobservant regions is summarized in the following phases. Initially, using graph theory an edge-vertex incidence matrix and a square matrix and, multiplying them to introduce a suitable matrix for appropriate calculations to compute the PMU's sites throughout the execution of integer program's solvers is carried out.

A graph $G = \{V, E\}$ consists of objects $V = \{v_1, v_2, ..\}$ called vertices, and another set $E = \{e_1, e_2, ..\}$, whose elements are called edges such that each edge e_k is recognized with an unordered pair $\{v_i, v_j\}$ of vertices [42]-[44]. The vertices $\{v_i, v_j\}$ associated with the edge e_k are called the end vertices of e_k [42]-[44]. The most frequent graph representative is by means of a diagram, in which the vertices are depicted as points and each edge a line section joining its end vertices [42]-[44].

Let G be the graph with n vertices, $e$edges, and no self-loops [42]-[44]. Given a graph $G$of n vertices and $m$arcs, the incidence matrix of G is a matrix $(n \times m)$ [42]-[44] such as:

$$c_{ij} = \begin{cases} 1\ if\ x_i\ is\ the\ initial\ vertex\ of\ arc\ a_j \\ -1 if\ x_i\ is\ the\ final\ vertex\ of\ arc\ a_j \\ 0\ if\ x_i\ is\ not\ ter\min a\,l\ vertex\ of\ arc\ a_j \end{cases} \tag{9}$$

If $G$is a nondirected graph then the incidence matrix is defined as follows: all entries of -1 changes to $+1$ [42]-[44]. Let us define, the "vertex-edge" or "node-arc" [42]-[44] incidence matrix or simply incidence matrix as of non-directed graph G [42]-[44] whose n rows match to the n vertices and the m columns to the m edges of the graph [42]-[44]: The $\left[c_{ij}\right]$ is as determined as follows [42]-[44], for a graph with n vertices, $e$edges and no self-loops [42]-[44]:

$$c_{ij} = \begin{cases} 1\ if\ jth\ edge\ is\ incident\ on\ ith\ vertex\ v_i \\ 0\ otherwise \end{cases} \tag{10}$$

Since each arc is adjacent to exactly two vertices, each column of the incidence matrix contains only two elements, $0\ and\ 1$ [29]-[44]. Such a matrix is called a binary matrix or a $(0,1)$-matrix [42]-[44]. Let us define a matrix $B = \left[b_{ij}\right]$ a matrix which is transpose of matrix Cas$B = C^T$, a matrix which is a edge-vertex incidence matrix [42]-[44] or branch-to-node incidence matrix [42]-[44].

Under normal conditions, each element of the vector $A_{PMU}x$ indicates the number of times the corresponding vertex of the graph is observed by PMUs [15]-[17], [19]-[31]. The degree of unobservability of one can be formulated as a set of linear inequalities so as the element of the resulting vector corresponding to two terminals of the branch to be larger than 1 ($\geq$ 1) [17], [38]. The design variable $\vec{x} = [x_1, x_2, .., x_n]^T$ is a column vector involved in this optimization problem [46]. Each integer variable must take $x_i \in \{0,1\}^n$ for a feasible minimum point in solving the minimization model i.e., the solution guarantees the model's constraints [52].

So, the depth-of-one unobservability can be formulated in a $0/1$ integer linear framework with graph theoretic approach, using the branch to node incidence matrix of graph representing an electric network and a square matrix with the dimension of the number of vertices of the graph [42]-[44].

There is a one on each diagonal and a one in the $ij\ th$position if vertex$i$ is connected to the vertex$j$. Let A be $(n \times n)$ matrix in which $a_{ij} = 1$ if vertices $i\ and\ j$are adjacent, that is connected by an edge, and 0otherwise. This matrix is symmetric for an undirected graph [42]-[44]. The adjacency matrix of a graph $G$with n points i.e., $n$vertices and no parallel edges is a $(n \times n)$ symmetric binary matrix $X = \left[x_{ij}\right]$ over the ring of integers i.e., "vertex-node adjacency matrix" or adjacency matrix, which defines the structure of the graph [42]-[44]:

$$x_{ij} = \begin{cases} 1, if\ there\ is\ an\ edge\ between\ i_{th} and\ j_{th} vertices, and \\ 0, if\ there\ is\ no\ edge\ between\ them \end{cases} \tag{11}$$

The binary connectivity matrix, A_{PMU} as proposed in [15]-[17], results from the "vertex-node adjacency matrix" for the matrix $X = \left[x_{ij}\right]$, on which the elements along the principal diagonal of its are set to 1, *that is* $\left[\mathrm{x_{ij}}\right] = 1, \forall \mathrm{i} = \mathrm{j}$, since all the diagonal elements of the matrix are 1since every vertex is "reachable" from itself by a path of cardinality 0; the set of $x_i$which are reachable by $x_j$along a path of cardinality 1, i.e., the set of the arcs $\left(x_i, x_j\right)$ exist in the graph, as noted in [42]-[44].

Therefore, the binary connectivity matrix as formulated for solving of OPP problem is the reachability matrix $R = \left[r_{ij}\right]$ as defined in [42]-[44]:

$$r_{ij} = \begin{cases} 1\ if\ vertex\ x_j is\ reachable\ from\ vertex\ x_i \\ 0\ otherwise \end{cases} \tag{12}$$

Obviously, all the diagonal elements of $R$are 1 since every vertex is reachable from itself, by a path of cardinality 0 [42]-[44]

$$A_{k,m} = A_{PMU} = \begin{cases} 1, \text{if } k = m, \text{ or } k \text{ and } m \text{ are connected} \\ 0, \text{otherwise} \end{cases} \tag{13}$$

A graph-theoretic process is proposed to implement the partial observability by placing a restricted PMUs number around a power grid. It makes use a vertex-node adjacency matrix and an edge-vertex matrix to locate the optimal arrangement of PMUs relying on DoOU topological scenarios.

A dispersed PMUs number are placed around a power grid which guarantees the restricted linear measure an unobservable network bus with observable buses including in a neighborhood adjacent to the unobservable bus [17], [39]. The optimization model is declared in a mixed-integer program as illustrated in Eqs.(14)-(15). The seven bus system is used to show the formulation.

$$min\, J\,(x) = w^T\vec{\boldsymbol{x}} = \sum_{k=1}^{n} w_k x_k = \sum_{i=1}^{n} x_k\ (w_k = 1, i = 1 \dots n) \tag{14}$$

$$\begin{aligned} & s.t. B \times A_{PMU} \times \vec{\boldsymbol{x}} \geq b \\ & B: {''}edge - vertex{''}\ incidence\ matrix \\ & x_i, the\ design\ var\ i\ ables \\ & b = [11\dots.1]_{l\times 1}^T \\ & l\ is\ the\ number\ of\ branches\ of\ power\ network \\ & A_{PMU} \text{incidence } matrix; a\ binary\ (0-1) matrix \\ & \vec{\boldsymbol{x}}_l \leq \vec{\boldsymbol{x}} \leq \vec{\boldsymbol{x}}_u \\ & x_i \forall\ i\ int\ e\ ger \\ & s.t \begin{cases} x_l = [0\ 0\ \dots.0]^T \\ x_u = [1\ 1\ \dots.1]^T \end{cases} \end{aligned} \tag{15}$$

Let us study the partial observability, using the seven-bus system (Fig 2) [57]:

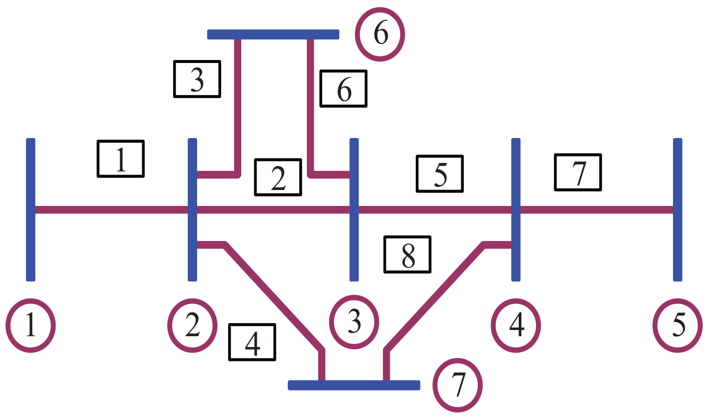

Fig. 2 IEEE-7 bus system

The graph for the "arc -node " or " edge-vertex" incidence matrix [42]-[44] for the 7-bus system is as:

$$B = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{16}$$

For the seven-bus system displayed in Fig 2, the $\mathbf{0/1}$ mixed-integer-program is as follows:

$$min J\,(\vec{x}) = \sum_{i=1}^{7} w_i x_i = \sum_{i=1}^{n} x_i, (w_i = 1, i = 1 \dots n) \tag{17}$$

$$s.t f(\vec{x}) = \begin{cases} 2x_1 + 2x_2 + x_3 + x_6 + x_7 \geq 1 \\ x_1 + 2x_2 + 2x_3 + x_4 + 2\,x_6 + x_7 \geq 1 \\ x_1 + 2x_2 + 2x_3 + 2\,x_6 + x_7 \geq 1 \\ x_1 + 2x_2 + x_3 + x_4 + x_6 + 2x_7 \geq 1 \\ x_2 + 2x_3 + 2x_4 + x_5 + x_6 + x_7 \geq 1 \\ 2x_2 + 2x_3 + x_4 + 2x_6 \geq 1 \\ x_3 + 2x_4 + 2x_5 + x_7 \geq 1 \\ x_2 + x_3 + 2x_4 + x_5 + 2x_7 \geq 1 \end{cases} \tag{18}$$

$$0 \leq x_i \leq 1 \tag{19}$$

The above convex formulation where the binary variables are involved, is minimized by the branch- and bound algorithm through the implementation of Gurobi [64]-[66]. The inequality relations are "multiple choice" constraints [50]. The MIP routine locates the vector$\vec{x} = \{0,0,1,0,0,0,0\}^T$, so as each element of $B \times A \times \vec{x} = \begin{bmatrix} 1 \\ 2 \\ 2 \\ 1 \\ 2 \\ 2 \\ 1 \\ 1 \end{bmatrix}$ corresponding to the two terminals of any edge of the graph is larger than 1 (> 1).

As can be observed, each edge of the graph is reachable by at least one observable vertex (directly or indirectly) [42]-[44]. BBA involved in Gurobi finds a solution for the above model. Gurobi optimizer solves the BILP towards optimality as the log output shows [64]-[66].

Table 2: Process Optimization being derived by Gurobi Optimizer Engine

Gurobi Optimizer version 10.0.0 build v10.0.0rc2 (win64)
Optimize a model with 8 rows, 7 columns and 41 nonzeros
Variable types: 0 continuous, 7 integer (7 binary)
Coefficient statistics:
Matrix range [1e+00, 2e+00]
Objective range [1e+00, 1e+00]
Bounds range [1e+00, 1e+00]
RHS range [1e+00, 1e+00]
Found heuristic solution: objective 2.0000000
Presolve removed 8 rows and 7 columns
Presolve time: 0.00s
Presolve: All rows and columns removed
Explored 0 nodes (0 simplex iterations) in 0.04 seconds (0.00 work units)
Thread count was 1 (of 4 available processors)
Solution count 2: 1 2
Optimal solution found (tolerance 1.00e-04) Best objective 1.000000000000e+00, best bound 1.000000000000e+00, gap 0.0000%
Elapsed time is 1.804717 seconds.
Optimal objective: 1.000000e+00
0 0 1 0 0 0 0
Elapsed time is 1.823886 seconds.
ans =
3

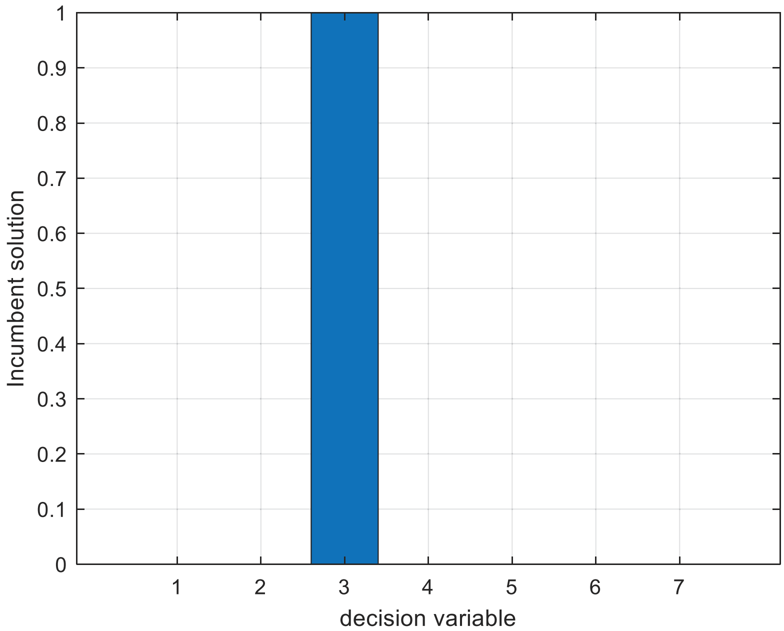

Fig.3 Plot diagram with the IEEE-7 bus system

Therefore, a PMU is installed at the bus 3 shown in Fig.3, so we have implemented the depth-of-one unobservability [15], [39]. Let us study the inequality constraints for the 7-bus network. The constraint function is based on logical methodology [47], and the logical equation $\boldsymbol{x}+\boldsymbol{x}=\boldsymbol{x}$ [51]-[52].

Based on this equation, the inequality constraint is further simplified and finally a nonlinear function is achieved for further analysis. The nonlinear program is minimized under a continuous domain to attain a local solution or a globally optimal solution via SCIP optimizer routine [58]-[60].

$$f(\vec{\boldsymbol{x}})= \begin{cases} 2x_1+2x_2+x_3+x_6+x_7 \geq 1 \rightarrow x_1+x_1+x_2+x_2+x_3+x_6+x_7 \geq 1 \\ x_1+2x_2+2x_3+x_4+2x_6+x_7 \geq 1 \rightarrow x_1+x_2+x_2+x_3+x_3+x_4+x_6+x_6+x_7 \geq 1 \\ x_1+2x_2+2x_3+2x_6+x_7 \geq 1 \rightarrow x_1+x_2+x_2+x_3+x_3+x_6+x_6+x_7 \geq 1 \\ x_1+2x_2+x_3+x_4+x_6+2x_7 \geq 1 \rightarrow x_1+x_2+x_2+x_3+x_4+x_6+x_7+x_7 \geq 1 \\ x_2+2x_3+2x_4+x_5+x_6+x_7 \geq 1 \rightarrow x_2+x_3+x_3+x_4+x_4+x_5+x_6+x_7 \geq 1 \\ 2x_2+2x_3+x_4+2x_6 \geq 1 \rightarrow x_2+x_2+x_3+x_3+x_4+x_6+x_6 \geq 1 \\ x_3+2x_4+2x_5+x_7 \geq 1 \rightarrow x_3+x_4+x_4+x_5+x_5+x_7 \geq 1 \\ x_2+x_3+2x_4+x_5+2x_7 \geq 1 \rightarrow x_2+x_3+x_4+x_4+x_5+x_7+x_7 \geq 1 \end{cases} \quad (20)$$

The linear inequality constraints can be declared in a nonlinear convex format, as an equality nonlinear equality constraint which has nonlinear structure [48], based on the consideration that these constraints are multiple choice [52]. The constraint function is written as follows [48]:

$$g(\vec{x})= \begin{cases} (1-x_1)^2(1-x_2)^2(1-x_3)(1-x_6)(1-x_7)=0 \\ (1-x_1)(1-x_2)^2(1-x_3)^2(1-x_4)(1-x_6)^2(1-x_7)=0 \\ (1-x_1)(1-x_2)^2(1-x_3)^2(1-x_6)^2(1-x_7)=0 \\ (1-x_1)(1-x_2)^2(1-x_3)(1-x_4)(1-x_6)(1-x_7)^2=0 \\ (1-x_2)(1-x_3)^2(1-x_4)^2(1-x_5)(1-x_6)(1-x_7)=0 \\ (1-x_2)^2(1-x_3)^2(1-x_4)(1-x_6)^2=0 \\ (1-x_3)(1-x_4)^2(1-x_5)^2(1-x_7)=0 \\ (1-x_2)(1-x_3)(1-x_4)^2(1-x_5)(1-x_7)^2=0 \end{cases} \quad (21)$$

The higher order terms can be simplified as and the nonlinear programming is as follows [51]-[52]:

$$min\, J(\vec{x})=\sum_{i=1}^{7} w_i x_i^2=\sum_{i=1}^{n} x_i^2, (w_i=1) \quad (22)$$

$$g(\vec{x})= \begin{cases} g_1=(1-x_1)(1-x_2)(1-x_3)(1-x_6)(1-x_7)=0 \\ g_2=(1-x_1)(1-x_2)(1-x_3)(1-x_4)(1-x_6)(1-x_7)=0 \\ g_3=(1-x_1)(1-x_2)(1-x_3)(1-x_6)(1-x_7)=0 \\ g_4=(1-x_1)(1-x_2)(1-x_3)(1-x_4)(1-x_6)(1-x_7)=0 \\ g_5=(1-x_2)(1-x_3)(1-x_4)(1-x_5)(1-x_6)(1-x_7)=0 \\ g_6=(1-x_2)(1-x_3)(1-x_4)(1-x_6)=0 \\ g_7=(1-x_3)(1-x_4)(1-x_5)(1-x_7)=0 \\ g_8=(1-x_2)(1-x_3)(1-x_4)(1-x_5)(1-x_7)=0 \end{cases} \quad (23)$$

$$\vec{0} \leq \vec{x} \leq \vec{1} \quad (24)$$

Each constraint is called as a "multiple-choice constraint", since our choice of investments is to be at least one of the available options [46]-[47]. The Weierstrass theorem ensures this even when specific statess are desired, that is,$\{x_k\}_{k=0}^{\infty} \rightarrow \vec{x}^*$**h**as a limit point in the feasible region [46]. The nonlinear programming model is defined on a closed and bounded set since it includes the boundary points.

Then, the problem attains its minimum [48]. The experimental test being optimized by SQP method is shown in the Table 3. As we can see, the step-length is equal to 1 avoiding the Maratos effect [53].

Using a totally arbitrary initial point for the purpose of starting the sequential quadratic programming, the optimum point is$\vec{x} = \{0,0,1,0,0,0,0\}^T$. So, a PMU is installed on the bus 3 to be the system "degree-of-one unobservability". Then, the minimization problem attains its minimum. Table 3&4 illustrated the optima points for the DoOU topological scenario [17], [39].

Table 3: Process Optimization being derived by MATLAB Optimizer Engine

Iter	Func-count	Fval	Feasibility	Step Length	Norm of step	First-order optimality
0	8	4.000000e+00	0.000e+00	1.000e+00	0.000e+00	2.000e+00
1	19	3.921984e+00	9.604e-05	4.900e-01	1.960e-02	1.490e+00
2	28	4.411668e+00	2.881e-05	7.000e-01	7.005e-01	1.428e+00
3	36	4.921934e+00	1.301e-14	1.000e+00	3.010e-01	1.980e+00
4	44	4.921913e+00	0.000e+00	1.000e+00	3.794e-02	1.980e+00
5	61	4.921632e+00	0.000e+00	4.035e-02	3.257e-02	1.979e+00
6	69	4.918283e+00	0.000e+00	1.000e+00	1.200e-02	1.979e+00
7	77	4.897419e+00	0.000e+00	1.000e+00	2.038e-02	1.974e+00
8	85	4.786454e+00	0.000e+00	1.000e+00	4.289e-02	3.468e+01
9	93	4.249772e+00	0.000e+00	1.000e+00	1.435e-01	3.244e+03
10	101	2.298242e+00	0.000e+00	1.000e+00	6.640e-01	2.933e+05
11	109	1.000812e+00	0.000e+00	1.000e+00	1.120e+00	3.776e+06
12	117	1.000083e+00	0.000e+00	1.000e+00	2.993e-02	3.804e+06
13	125	1.000000e+00	0.000e+00	1.000e+00	9.135e-03	3.854e+06
14	133	1.000000e+00	0.000e+00	1.000e+00	1.313e-05	2.721e+03
15	141	1.000000e+00	0.000e+00	1.000e+00	3.274e-06	2.229e+03
16	149	1.000000e+00	0.000e+00	1.000e+00	9.808e-07	9.683e-09

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the optimality tolerance,
and constraints are satisfied to within the selected value of the constraint tolerance.

<stopping criteria details>

Elapsed time is 2.352601 seconds.

ans =

0 0 1 0 0 0 0

ans =

3

struct with fields:

Table 3: Process Optimization being derived by MATLAB Optimizer Engine continued

iterations: 16
funcCount: 149
algorithm: 'sqp'
message: 'Local minimum found that satisfies the constraints Metric
Options relative first-order optimality = 4.84e-09 OptimalityTolerance = 1e-06 (default)↵relative max(constraint violation) = 0.00e+00 ConstraintTolerance = 1e-06 (selected)'
constrviolation: 0
stepsize: 9.8081e-07
lssteplength: 1
firstorderopt: 9.6829e-09

As we observe, the PMU placement set is$\vec{x} = \{0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0\}$. The PMU site is displayed in the plot displayed in Fig.2. As we see, the bus 3 satisfies the depth-first-of-one partial observability [17]. Bus 3 is linked together with observable buses by PMUs inside the power grid [17], [39]. SQP spent reasonable function evaluations leading to a local optimum point avoiding any constraint violation [61]. All optimality metrics are well-accepted and an optimal solution is reached.

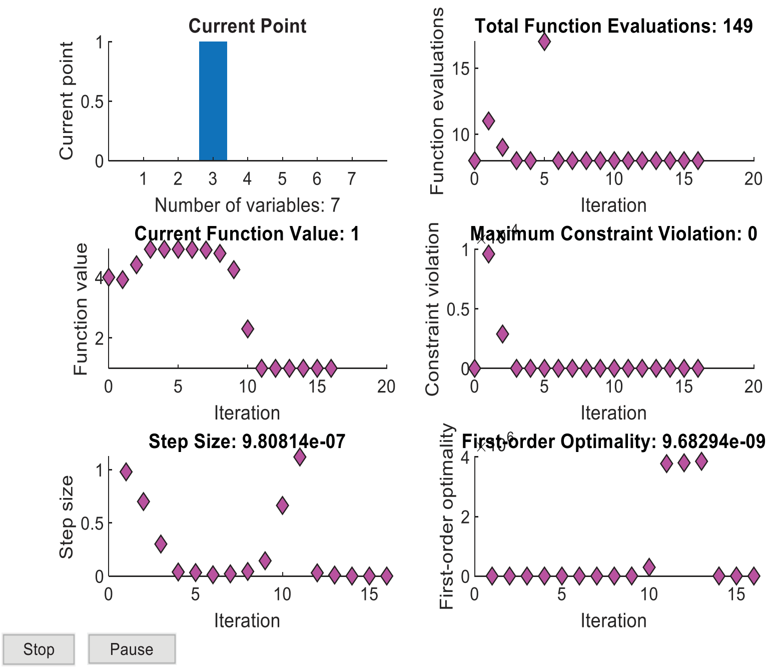

Fig.4 Plot diagram with the IEEE-7 bus system

The experimental test being optimized by IPMs is shown in the Table 4 and the plot diagram showing the convergence properties shown in Fig.5. An unbounded NLP program is solved by using IPMs towards a localized point starting from an arbitrary starting point [45]. IPMs solve the NLP model without the restriction $\vec{0} \leq \vec{x} \leq \vec{1}$ to be present [27]. An optimal solution is achieved within tolerances.

Table 4: Process Optimization being derived by MATLAB Optimizer Engine

Journal of Physics: Conference Series **2701** (2024) 012013 doi:10.1088/1742-6596/2701/1/012013

```
                                        First-order    Norm of
 Iter F-count          f(x)  Feasibility  optimality     step
    0       8  4.000000e+00  0.000e+00   2.000e+00
    1      18  0.000000e+00  1.000e+00   1.490e-08   2.000e+00
    2      26  9.999999e-01  4.000e-08   1.000e+00
    3      34  1.000000e+00  1.250e-09   7.388e-08   6.897e-08
    4      42  1.000000e+00  0.000e+00   4.470e-08   3.758e-08

Local minimum found that satisfies the constraints.

Optimization completed because the objective function is non-decreasing in
feasible directions, to within the default value of the optimality tolerance,
And constraints are satisfied to within the selected value of the constraint tolerance.
<stopping criteria details>
Elapsed time is 3.340549 seconds.

ans =

    0.0000   -0.0000    1.0000    0.0000    0.0000   -0.0000   -0.0000

ans =

     3

Optimization completed: The relative first-order optimality measure, 2.235174e-08,
is less than options.OptimalityTolerance = 1.000000e-06, and the relative maximum con-
straint
Violation, 0.000000e+00, is less than options.ConstraintTolerance = 1.000000e-16.
Optimization Metric                                    Options
relative first-order optimality =   2.24e-08       OptimalityTolerance =   1e-06 (default)
relative max(constraint violation) =   0.00e+00    ConstraintTolerance =   1e-16 (selected)

output

  struct with fields:

         iterations: 4
          funcCount: 42
    constrviolation: 0
           stepsize: 3.7581e-08
          algorithm: 'interior-point'
      firstorderopt: 4.4703e-08
       cgiterations: 0
            message: 'Local minimum found that satisfies the constraints Optimization completed:
```

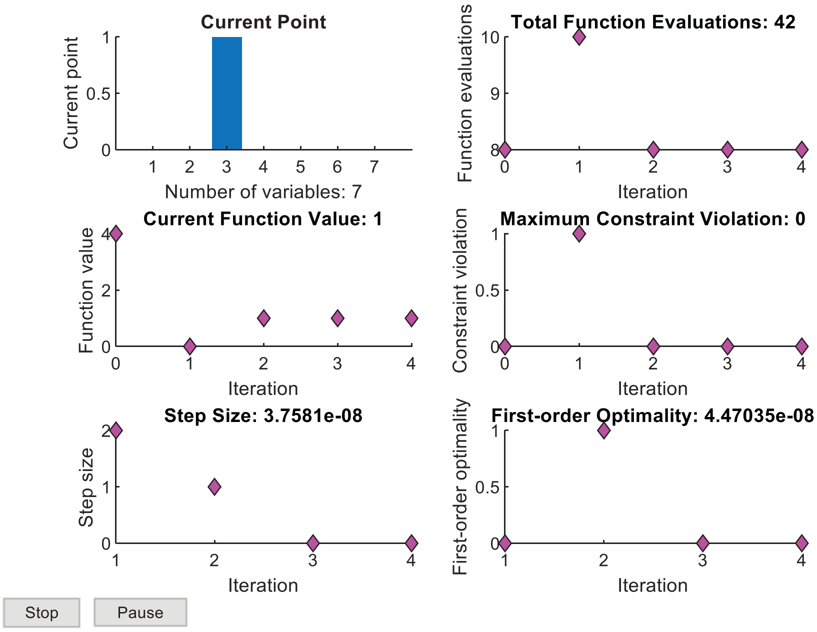

Fig.5 Plot diagram with the 7 bus system

Moreover we can implement the nonlinear model in the YALMIP program [68]. We use the YALMIP nonlinear BBA optimizer function to get an optimal solution. BMIBNB routine is called to minimize the problem in a binary symbolic decision domain [69]-[70]. To construct the binary tree, an LP solver is invoked for a suitable branch strategy [49]-[50]. The 7-bus system is used and t binary polynomial formulation consists of a linear cost function subject to a polynomial constraint whereas the decision variables are declared in a binary format as follows [47]-[49].

$$min\, J\,(\vec{x}) = \sum_{i=1}^{7} w_i x_I = \sum_{i=1}^{n} x_I, (w_i = 1, i = 1 \dots n) \tag{25}$$

$$g(\vec{x}) = \begin{cases} g_1 = (1-x_1)(1-x_2)(1-x_3)(1-x_6)(1-x_7) = 0 \\ g_2 = (1-x_1)(1-x_2)(1-x_3)(1-x_4)(1-x_6)(1-x_7) = 0 \\ g_3 = (1-x_1)(1-x_2)(1-x_3)(1-x_6)(1-x_7) = 0 \\ g_4 = (1-x_1)(1-x_2)(1-x_3)(1-x_4)(1-x_6)(1-x_7) = 0 \\ g_5 = (1-x_2)(1-x_3)(1-x_4)(1-x_5)(1-x_6)(1-x_7) = 0 \\ g_6 = (1-x_2)(1-x_3)(1-x_4)(1-x_6) = 0 \\ g_7 = (1-x_3)(1-x_4)(1-x_5)(1-x_7) = 0 \\ g_8 = (1-x_2)(1-x_3)(1-x_4)(1-x_5)(1-x_7) = 0 \end{cases} \tag{26}$$

$$\vec{x} \in \{0,1\} \tag{27}$$

YALMIP's global BBA solves the nonlinear problem involving binary decision variables towards optimality. For that purpose, external optimization functions are invoked to get the optimality [69]-[70]. The iterative process is displayed in Table 5, the PMUs sites are shown in Fig.6.

It is essential to detect an upper and a lower bound in this case, We can detect the upper bound using any local optimizer solver [61], [63]. On the other side, the lower bound is obtained through a convex relaxation or duality [61], [63]-[64]. The convex relaxations are solved through calling an external integer linear programming solver. With this manner, the YALMIP BBA detects the global optimum point in reasonable computational time. SCIP performs as a lower solver to count the lower bound whereas the FMINCON routine estimates the upper bound of the objective function value [61]-[63]. MOSEK performs as an LP optimizer solver to solve the relaxed linear problems whereas the binary tree is built for the purpose of getting an optimal solution; a global one within specific relative gap and a zero-absolute gap [67].

Additionally, the upper bound is measured by an NLP solver whereas the lower bound is evaluated using an ILP solver. The optimization problem is solved at a specific root node with zero-gap tolerance and a relative gap being optimized [69]-[70]. The zero-gap appears because the difference of those bounds is evaluated to be zero.

The upper bound is considered to be the cost value within specific tolerance gaps and the lower bound is culprit to close the gap [69]-[70]. Hence, a global optimum point is attained within specific tolerances being satisfied [46]. The optimum point is achieved within a zero-gap tolerance whereas the entire process optimization is satisfied within an acceptable relative gap equal to $5e-09\%$ [68]-[70]. Based on the above criteria, an optimal point is achieved with a 0.00% certificate of optimality [45].

Table 5: Process Optimization being derived by YALMIP Optimizer Engine

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|  ID|                    Constraint|  Coefficient range|
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|  #1|  Equality constraint (polynomial) 1x1|          1 to 1|
|  #2|  Equality constraint (polynomial) 1x1|          1 to 1|
|  #3|  Equality constraint (polynomial) 1x1|          1 to 1|
|  #4|  Equality constraint (polynomial) 1x1|          1 to 1|
|  #5|  Equality constraint (polynomial) 1x1|          1 to 1|
|  #6|  Equality constraint (polynomial) 1x1|          1 to 1|
|  #7|  Equality constraint (polynomial) 1x1|          1 to 1|
|  #8|  Equality constraint (polynomial) 1x1|          1 to 1|
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Starting YALMIP global branch & bound.
* Upper solver     : fmincon
* Lower solver     : SCIP
* LP solver        : MOSEK
* -Extracting bounds from model
* -Perfoming root-node bound propagation
* -Calling upper solver (no solution found)
* -Branch-variables : 7
* -More root-node bound-propagation
* -Performing LP-based bound-propagation
* -And some more root-node bound-propagation
* Starting the b&b process
 Node      Upper       Gap (%)      Lower      Open      Time
    1 :  1.00000E+00     0.00      1.00000E+00    2     0s  Solution found by heuristics
* Finished.  Cost: 1 (lower bound: 1, relative gap 5e-09%)
* Termination with relative gap satisfied
* Timing: 26% spent in upper solver (2 problems solved)
*            20% spent in lower solver (1 problems solved)
```

Table 5: Process Optimization being derived by YALMIP Optimizer Engine continued

* 11% spent in LP-based domain reduction (14 problems solved)
* 1 % spent in upper heuristics (1 candidates tried)
yalmipversion: '20210331'
matlabversion: '9.4.0.813654 (R2018a)'
yalmiptime: 0.2259
solvertime: 0.6941
info: 'Successfully solved (BMIBNB)'
problem: 0
Elapsed time is 0.921325 seconds.
ans =
0 0 1 0 0 0 0
Linear scalar (real , binary, 7 variables)
Current value: 1
Coefficients range: 1 to 1

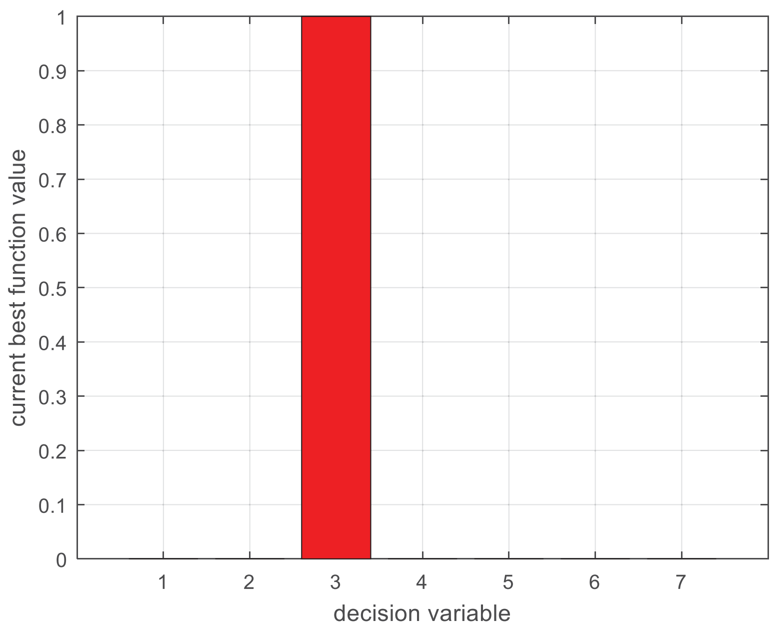

Fig.6 Plot diagram with the IEEE-7 bus system

6. Experimental Tests and Resulting Simulation Run

Complete observability (CO) is a scenario which permits us to compute the voltage phasors at all network buses [15]-[17]. In real-time monitoring, the operator can still misplace the observability if a restricted PMU numbers are deployed at selected buses [34].

Incomplete or partial observability is a topological scenario where the PMU numbers and their positioning sites are not adequate to declare the complete observability [17], [34]-[40]. The depth of unobservability calculates the linear measure of a bus not being monitored from its nearby observed network nodes [17], [34]-[38]. In this work, we work the DoOU formulation in a $0/1$constraint integer linear program as well as in a nonlinear model. Some experimental runs are executed on standard IEEE power systems, resulting in optima points being derived [54]-[57].

The binary integer linear program is easily implemented to get the global optimum solution [49]. A systematic graph study is analyzed in getting its effect on PMU numbers and its set solution with DoOU [17]. We make use of a branch to node incidence matrix representing the graph which reflects a power system concluding transmission lines and nodes [42]-[44].

Also, a square matrix with dimension equal to the number of vertices is used. Both matrices are multiplying to formulate the appropriate inequality constraint of the integer linear program in getting the optimal solution within a binary integrality$\{0.1\}$ [49]-[50]. We present the PMU placements for CO and DoOU scenarios in order to display a solution pool for both case studies [64].

For that reason, we use a powerful optimizer engine as the Gurobi to run the constraint binary integer programs for both case studies [64]-[66]. Gurobi finds with efficiency optimum points within a zero-gap tolerance; meaning that globally optimal solutions are achieved for both studies.

Gurobi is a technology advanced and powerful optimizer engine capable of solving constraint integer programs [64]-[66]. The integer linear solver embedded in Gurobi solves both case studies, and succeeds to find the minimum integral solution without this to suffer from floating points (FPs).

Each optimal solution satisfies the binary integral$\vec{\boldsymbol{x}} \in [\mathbf{0}, \mathbf{1}]\}$ without affecting the existence of FPs [48]-[52]. For the CO case study, we present optimum points which cover the power system observability and in unison the maximum observability is attained [54]. The objective function consists of a multiple-choice constraint function under decision variables to be either in binary nature or in continuous declaration. Mathematical and evolutionary algorithms optimize the design programming models with sufficient precision in finding binary optimum points [14].

On that condition, the best possible solution for each power network is considered to be the optimum point within a zero-gap tolerance. The Primal Bound is the leader to achieve the best integral and feasible solution. Constantly a Dual Bound is computed and compared to the Primal Bound.

Finally, Gurobi ILP solver founds iteratively a zero variation between them discovering a reasonable number of nodes of the binary tree; hence a globally solution was found [64]-[66].

Whereas the Gurobi ILP finds out a verified optimum point to our model specific for complete observability [15], the optimizer engine permits us to change the setting parameters [64]. Using the command namely PoolSearchMode parameter with syntax as params.PoolSearchMode=2 [64]-[66]; a solution pool of optima points are gathered [66]. All optima points are globally solutions within a zero-gap tolerance [64]-[66]. We continue this tactic with incomplete observability with degree one. We are interested in optima points within an absolute gap of the best possible solution found [45]-[50].

The solution points are illustrated as depicted in the above tables. If we set the tuning parameter equal to 2; Gurobi seeks out the best solutions following a systematic way. As output, Gurobi delivers the n best possible solutions with an indisputable gap which is defined as the absolute gap delivering a globally optimal solution [64]-[66].

Table 6 displays the iterative process delivered by the ILP solver included in Gurobi optimization library. Tables 7 &8 display the solution pool derived by Gurobi ILP solver for the purpose of getting non-unique constraint optima points [64]-[66]. Table 8 gives those optima points which are convenient with maximum measurement observability redundancy as proposed in our studies [54]-[55].

Journal of Physics: Conference Series **2701** (2024) 012013 doi:10.1088/1742-6596/2701/1/012013

Table 6: Process Optimization being derived by Gurobi Optimizer Engine

Set parameter TimeLimit to value 100
Set parameter Presolve to value 2
Set parameter PoolSearchMode to value 2
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (win64)
Optimize a model with 14 rows, 14 columns and 54 nonzeros
Variable types: 0 continuous, 14 integer (14 binary)
Coefficient statistics:
Matrix range [1e+00, 1e+00]
Objective range [1e+00, 1e+00]
Bounds range [1e+00, 1e+00]
RHS range [1e+00, 1e+00]
Presolve removed 6 rows and 0 columns
Presolve time: 0.03s
Presolved: 8 rows, 14 columns, 25 nonzeros
Variable types: 0 continuous, 14 integer (14 binary)
Root relaxation presolved: 8 rows, 14 columns, 25 nonzeros
Root relaxation: objective 4.000000e+00, 7 iterations, 0.01 seconds (0.00 work units)
Nodes | Current Node | Objective Bounds | Work
Expl Unexpl | Obj Depth IntInf | Incumbent BestBd Gap | It/Node Time
* 0 0 0 4.0000000 4.00000 0.00% - 0s
Optimal solution found at node 0 - now completing solution pool...
Nodes | Current Node | Pool Obj. Bounds | Work
| | Worst |
Expl Unexpl | Obj Depth IntInf | Incumbent BestBd Gap | It/Node Time
0 0 - 0 - 4.00000 - - 0s
Expl Unexpl | Obj Depth IntInf | Incumbent BestBd Gap | It/Node Time
0 0 - 0 - 4.00000 - - 0s
0 2 - 0 - 4.00000 - - 0s
Explored 79 nodes (49 simplex iterations) in 0.26 seconds (0.00 work units)
Thread count was 12 (of 12 available processors)
Solution count 10: 4 4 4 ... 5
No other solutions better than 5
Optimal solution found (tolerance 1.00e-04)
Best objective 4.000000000000e+00, best bound 4.000000000000e+00, gap 0.0000%
Elapsed time is 0.779711 seconds.
Optimal objective: 4.000000e+00
0 1 0 0 0 0 1 0 0 0 1 0 1 0
Elapsed time is 0.821109 seconds.
2 7 11 13
2 6 7 9
2 6 8 9
2 8 10 13
2 7 10 13

Table 7. The PMU Placement Results required for the complete observability

IEEE Networks	CO PMU location (Bus #)
14 bus system	2 7 11 13
	2 6 7 9
	2 6 8 9
	2 8 10 13
	2 7 10 13
30 bus system	1 2 6 9 10 12 18 24 25 27
	2 3 6 9 10 12 18 24 25 27
	3 6 7 9 10 12 18 24 25 27
	1 6 7 9 10 12 18 24 25 27
	2 3 6 9 10 12 15 19 25 27
	1 7 9 10 12 15 18 25 27 28
	2 3 6 9 10 12 15 18 25 27
	3 5 9 10 12 15 18 25 27 28
	1 2 6 9 10 12 15 18 25 27
	3 6 7 9 10 12 15 18 25 27
57 bus system	2 6 12 19 22 25 27 32 36 38 41 45 46 50 52 55 57
	1 4 9 19 22 25 27 29 32 36 41 45 46 47 50 53 57
	1 4 9 19 22 25 27 29 32 36 41 45 46 48 50 53 57
	1 4 9 19 22 25 27 29 32 36 38 41 45 46 50 53 57
	1 4 9 19 22 25 27 29 32 36 41 45 46 47 50 54 57
	1 6 10 15 19 22 25 27 32 36 38 41 47 49 52 55 57
	1 6 10 15 19 22 25 27 32 36 41 44 47 49 52 55 57
	1 6 10 15 19 22 25 27 32 36 41 45 47 49 52 55 57
	1 6 10 15 19 22 25 27 32 36 39 41 44 47 49 52 55
	1 6 9 15 19 22 25 27 32 36 38 41 47 50 52 55 57
118 bus system	2 5 9 12 15 17 20 23 26 28 34 37 41 45 49 53 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 114
	2 5 9 11 12 17 20 23 26 28 34 37 41 45 49 53 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 114
	2 5 9 12 15 17 20 23 26 28 34 37 41 45 49 53 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 115
	2 5 10 11 12 17 20 23 26 28 34 37 41 45 49 53 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 114
	2 6 9 11 12 17 20 23 26 28 34 37 41 45 49 53 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 114
	1 5 9 11 12 17 20 23 26 28 34 37 41 45 49 53 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 114
	2 5 9 11 12 17 20 23 28 30 34 37 41 45 49 53 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 114
	1 5 10 11 12 17 21 24 26 28 34 37 40 45 49 52 56
	62 64 73 75 77 80 85 87 91 94 101 105 110 114 116
	2 5 10 11 12 17 21 24 26 28 34 37 40 45 49 52 56
	62 64 73 75 77 80 85 87 91 94 101 105 110 114 116

Table 8. The PMU Placement Results required with maximum observability

IEEE Network	CO with Maximum Indication PMU location (Bus #)
14 bus system	2 6 7 9
30 bus system	2 4 6 9 10 12 15 18 25 27
	2 4 6 9 10 12 15 20 25 27
	2 4 6 9 10 12 15 19 25 27
57 bus system	1 4 6 9 15 20 24 25 28 32 36 38 41 47 51 53 57
	1 4 6 9 15 20 24 28 30 32 36 38 41 47 50 53 57
	1 4 6 9 15 20 24 28 31 32 36 38 41 47 50 53 57
	1 4 6 9 15 20 24 25 28 32 36 38 41 47 50 53 57
	1 4 6 9 15 20 24 25 28 32 36 38 41 46 50 53 57
	1 4 6 9 15 20 24 25 28 32 36 38 41 46 51 53 57
	1 4 6 9 15 20 24 25 28 32 36 38 39 41 46 51 53
	1 4 6 9 15 20 24 28 30 32 36 38 39 41 46 51 53
	1 4 6 9 15 20 24 28 30 32 36 38 39 41 47 51 53
	1 4 6 9 15 20 24 28 31 32 36 38 39 41 46 51 53
118 bus system	3 5 9 12 15 17 20 23 28 30 34 37 40 45 49 52 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 114
	3 5 9 12 15 17 21 23 28 30 34 37 40 45 49 52 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 114
	3 5 9 12 15 17 21 25 29 34 37 40 45 49 52 56 62
	64 68 70 71 75 77 80 85 86 90 94 102 105 110 114
	3 5 9 12 15 17 20 23 29 30 34 37 40 45 49 53 56
	62 64 68 71 75 77 80 85 86 91 94 102 105 110 115
	3 5 9 12 15 17 21 23 28 30 34 37 40 45 49 53 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 114
	3 5 9 12 15 17 21 25 28 34 37 40 45 49 53 56 62
	64 68 70 71 75 77 80 85 86 90 94 102 105 110 114
	3 5 9 12 15 17 20 23 29 30 34 37 40 45 49 53 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 115
	3 5 9 12 15 17 21 25 28 34 37 40 45 49 52 56 62
	64 68 70 71 75 77 80 85 86 90 94 102 105 110 114
	3 5 9 12 15 17 20 23 28 30 34 37 40 45 49 53 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 115
	3 5 9 12 15 17 20 23 29 30 34 37 40 45 49 52 56
	62 64 68 71 75 77 80 85 86 90 94 102 105 110 115

Each solution is an adequate optimum point to satisfy the real-time power monitoring system. As we can observe, the solver solves exactly the classical $0-1$ integer program within a zero-gap tolerance. A solution pool of optima points is derived inside a zero-gap tolerance meaning that all are non-unique constraint global optima points. The best bound is considered to be the optimal objective value for the minimization problem [64]-[65]. The optimal solutions are derived for both case studies, that is, full observability with one and two criterion and incomplete observability [39], [54]-[55].

Finally, we manage to gather the best possible solution with a best objective, a best bound within a zero-gap tolerance [45]-[52]. All optima points presented in Table 8 have the maximum SORI satisfying the maximum reliability of state of the power network [17]. All optima points are derived at a single run [64]-[65]. Table 9 illustrates the optimization on the 57 bus system [56]. The online diagram and data lines information needed to build the matrices can be found in [56], [64]-[65]. Table 9 illustrates the iterative process produced by Gurobi ILP optimizer function in [64]-[66].

Gurobi attacks the constraint binary program with absolute precision finding an optimum point inside a zero-gap tolerance [66].The log files produced by the Gurobi are shown in the Table 9.

Gurobi completes the optimization process when the MIP Gap is equal to a value within tolerance criteria as declared by default settings. This solution may not be the optimal solution. In this worst case, the incumbent solution is considered to be ε-suboptimal. In this work, there is no issue about this task. The gap is zero and the Gurobi returns a globally optimal solution as the above log file shows. We illustrated the process optimization based on the 57 bus system [57].

Table 9: Process Optimization based on IEEE-57 bus system

Optimize a model with 80 rows, 57 columns and 480 nonzeros
Variable types: 0 continuous, 57 integer (57 binary)
Coefficient statistics:
Matrix range [1e+00, 2e+00]
Objective range [1e+00, 1e+00]
Bounds range [1e+00, 1e+00]
RHS range [1e+00, 1e+00]
Found heuristic solution: objective 14.0000000
Presolve removed 20 rows and 0 columns
Presolve time: 0.00s
Presolved: 60 rows, 57 columns, 348 nonzeros
Variable types: 0 continuous, 57 integer (57 binary)
Root relaxation presolved: 60 rows, 57 columns, 348 nonzeros
Root relaxation: objective 1.100000e+01, 51 iterations, 0.00 seconds (0.00 work units)
Nodes | Current Node | Objective Bounds | Work
Expl Unexpl | Obj Depth IntInf | Incumbent BestBd Gap | It/Node Time
0 0 11.00000 0 8 14.00000 11.00000 21.4% - 0s
H 0 0 13.00000 11.00000 15.4% - 0s
H 0 0 11.00000 11.00000 0.00% - 0s
0 0 - 0 11.00000 11.00000 0.00% - 0s
Optimal solution found at node 0 - now completing solution pool...
Nodes | Current Node | Pool Obj. Bounds | Work
| | Worst |
Expl Unexpl | Obj Depth IntInf | Incumbent BestBd Gap | It/Node Time
0 0 - 0 15.00000 11.00000 26.7% - 0s
0 0 - 0 15.00000 11.00000 26.7% - 0s
0 2 - 0 15.00000 11.00000 26.7% - 0s
Explored 43 nodes (73 simplex iterations) in 0.13 seconds (0.00 work units)
Thread count was 12 (of 12 available processors)
Solution count 10: 11 11 11 ... 11
Optimal solution found (tolerance 1.00e-04)
Best objective 1.100000000000e+01, best bound 1.100000000000e+01, gap 0.0000%
Elapsed time is 0.170926 seconds.
Optimal objective: 1.100000e+01

Table 10: Optimal Results based on Gurobi optimizer

Columns 1 through 15
0 0 0 1 0 0 0 0 0 1 0 0 0 0 1
Columns 16 through 30
0 0 0 0 0 1 0 0 1 0 0 0 0 1 0
Columns 31 through 45
0 1 0 0 0 0 1 0 0 0 1 0 0 0 0
Columns 46 through 57
0 0 1 0 0 0 0 1 0 0 0 0
Elapsed time is 0.173968 seconds.
ans =
4 10 15 21 24 29 32 37 41 48 53
ans =
11

The optimal set solution is displayed in Fig.7 whereas the PMU sites are displayed in Table 11.

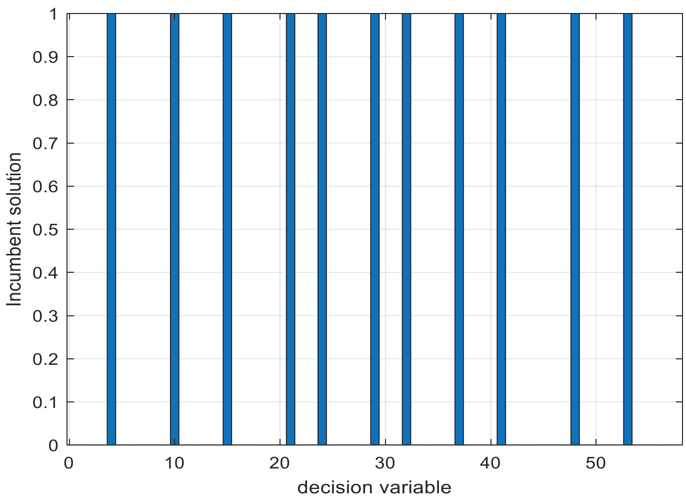

Fig.7 Plot diagram with the IEEE-57 bus network

Table 11. Optimal Placement Outcome for IEEE-57-bus network

PMU locations with partial observability with depth-of-one degree
4, 10, 15, 21, 24, 29, 32, 37, 41, 48, 53

Gurobi results in an optimal solution with a Gap equal to zero [64]-[66]. As we can observe, a set solution is produced given in Table 12. As we see a solution pool is derived for the 57- and 118 bus systems [56]-[57]. A set solution is derived by a number of PMU placements considering the DoOU topological scenario. All solutions are appeared to be adequate for the DoOU scenario [17].

Gurobi gives a set solution $\bar{x} = \{4{,}6\}$ for the 14-bus system whereas it gives a set solution $\bar{x} = \{2{,}10{,}15{,}27\}$ for the 30-bus system. This is an adequate optimum point for that power system [64]-[65]. For the 57- and 118- bus systems, a solution pool of optima points is derived. For each case study, a set of 10 placements are given covering the topological scenario of DoOU [17], [39].

The optimal results reveal that the straight solution of DoOU, on that condition, does not gives the CO but the partial observability with level one [34], [39]. Each optimum point ensures a minimum linear measure between an unobservable power network and the adjacent observable buses including in a local area [17], [39].

The location of PMUs is selected based on a DoOU topological scenario [15], [39]; to such an extent the power network is unobservable with a level of one. The distance between an unobservable power network node with a local area consisting of observable bus is minimized [15], [39].

Table 12: The PMU Placement Results required for the incomplete observability

IEEE Power Network	PMU location (Bus #)
14 bus system	4, 6
30 bus system	2, 10, 15, 27
57 bus system	4, 10, 15, 21, 24, 29, 32, 37, 41, 48, 53
	6, 10, 15, 20, 24, 29, 32, 37, 41, 48, 53
	4, 10, 15, 20, 24, 29, 32, 37, 41, 48, 53
	5,10, 15, 20, 24, 29, 32, 37, 41, 48, 53
	4, 9, 10, 15, 21, 26, 31,36, 48, 52,56
	6, 10, 15, 20, 24, 28, 32, 37, 41, 48, 53
	5, 10, 15, 20, 24, 29, 32, 37, 41, 48, 54
	5, 10, 15, 20, 24, 29, 32, 37, 41, 48, 55
	6, 10, 15, 20, 24, 27, 32, 37, 41, 48, 53
	4, 10, 15, 20, 24, 29, 32, 37, 41, 48, 54
118 bus system	8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 71, 77, 80, 85, 92,105,111, 116
	8, 12, 17, 21, 27, 34, 37, 49, 54, 61, 70, 77, 80, 85, 92,105, 111, 116
	8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 65, 71, 77, 82, 85, 92, 103, 105
	8,12, 17, 22, 27, 34, 39, 49, 54, 61, 68, 70, 77, 80 , 85, 92, 105, 110
	8, 12, 17, 21, 27, 34, 39, 49, 54, 61, 68, 70, 77, 80 ,85, 92, 105, 110
	8, 12, 17, 21, 27,37, 45, 49, 54, 61, 70, 77, 80, 81, 85, 92, 105, 110
	8, 12, 17, 22, 27, 34, 37, 49, 54, 61,70, 77, 80, 81, 85, 92, 105, 110
	8, 12,17, 22, 27, 34, 40,49, 54, 61, 68, 70, 77, 80, 85, 92, 105, 110
	8, 12, 17, 21, 27, 37, 43, 49, 54, 61, 70, 77, 80, 81, 85, 92, 105, 110
	8, 12, 17, 22, 27, 34, 37, 49, 54, 60, 65, 70,77, 80, 85, 92, 105, 110

We are able to solve the Nonlinear Program (NLP) Optimization related to partial observability with DoOU using the open-source optimizer function Solve Constraint Integer Programming (SCIP) [48]-[50]. The specific function solves the NLP model towards a globally optimal solution being derived within a zero-gap tolerance [49]-[50]. SCIP solves the Nonlinear Program (NLP) Optimization towards getting an optimal solution exploring a number of BBNodes as displayed in Tables 13 and 15.

As the output illustrates, a trivial time is spent by the Spatial Branch-and-Bound using IPOPT and Soplex algorithmic schemes. The desired outcome is that the optimizer function gets the globally optimal for both case studies.

To solve the nonlinear problem with SCIP optimization function, a linearization methodology is first coming into being to give the primal bound to calculate the global minimum point [48]. The relaxed problems are solved by the linear solver (LP) embedded in the SCIP optimizer function [52].

As we observe in Tables 13 & 15, an optimal solution is derived within a zero-gap tolerance justifying in this way the achievement of getting the global solution [45]-[50].

Hence, a globally optimal solution is given even so; the programming model is characterized by non-convexity in the constraint function. The non-convexity is suitable to linearize during the iterative process [45]. The binary enumeration tree counts a reasonably explored node to get the optimality within a zero-gap tolerance. The Primal and Dual bounds are evaluated to be equal giving the optimality at the final root node [45]-[50].

As we can observed Primal and Dual bounds are produced at each explored node and finally the two bounds are found to be equal giving a global optimality certificate [47], [49]-[50]. This product is a crucial outcome because it justifies that a globally optimal solution is finally attained [45], [48]-[49]. In such an iterative process, the benchmark output is the primal bound as observed in the above table.

The calculation of dual bound is following and we get the difference of them to be found equal [45]. For minimization models the primal bound is considered to be the desired outcome while the lower bound is culprit for the minimization of the gap. Hence, a global optimum point is reached within an absolute gap zero whereas the iterative iteration is satisfied within a relative gap tolerance.

Hence, the Primal Bound is the leader certificate of optimality constantly compared to the Dual bound. The zero difference of those bounds is an adequate certificate that gives the global solution.

Also, Table 13 displayed the iterative process produced by solving the nonlinear program with SCIP optimizer for the 14 bus system whose data lines' information can be found in MATPOWER [56]. The optimal solution is a restricted PMU numbers at selected power system nodes that is, $\vec{\boldsymbol{x}} = \{\boldsymbol{4}, \boldsymbol{6}\}$ for the 14 bus system [17], [34]-[38]. The incomplete observability condition and gives a feedback to the state estimation routine as noted in [36].

The set solution is displayed in the following plot-diagram. As we can see, the SCIP optimizer minimizes the algorithmic scheme, gives the optimal solution without suffering from the existence of floating points. The optimal solution strictly satisfies the binary restriction $\vec{\boldsymbol{x}} \in \{0.1\}$ [45]-[50].

The difference between the primal and dual bounds are evaluated and found equal to zero [50]. Hence, the global solution is delivered by the SCIP optimizer function meaning that the optimizer routine handles the nonlinear program and its non-convexity with adequate efficiency towards optimality meaning that this solution is a global one being succeeded within a zero-gap [45]-[50].

The placement set solution is displayed in Table 13 where the PMU sites are shown in the plot diagram shown in Fig.8. Also, Table 15 illustrates the iterative process produced by solving the nonlinear program with SCIP optimizer for the 30 bus system whose data lines' information can be also found in MATPOWER [56]. The set solution is$\vec{x} = \{5, 10, 15, 27\}$ as found in [39]. BBA is using a primal or dual simplex during the iterative process problem solving [49]-[50].

A global optimal solution is achieved as the log file is shown in Table 13. BBA utilizes a tree search strategy to unquestioningly calculate all solutions that can exist to the given minimization problem, using pruning rules to get rid of regions of the search space that cannot give a high degree solution point. SCIP utilizes three options to construct the binary tree such as search and branching strategies and rules to prune infeasible regions to find a feasible and optimal solution point. We find out that if the binary tree is small in size, the solving process is fast. The solving time is 0.42 s and SCIP returns an optimal solution without the entire optimization process being computational heavy.

Journal of Physics: Conference Series **2701** (2024) 012013 doi:10.1088/1742-6596/2701/1/012013

That enumeration tree has been implemented without computation complexity as the algorithmic scheme's experimental outcome shows. That optimal result consists of a desired outcome in the direction of getting a feasible and global solution in a robust way without computation burden.

As we observed, the optimal solution for each power network is a pure binary point without suffering from the floating points. The final position is an outcome produced by an appropriate algorithm to the optimization problem either in combinatorial format a continuous domain [45]-[52].

We assign positive feedback to the minimization problem solving and successfully terminate the iterative process with a global minimum point**.** SCIP optimizer solver executes a powerful branching regulation which results in a small-sized B&B tree without producing many branching decisions.

The routine calculates the upper and lower bounds in the b&b tree's development where it terminates with the best incumbent solution [49]-[50], [62]-[64].

SCIP successfully closes the optimality gap. Gap zero means that no better possible solution can be found [58]-[60]. Hence, SCIP gives a global optimal solution and the minimization model has been solved exactly avoiding being trapped into a local solution or a sub-optimal solution [62]-[64].

That enumeration tree is solved without computation complexity as the algorithmic scheme's experimental outcome illustrates [45]. Table 13 displays the results where a simulation run is successful because the Primal and Dual Bounds are found to be equal [58]-[60].

The Primal and Dual Bounds are equal; hence, the optimization problem has been solved as shown in the status log file. The Primal Bound is considered the optimal solution whereas the Dual Bounds closed the gap. Hence, an optimal solution is achieved within a zero-gap tolerance; a global solution was found within a 0.00 % optimality criterion [58]-[60].

Table 13: Process Optimization by SCIP optimizer function

```
Nonlinear Program (NLP) Optimization
 min  f(x)
 s.t. lb <= x <= ub
      cl <= c(x) <= cu
------------------------------------------------------
   Problem Properties:
# Decision Variables:        14
# Constraints:               37
  # Bounds:                  28
  # Nonlinear Equality:       9
------------------------------------------------------
  Solver Parameters:
Solver:                      SCIP
Objective Gradient:      @(x)mklJac(fun,x)
Constraint Jacobian:     @(x)mklJac(nlcon,x)
Jacobian Structure:        Supplied
------------------------------------------------------
presolving:
(round 1) 0 del vars, 0 del conss, 0 add conss, 0 chg bounds, 0 chg sides, 0 chg coeffs, 1 upgd conss, 0 impls, 0 clqs
(round 2) 1 del vars, 1 del conss, 0 add conss, 2 chg bounds, 0 chg sides, 0 chg coeffs, 1 upgd conss, 0 impls, 0 clqs
(round 3) 1 del vars, 1 del conss, 22 add conss, 2 chg bounds, 0 chg sides, 0 chg coeffs, 1 upgd conss, 0 impls, 0 clqs
(round 4) 1 del vars, 1 del conss, 22 add conss, 2 chg bounds, 0 chg sides, 0 chg coeffs, 32 upgd conss, 0 impls, 0 clqs
(round 5) 1 del vars, 1 del conss, 22 add conss, 2 chg bounds, 0 chg sides, 0 chg coeffs, 41 upgd conss, 0 impls, 0 clqs
presolving (6 rounds):
 1 deleted vars, 1 deleted constraints, 22 added constraints, 2 tightened bounds, 0 added holes, 0 changed sides, 0 changed coefficients
 0 implications, 0 cliques
presolved problem has 36 variables (0 bin, 0 int, 0 impl, 36 cont) and 31 constraints
     9 constraints of type <bounddisjunction>
    22 constraints of type <quadratic>
Presolving Time: 0.03
```

time	node	left	LP iter	LP it/n	mem	mdpt	frac	vars	cons	cols	rows	cuts	confs	strbr	dualbound	primalbound	gap
0.1s	1	0	2	-	284k	0	0	36	31	36	50	0	0	0	0.000000e+000	--	Inf
U 0.1s	1	0	2	-	293k	0	0	36	31	36	50	0	0	0	0.000000e+000	4.000000e+000	Inf

```
******************************************************************************
This program contains Ipopt, a library for large-scale nonlinear optimization.
 Ipopt is released as open-source code under the Eclipse Public License (EPL).
        For more information visit http://projects.coin-or.org/Ipopt
******************************************************************************

  0.2s|   1 |     0 |    14 |     - | 296k|   0 |   0 |  36 |  32 |  36 |  58 |   8 |   0 |   0 |0.000000e+000 |4.000000e+000 |    Inf
  0.2s|   1 |     0 |    19 |     - | 299k|   0 |   0 |  36 |  32 |  36 |  63 |  13 |   0 |   0 |0.000000e+000 |4.000000e+000 |    Inf
  0.2s|   1 |     0 |    22 |     - | 301k|   0 |   0 |  36 |  32 |  36 |  66 |  16 |   0 |   0 |0.000000e+000 |4.000000e+000 |    Inf
  0.3s|   1 |     2 |    45 |     - | 304k|   0 |   0 |  36 |  32 |  36 |  66 |  16 |   0 |   0 |0.000000e+000 |4.000000e+000 |    Inf
* 0.4s|  11 |     7 |    64 |   4.2 | 290k|   5 |   - |  36 |  32 |  36 |  25 |  21 |   0 |   0 |2.000000e+000 |3.000000e+000 |  50.00
* 0.4s|  20 |     0 |    72 |   2.6 | 286k|   5 |   - |  36 |  32 |  36 |  25 |  21 |   0 |   0 |2.000000e+000 |2.000000e+000 |   0.00

SCIP Status        : problem is solved [optimal solution found]
Solving Time (sec) : 0.42
Solving Nodes      : 20
Primal Bound       : +2.00000000000000e+000 (3 solutions)
Dual Bound         : +2.00000000000000e+000
Gap                : 0.00

ans =

   0   0   0   1   0   1   0   0   0   0   0   0   0   0

ans =

   4   6
```

$\text{percentage relative gap} = 100 * |\text{primal} - \text{dual}|/\text{MIN}(|\text{dual}|, |\text{primal}|)$; Termination criteria: $\text{Gap} = 0$ and Solving Time=0.42 s;

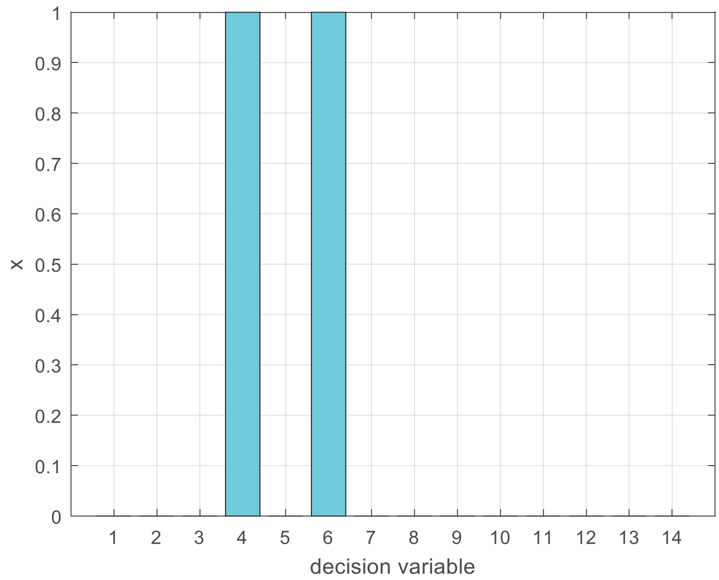

Figure 8. Plot diagram of Placement Result for IEEE 14-bus network

Table 14. Optimal Placement Outcome for IEEE-14-bus network

PMU locations with partial observability with depth-of-one degree
4, 6

The utilization of the SCIP has enabled the resolution of NLP optimization issues associated with partial observability, specifically those involving a DoOU topological scenario following the study published in [17]. The output can be considered a desired outcome [63].

Therefore, a constraint integer linear programming algorithmic shceme is utilized to obtain a strong, non-unique global solution point within a zero-gap [63].

As we observed, the primal bound takes the lead in the optimization process, with an equal dual bound estimated to achieve zero-gap tolerance [45]. The dual bound close the gap so the solution is achieved within 0.00 % optimality [69]. Therefore, the solution is characterized as a global one [70].

The relative gap is a benchmark termination criterion which shows if the B&B terminates to a global solution or to a suboptimal solution [58]-[60]. This study gives an inventive model of unambiguous global optimal solutions either in a combinatorial domain or continuous nonlinear programming. Both algorithmic schemes deliver solutions within a 0.00 % criterion [49].

The optimal objective function is the best possible with the one derived in Gou's work [17] and some placement results related to that objective function have been calculated with accuracy due to zero-gap as claimed in previous studies [17], [34], [39]. An optimal solution is given within a zero-gap tolerance as the log output shows in Table 15 [63].

Table 15: Process Optimization by SCIP optimizer function

```
Nonlinear Program (NLP) Optimization
 min  f(x)
 s.t. lb <= x <= ub
      cl <= c(x) <= cu
------------------------------------------------------
   Problem Properties:
# Decision Variables:         30
# Constraints:                79
  # Bounds:                   60
  # Nonlinear Equality:       19
------------------------------------------------------
  Solver Parameters:
Solver:                  SCIP
Objective Gradient:      @(x)mklJac(fun,x)
Constraint Jacobian:     @(x)mklJac(nlcon,x)
Jacobian Structure:      Supplied
------------------------------------------------------
presolving:
(round 1) 0 del vars, 0 del conss, 0 add conss, 0 chg bounds, 0 chg sides, 0 chg coeffs, 1 upgd conss, 0 impls, 0 clqs
(round 2) 1 del vars, 1 del conss, 0 add conss, 2 chg bounds, 0 chg sides, 0 chg coeffs, 1 upgd conss, 0 impls, 0 clqs
(round 3) 1 del vars, 1 del conss, 76 add conss, 2 chg bounds, 0 chg sides, 0 chg coeffs, 1 upgd conss, 0 impls, 0 clqs
(round 4) 1 del vars, 1 del conss, 76 add conss, 2 chg bounds, 0 chg sides, 0 chg coeffs, 96 upgd conss, 0 impls, 0 clqs
(round 5) 1 del vars, 1 del conss, 76 add conss, 2 chg bounds, 0 chg sides, 0 chg coeffs, 115 upgd conss, 0 impls, 0 clqs
presolving (6 rounds):
 1 deleted vars, 1 deleted constraints, 76 added constraints, 2 tightened bounds, 0 added holes, 0 changed sides, 0 changed coefficients
 0 implications, 0 cliques
presolved problem has 106 variables (0 bin, 0 int, 0 impl, 106 cont) and 95 constraints
     19 constraints of type <bounddisjunction>
     76 constraints of type <quadratic>
Presolving Time: 0.03
```

time	node	left	LP iter	LP it/n	mem	mdpt	frac	vars	cons	cols	rows	cuts	confs	strbr	dualbound	primalbound	gap
0.1s	1	0	12	-	553k	0	0	106	95	106	167	0	0	0	0.000000e+000	--	Inf
0.2s	1	0	30	-	585k	0	0	106	96	106	184	17	0	0	0.000000e+000	--	Inf
0.2s	1	0	42	-	592k	0	0	106	96	106	196	29	0	0	0.000000e+000	--	Inf
0.2s	1	0	52	-	598k	0	0	106	96	106	206	39	0	0	0.000000e+000	--	Inf
0.2s	1	0	60	-	602k	0	0	106	96	106	214	47	0	0	0.000000e+000	--	Inf
0.2s	1	0	65	-	605k	0	0	106	96	106	219	52	0	0	0.000000e+000	--	Inf

```
  0.2s|    1 |    0 |  67 |  -  | 606k|  0  |  0 | 106 |  96 | 106 | 221 |  54 |  0 |  0 |0.000000e+000 |      --       |  Inf
  0.3s|    1 |    0 | 170 |  -  | 616k|  0  |  0 | 106 |  96 | 106 | 223 |  56 |  0 |  0 |0.000000e+000 |      --       |  Inf
  0.3s|    1 |    0 | 172 |  -  | 617k|  0  |  0 | 106 |  96 | 106 | 225 |  58 |  0 |  0 |0.000000e+000 |      --       |  Inf
  0.3s|    1 |    0 | 174 |  -  | 618k|  0  |  0 | 106 |  96 | 106 | 227 |  60 |  0 |  0 |0.000000e+000 |      --       |  Inf
  0.3s|    1 |    0 | 175 |  -  | 619k|  0  |  0 | 106 |  96 | 106 | 228 |  61 |  0 |  0 |0.000000e+000 |      --       |  Inf
  0.3s|    1 |    2 | 175 |  -  | 619k|  0  |  0 | 106 |  96 | 106 | 228 |  61 |  0 |  0 |0.000000e+000 |      --       |  Inf
* 0.5s|   79 |   61 | 374 | 3.8 | 598k| 11  |  - | 106 |  96 | 106 |  86 | 121 |  0 |  0 |4.000000e+000 |5.000000e+000 | 25.00
  0.5s|  100 |   50 | 389 | 3.2 | 595k| 11  |  - | 106 |  96 |   0 |   0 | 121 |  0 |  0 |4.000000e+000 |5.000000e+000 | 25.00
* 0.5s|  193 |    0 | 533 | 2.4 | 557k| 11  |  - | 106 |  94 | 106 |  79 | 146 |  0 |  0 |4.000000e+000 |4.000000e+000 |  0.00

SCIP Status        : problem is solved [optimal solution found]
Solving Time (sec) : 0.50

Solving Nodes      : 193
Primal Bound       : +4.00000000000000e+000 (2 solutions)
Dual Bound         : +4.00000000000000e+000
Gap                : 0.00

ans =

  Columns 1 through 17

    0    0    0    0    1    0    0    0    0    1    0    0    0    0    1    0    0

  Columns 18 through 30

    0    0    0    0    0    0    0    0    0    1    0    0    0

info =

      BBNodes: 193
        BBGap: 0
         Time: 0.5885
    Algorithm: 'SCIP: Spatial Branch and Bound using IPOPT and SoPlex'
       Status: 'Globally Optimal'

ans =

     5    10    15    27
```

percentage relative gap $= 100 * |\text{primal} - \text{dual}| / \text{MIN}(|\text{dual}|, |\text{primal}|)$; Termination criteria: $\text{Gap} = 0$ and Solving Time=0.50 s;

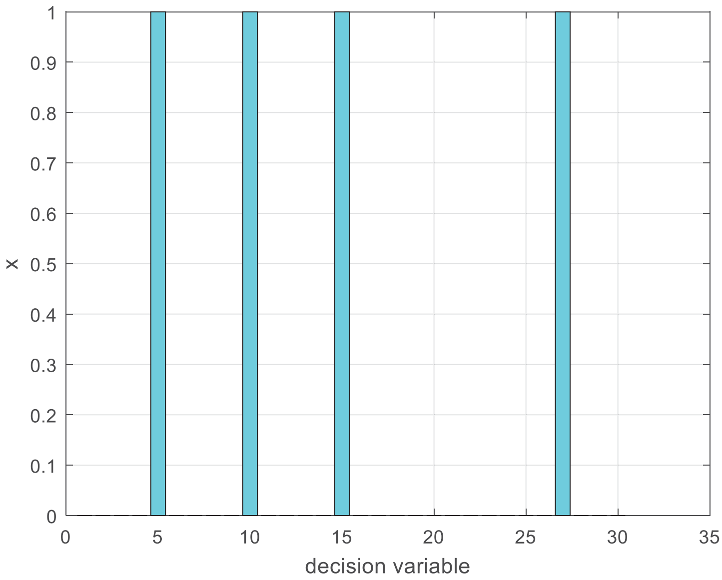

Figure 9. Plot diagram of Placement Result for IEEE 30-bus network

Table 16. Optimal Placement Outcome for IEEE-30-bus network

PMU locations with partial observability with depth-of-one degree
5, 10, 15, 27

SCIP optimizer routine executes a powerful branching regulation which leads to a small-sized B&B tree. The solver calculates the upper and lower bounds in the b&b tree's development where it results in the best incumbent solution [57]-[59].

Thus, an optimal solution is attained within a zero-gap tolerance; a global solution was figured out within a 0.00 % optimality criteria. SCIP successfully closes the optimality gap. Gap zero means that no better possible solution can be found. Therefore, SCIP returns a global optimal solution and the problem have been solved exactly avoiding being trapped into a local solution or a sub-optimal solution [57]-[59].

We assign a positive feedback to the optimization problem solving and successfully terminate the iterative process with a global minimum point [57]**.**

As we observed, we solved the Nonlinear Program (NLP) Optimization related to partial observability with DoOU using the open-source optimizer function Solve Constraint Integer Programming (SCIP) (Tables 13 & 15). [58]-[60]. The specific function solves the NLP model towards a global optimal solution being derived within a zero-gap tolerance [59]-[60].

To solve the polynomial (nonlinear) problem with SCIP optimization function, a linearization methodology is first coming into being to construct the branch strategy [45]-[50].

Additionally, a Dual bound is considered at each node where the relaxed problems are solved, the regions are explored [49]-[50]. Regions that give objective value larger than the Primal bound are pruned [49]-[50]. At the final stage of the iteration, the two bounds are evaluated to be equal thus a zero-gap is computed. Hence, the global optimal solution is attained within 0.00 % optimality.

At this point we clearly declare that this output is a desired outcome from the point of view of global optimality view [45]-[50]. It is a strong innovation of this paper separating from the local solution being achieved by SQP and IPMs [27]-[30], [45]-[49]. The 14-bus and 30-bus systems are utilized as benchmark experimental tests and some results are derived in Tables 13 & 15 [56]-[57].

The non-convexities declared in the constraint function give us a nonlinear program can be solved by the SCIP optimizer function [58]. A spatial branch-and-bound algorithm is an algorithm that composed of some kind of building an enumeration tree in the direction of getting an optimal solution dependently by the gap at the given root node [58]-[60].

SCIP optimizes the nonlinear program (NLP) optimization based on the 14 – and 30- bus systems [56]-[57]. SCIP finds the optimality using linear relaxations solved at each explored node (Tables 13 & 15) [49]-[52]. These experimental tests are working as benchmark systems to show that SCIP optimizer leads to a global optimal solution; as optimization function claims in its output [58]-[60].

To solve any integer program, a deeper relationship between integer programming with linear programming is built [58]-[60]. The binary integrality $\vec{x} \in \{0,1\}$ is relaxed to a continuous domain that is,$\vec{x} \in [0,1]$ [45]-[50]. A number of nodes are explored where a linear problem is solved as follows: $min\{w^T\vec{x}: Ax \leq b, \vec{x} \in R_+^n\}$ through a linear programming (LP) solver [45]. Finally, the optimizer function will give a feasible and binary solution in the following format $\vec{x} \in \{0,1\}$ [45]-[50].

The relaxation's tightness relies on the variable bounds; hence the tightness methodology is very important for a well-performing SCIP optimizer in getting the optimality within zero-gap toleration [49]-[52]. SCIP optimizer function is applied to continuous and discrete global optimization problems [58].

The optimizer function delivers the optima point within a zero-gap tolerance for each power network. Hence, we use a convergent ILP algorithm in getting a strong non-unique global solution point with a zero-gap being achieved [49]-[50]. Our validation output is that the gap claims a globally solution has been found [49]-[50]

The solutions don't appear to be floating point's phenomenon [45]-[50]. Also, it is characterized as a globally optimum point since the percentage relative gap goes to zero. The best cost function value is equal to the upper bound of the B&B tree while the lower bound minimizes the absolute gap. Hence, the global certificate of optimality is verified.

Optima points are achieved within a zero-gap difference between the Primal and Dual Bounds. SCIP estimates that difference equal to zero; a global solution is attained using reasonable explored nodes in the construction of the enumeration tree [45]. The nonlinear problem is approached by a polyhedron and it is solved through the SCIP optimizer function which constructs the enumeration tree [58].

The entire process optimization is reached when a globally optimal solution is found. The dual bound is considered to be the desired outcome within a zero-gap tolerance at the explored and given root node [58]-[63]. The optimizer SCIP function can be found in OPTI-toolbox [63].

SCIP is able to detect a best integral solution at the initial stage that is the primal bound relying on decision values declared by the program's user for all variables [58]-[60]. Also, a dual bound is constantly evaluated so as the difference of those bounds to lead to zero justifying the approach and finding of a globally optimal solution [58]-[60].

As observed, SCIP optimizer function linearized each polynomial equality constraint, solved the process optimization through a construction of a branch-and-bound tree and finally delivered a globally optimal solution [48]. At each node where the relaxed problems are solved, the regions are explored and infeasible regions are pruned [45]-[50].

At the final stage of the iteration, the two bounds are evaluated to be equal thus a zero-gap is computed [45]-[50]. As we can see the primal bound is the leader in this optimization process, an equal dual bound has been evaluated that leads to zero-gap toleration [49]-[50].

The standard IEEE systems are used and some results are derived displayed in Tables 13 &15 [46]-[47]. These experimental tests are working as benchmark systems to show that spatial branch-and-bound embedded in the open-source SCIP optimizer leads to a globally optimal solution; as the output of the solver claims [58]-[60], [63]. Hence, we use a convergent ILP algorithm in getting a strong non unique global solution point with a zero-gap being achieved [61]-[67].

Our validation output is that the gap claims a global solution has been found. The optimizer SCIP function can be found in OPTI-toolbox [52]. SCIP optimizer function is able to detect a best integral solution at the initial stage that is a primal bound relying on decision values declared by the

Program's user for all variables and then a dual bound is reached [48]-[50]. The optimal best objective function is the primal bound and some placement results are derived within 0.00% optimality [45]-[50].

Then, we execute the binary polynomial problem with YALMIP global nonlinear branch-and-bound algorithmic (BBA) scheme to get the optimality [70]. YALMIP BBA utilizes external routines to count the upper and lower bounds whereas the binary tree is constructed [68]. To test this optimization model, the 14-bus, 30-and 57-bus systems are used to find out the optimality [56]-[57].

This global optimizer function is performed by calling external integer linear as well nonlinear optimization functions to build the enumeration tree for the purpose of getting an optimum point within a zero-gap tolerance [68], [69]-[70].

During the iterative process, the upper and lower bounds are evaluated by an Upper solver and Lower solver respectively [70]. Meanwhile an LP solver takes care of solving the relaxed problems to get an optimum point [69]. The difference of those bounds gives the certificate of global optimality [70]. For minimization problems as the proposed problem studied in this work, the upper bound is considered to be the best possible solution whereas the lower bound is culprit to close the gap [50]. Hence, optimality is achieved within a 0.00% gap tolerance [45].

As an upper solver is used the FMINCON to get the first feasible integral solution and based on this optimum, the lower solver is evaluated to get the lower bound [47]. We use SCIP optimizer function [58] as well as Gurobi solver [64] to solve the relaxed problems during the branch strategy and to estimate the lower bound [45]-[50]. An optimum point is found to be within a zero-difference between those bounds. Hence, a global optimum is attained within a satisfied relative gap.

Then, increasing the PMUs number, a complete PMU arrangement can be implemented. The incremental PMU quantity starts with a few numbers of synchronized phasor devices, in phase if it is required until the entire complete observability can be calculated [40]. The iterative process is shown in the Table 17 for the benchmark IEEE-14 case study [56].

For a 30-bus and 57-bus system, we can find an optimal solution under the concept of incomplete observability [17], [39]. Table 19 illustrates the iterative process derived by YALMIP BBA [70]. YALMIP BBA is proved to be scalable since it can give an optimum point; a globally optimal solution is found within a zero-gap tolerance [69]-[70].

External optimization functions such as FMINCON or Gurobi can be invoked for the purpose of counting upper and lower bounds respectively. Gurobi also performs as an LP solver to build the branching strategy and to implement relaxed problems in getting the optimality [45]-[50], [64].

Table 17: Process Optimization by YALMIP BBA optimizer function

ID	Constraint	Coefficient range
#1	Equality constraint (polynomial) 1x1	1 to 1
#2	Equality constraint (polynomial) 1x1	1 to 1
#3	Equality constraint (polynomial) 1x1	1 to 1
#4	Equality constraint (polynomial) 1x1	1 to 1
#5	Equality constraint (polynomial) 1x1	1 to 1
#6	Equality constraint (polynomial) 1x1	1 to 1
#7	Equality constraint (polynomial) 1x1	1 to 1
#8	Equality constraint (polynomial) 1x1	1 to 1
#9	Equality constraint (polynomial) 1x1	1 to 1

Table 17: Process Optimization by YALMIP BBA optimizer function continued

Journal of Physics: Conference Series **2701** (2024) 012013 doi:10.1088/1742-6596/2701/1/012013

```
* Starting YALMIP global branch & bound.
* Upper solver     : fmincon
* Lower solver     : GUROBI
* LP solver        : GUROBI
* -Extracting bounds from model
* -Perfoming root-node bound propagation
* -Calling upper solver (no solution found)
* -Branch-variables : 14
* -More root-node bound-propagation
* -Performing LP-based bound-propagation
* -And some more root-node bound-propagation
* Starting the b&b process
 Node       Upper       Gap (%)       Lower       Open   Time
    1 :   2.00000E+00     0.00     2.00000E+00    2     1s  Solution found by heuristics
* Finished.  Cost: 2 (lower bound: 2, relative gap 3.3333e-09%)
* Termination with relative gap satisfied
* Timing: 26% spent in upper solver (2 problems solved)
*             6% spent in lower solver (1 problems solved)
*             10% spent in LP-based domain reduction (28 problems solved)
*             1% spent in upper heuristics (1 candidates tried)

sol =

    yalmipversion: '20210331'
    matlabversion: '9.4.0.813654 (R2018a)'
       yalmiptime: 0.2703
       solvertime: 1.1047
             info: 'Successfully solved (BMIBNB)'
          problem: 0

Elapsed time is 1.385076 seconds.
ans =

     0     0     0     1     0     1     0     0     0     0     0     0     0     0

ans =

     4     6
Linear scalar (real , binary, 14 variables)
Current value: 2
Coefficients range: 1 to 1
```

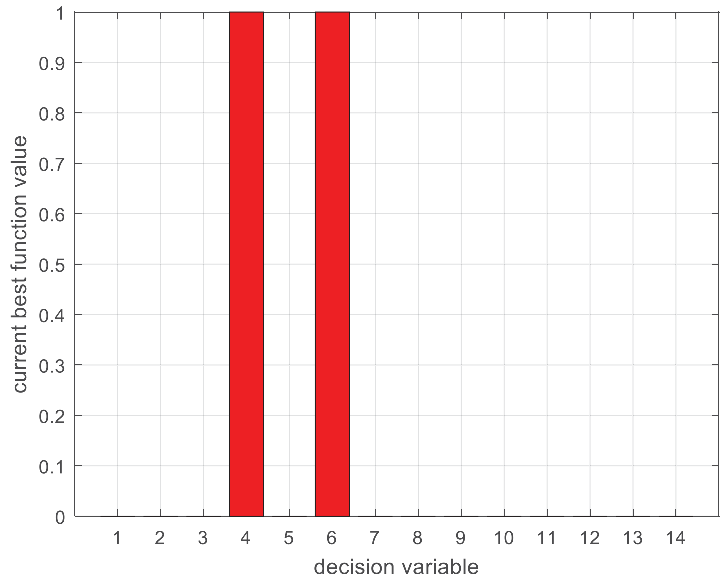

Figure 10. Plot diagram of Placement Result for 14-bus network

Table 18. Optimal Placement Outcome for 14-bus network

PMU locations with partial observability with depth-of-one degree
4, 6

Table 19: Process Optimization by YALMIP BBA optimizer function

* Starting YALMIP global branch & bound.
* Upper solver : fmincon
* Lower solver : GUROBI
* LP solver : GUROBI
* -Extracting bounds from model
* -Performing root-node bound propagation
* -Calling upper solver (no solution found)
* -Branch-variables : 30
* -More root-node bound-propagation
* -Performing LP-based bound-propagation
* -And some more root-node bound-propagation
* Starting the b&b process
Node Upper Gap (%) Lower Open Time
1 : 4.00000E+00 0.00 4.00000E+00 2 438s Solution found by heuristics
* Finished. Cost: 4 (lower bound: 4, relative gap 2e-09%)
* Termination with relative gap satisfied
* Timing: 48% spent in upper solver (2 problems solved)

Table 19: Process Optimization by YALMIP BBA optimizer function continued

```
*             1% spent in lower solver (1 problems solved)
*             1% spent in LP-based domain reduction (60 problems solved)
*             1% spent in upper heuristics (1 candidates tried)
*             1% spent in lower solver (1 problems solved)
sol =
   yalmipversion: '20210331'
   matlabversion: '9.4.0.813654 (R2018a)'
     yalmiptime: 1.6874
     solvertime: 438.9766
           info: 'Successfully solved (BMIBNB)'
        problem: 0

Elapsed time is 240.868845 seconds.

ans =

    4


ans =

  Columns 1 through 17

    1    0    0    0    0    0    0    0    0    1    0    0    0    0    1    0    0

  Columns 18 through 30

    0    0    0    0    0    0    0    0    0    1    0    0    0

ans =

    1   10   15   27

Linear scalar (real , binary, 30 variables)
Current value: 4
Coefficients range: 1 to 1
```

FMINCON is invoked to count the upper bound, the Gurobi solver estimates the lower bound and an LP solver solves the relaxed problems [61]-[64]. Finally, the optimization problem is completed at one root node is explored where an optimal solution is reached within a zero-gap tolerance [49]-[50].

Using the 30 bus system, the upper bound is considered to be the cost value within a zero-gap tolerance and a relative gap equal to $2e - 09\%$ [69]-[70]. The iterative process is terminated within a relative gap satisfied [69]-[70]. The optimal solution is displayed in Fig.11 and Table 20.

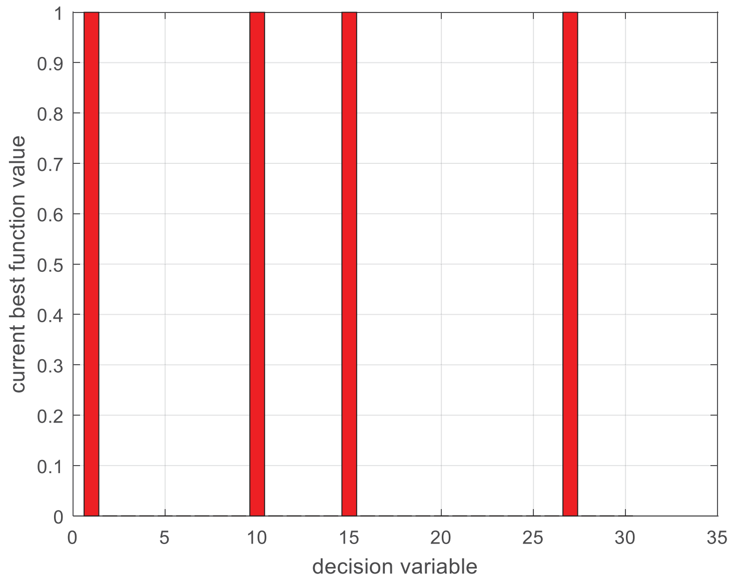

Figure 11. Plot diagram of Placement Result for 30-bus network

Table 20. Optimal Placement Outcome for 30-bus network

PMU locations with partial observability with depth-of-one degree
1, 10, 15, 27

Afterward, we carried out our model on the 30 bus system using SCIP as a lower solver [56], [57]. The iterative process is shown in Table 21. The FMINCON solver can count the upper bound which is the best integral solution; also, a leadership to give optimality [70].

SCIP solver develops a branch strategy, exploring nodes where linear problems are solved and some infeasible regions are pruned. The relaxed problems are solved by the linear programming (LP) solver embedded in the SCIP optimizer function [63]. Also, SCIP estimates the lower bound [70].

YALMIP BBA estimates the difference between those bounds equal to zero and a global solution is given within a zero-gap tolerance [69]-[70]. Hence, the cost value was found equal to the upper bound. The binary tree results in an optimum point within a zero-gap justifying its global nature [58]-[60].

Table 21: Process Optimization by YALMIP BBA optimizer function

++
* Starting YALMIP global branch & bound.
* Upper solver : fmincon
* Lower solver : SCIP
* LP solver : SCIP
* -Extracting bounds from model
* -Perfoming root-node bound propagation
* -Calling upper solver (no solution found)

Table 21: Process Optimization by YALMIP BBA optimizer function continued

```
* -Branch-variables : 30
* -More root-node bound-propagation
* -Performing LP-based bound-propagation
* -And some more root-node bound-propagation
* Starting the b&b process
 Node    Upper     Gap (%)    Lower     Open   Time
   1 :  4.00000E+00   0.00   4.00000E+00   2   207s  Solution found by heuristics
* Finished.  Cost: 4 (lower bound: 4, relative gap 2e-09%)
* Termination with relative gap satisfied
* Timing: 50% spent in upper solver (2 problems solved)
*          2% spent in lower solver (1 problems solved)
*         17% spent in LP-based domain reduction (60 problems solved)
*          1% spent in upper heuristics (1 candidates tried)

sol =

  struct with fields:

    yalmipversion: '20210331'
    matlabversion: '9.4.0.813654 (R2018a)'
       yalmiptime: 0.9389
       solvertime: 207.8221
             info: 'Successfully solved (BMIBNB)'
          problem: 0

Elapsed time is 208.985846 seconds.

ans =

     4

ans =

  Columns 1 through 16

     0     0     0     0     1     0     0     0     0     1     0     0     0     0     1     0

  Columns 17 through 30

     0     0     0     0     0     0     0     0     0     0     1     0     0     0
ans =

     5    10    15    27

Linear scalar (real , binary, 30 variables)
Current value: 4
Coefficients range: 1 to 1
```

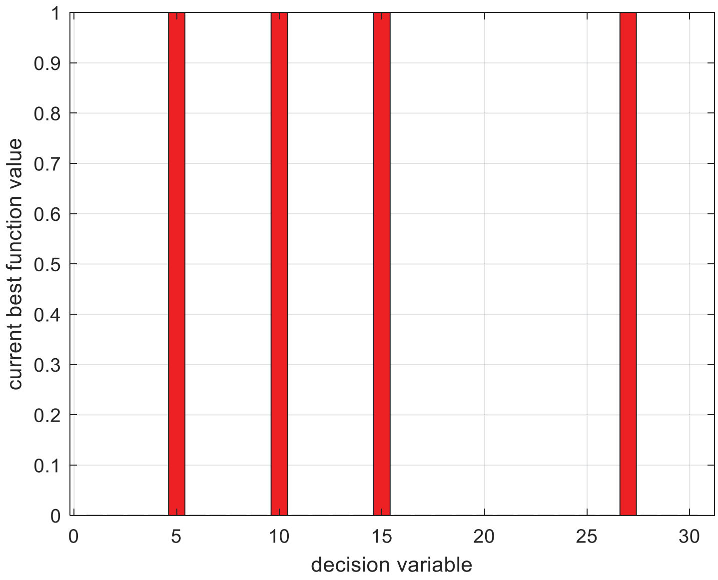

Figure 12. Plot diagram of Placement Result for IEEE 30-bus network

Table 22. Optimal Placement Outcome for IEEE-30-bus network

PMU locations with partial observability with depth-of-one degree
5, 10, 15, 27

The iterative process is shown in the Table 23 for the 57 network [57].

Table 23: Process Optimization by YALMIP BBA optimizer function

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Starting YALMIP global branch & bound.
* Upper solver    : fmincon
* Lower solver    : SCIP
* LP solver       : SCIP
* -Extracting bounds from model
* -Perfoming root-node bound propagation
* -Calling upper solver (no solution found)
* -Branch-variables : 57
* -More root-node bound-propagation
* -Performing LP-based bound-propagation
* -And some more root-node bound-propagation
* Starting the b&b process
 Node      Upper        Gap (%)      Lower       Open     Time
    1 :   1.10000E+01    0.00      1.10000E+01    2   153s  Solution found by heuristics
* Finished.  Cost: 11 (lower bound: 11, relative gap 8.3333e-10%)
* Termination with relative gap satisfied
* Timing: 19% spent in upper solver (2 problems solved)
*              21% spent in lower solver (1 problems solved)
*              23% spent in LP-based domain reduction (114 problems solved)
```

Journal of Physics: Conference Series **2701** (2024) 012013 doi:10.1088/1742-6596/2701/1/012013

Table 23: Process Optimization by YALMIP BBA optimizer function continued

* 1% spent in upper heuristics (1 candidates tried)
sol =
struct with fields:
yalmipversion: '20210331'
matlabversion: '9.4.0.813654 (R2018a)'
yalmiptime: 0.1411
solvertime: 153.0649
info: 'Successfully solved (BMIBNB)'
problem: 0
Elapsed time is 153.208162 seconds.
ans =
Columns 1 through 16
0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0
Columns 17 through 32
0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1
Columns 33 through 48
0 0 0 0 1 0 0 0 1 0 0 0 0 0 0 1
Columns 49 through 57
0 0 0 0 0 0 1 0 0
Linear scalar (real , binary, 57 variables)
Current value: 11
Coefficients range: 1 to 1
ans =
11
ans =
5 10 15 20 24 29 32 37 41 48 55

Using the 57 bus system [56]-[57], the upper bound is considered to be the cost value within a zero-gap tolerance and a relative gap equal to $8.3333e - 10\%$ [69]-[70]. The iterative process is terminated within a relative gap satisfied [69]-[70]. The relative gap is meaningless.

The optimum set solution is tabulated in Table 24 whereas the Fig. 13 shows the placement sites. The PMU configuration is in full agreement with those published in Bei Gou's [17] and Rohit Babu's works [39]. This fact reveals the truth that our algorithmic scheme delivers true solutions.

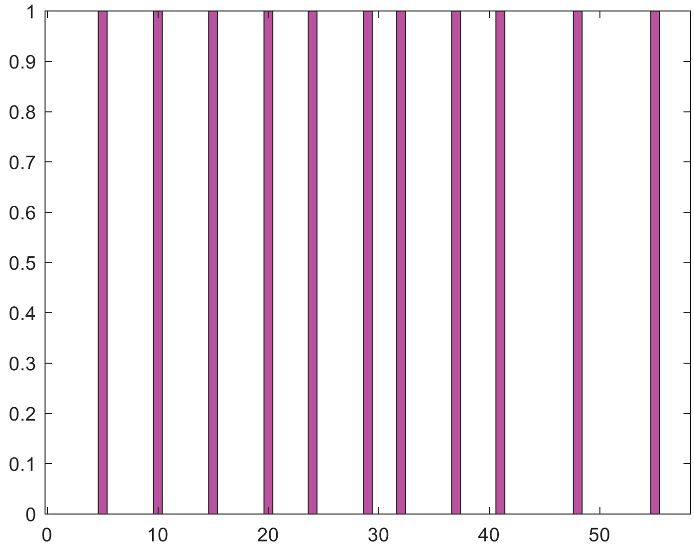

Figure 13. Plot diagram of Placement Result for IEEE 30-bus network

Table 24. Optimal Placement Outcome for IEEE-57-bus network

PMU locations with partial observability with a depth-of-one degree
5, 10, 15, 20, 24, 29, 32, 37, 41, 48, 55

Mathematical and evolutionary algorithms give non-unique optimum points by which the DoOU is clearly satisfied. In that case, the context of all nearby nodes of any observable node must be observed [17], [34]-[40]. It is observed that the methodologies consider a graph theoretical approach reflecting the one-line diagram with a binary connectivity matrix and an incidence matrix [42]-[44].

The discrete mathematical model is solved by BBA and algorithms based on population like genetic algorithms and binary particle swarm optimization are also adopted to find optimality.

BBA solves a binary linear integer program with guaranteed zero-gap optimality [49]-[52]. Using this solution as a benchmark study model, GAs and BPSO result in a globally optimal solution. Both evolutionary algorithms are proved to be convergent to a PMU placement under the concept of partial observability with low level equal to 1 [17], [39].

After the discrete standpoint of optimization problem solving, we solve it using nonlinear algorithms such as SQP and IPMs [27]-[29], [45]-[49]. Both algorithms found locally solutions which are global in unison since they are in full agreement with standard BBA's optimality output.

They satisfy the DoOU and accordingly less accuracy whereas the linear estimation runs checking the power system observability [6]. For a specific IEEE power system, a PMU placement set solution is derived within a pre-given tolerance criterion [45].

Such a placement remarks that we can find an unobservable bus within the power grid which is immediately adjacent with observable power network nodes. This method has a factor leading to success in a manner as it guarantees a least distance between an unobservant power network node by either traditional or synchronized measurements with a neighborhood consisting of observable buses.

Complete observability as well as partial observability is two topological scenarios to choose appropriate PMUs in numbers, their locations around a power transmission grid aiming at complete or incomplete power system monitoring [17], [39]. Also, through the run of state estimation tool, we can examine when we lose the observability in any contingency or abnormal situation occurring during the performance of the power grid operation [1]-[2].

DoOU is stated as the situation in which do not permit buses not monitored to link together as declared in [17], [38]. This means that an unobservable bus is linked to observable buses. Based on this statement, we develop a binary integer linear program as well as a nonlinear model to return the appropriate PMU numbers and their locations for such a condition [45]-[52].

This study suggests a programming procedure being solved by mathematical and evolutionary algorithms that relies on the context of the situation of unobservability with degree one [17], [34]-[40].

In order to get optimality, the BIP model into a nonlinear model is converted after analyzing convergence to desirable outcomes from an evolutionary perspective. A discrete, continuous, and evolutionary approach is presented to demonstrate that achieving partial observability using only PMUs can be accomplished exactly within a reasonable runtime.

GAs and BPSO are employed to address this combinatorial optimization problem. Let us examine the performance of BPSO on 30, 57 and 118-bus systems.

Table 25 displays the last iteration of BPSO to get the best possible optimum point. BPSO is an extended version of PSO presented in [23] that results in an optimum solution point where the cost function is minimized [17], [39]. The heuristic algorithm leads to an optimal solution where the objective function is the least value. The optimum solution point is found to be as follows: $\vec{x} = \{2, 10, 15, 27\}$ which agrees to the one found in [39]-[40]. This solution is a constrained non-unique global optimum point.

Table 25. Iterative Process produced by the execution of BPSO on IEEE-30 bus system

Iteration	Best particle	Objective function
1	1	13.0000
2	1	11.0000
3	1	10.0000
4	1	10.0000
5	1	10.0000
6	1	7.0000
7	1	7.0000
8	1	7.0000
9	1	7.0000
10	1	7.0000
11	1	6.0000
12	1	4.0000
Elapsed time is 0.670224 seconds.		
PMU =		
2 10 15 27		

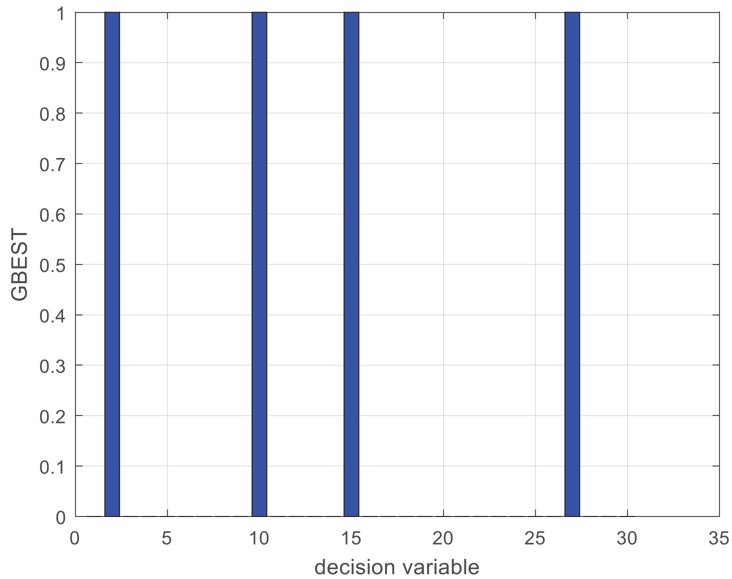

Figure 14. Plot diagram of Placement Result for 30-bus network

Table 26. Optimal Placement Outcome for 30-bus network

PMU locations with partial observability with a depth-of-one degree
2, 10, 15, 27

A plot diagram illustrated the PMU sites derived by PSO [23] using the 57-and -118 bus systems [56]. The PMU sites are displayed in the following tables and the plot diagrams. This set solution is an adequate configuration under the DoOU [39]. PSO is executed on the 57 bus system and the last iteration is displayed in Table 27. The objective function results in a least value at specific PMU locations (Fig.15). Tables 28 illustrate the PMU sites for the DoOU related to the 57-bus system.

PSO also gives the adequate PMU numbers for the 118 bus system. The last iteration is given in Table 29. A plot diagram illustrating the PMU sites derived by PSO [23], [55] is given in Figure 16 whereas the PMU localization sites are included in Table 30. All optima points satisfy the DoOU scenario for both case studies without constraint violation to exist [46].

As the BPSO algorithmic scheme is based on a population strategy, it mainly depends on the behavior of the particles in the population to define a solution with a promising quantity and quality as noted in [14], [23]-[25], [55]. They search the entire feasible region executing the solution space globally as well as using a local search to find the best possible solution [14], [55].

Two inertial weight parameters are used to get a proper tuning between the local and global best of particles at each iteration. Each inertia weight helps a lot to derive a precise optimum point. The optimum point satisfies the model's feasibility based on the given inertia parameter. Specifically we use either Random Inertia Weight that is displayed in the Esq. (28) [23], [55].

$$w = 0.5 + \frac{rand()}{2} \tag{28}$$

Or the Linear Decreasing Inertia Weight that is in Esq. (29) [55]:

$$W = W_{max} - \left(\frac{(W_{max} - W_{min}) \times j}{maxiter}\right) \tag{29}$$

Table 27. Iterative Process produced by the BPSO on 57 bus system

Iteration	Best particle	Objective function
1	1	30.0000
2	1	24.0000
3	1	23.0000
4	1	21.0000
5	1	20.0000
6	1	20.0000
7	1	18.0000
8	1	16.0000
9	1	16.0000
10	1	16.0000
11	1	15.0000
12	1	15.0000
13	1	14.0000
14	1	14.0000
15	1	12.0000
16	1	12.0000
17	1	11.0000
18	1	11.0000
19	1	11.0000
20	1	11.0000
Elapsed time is 0.619065 seconds.		
PMU =		
4 9 15 21 26 31 36 48 51 52 56		

```
min(BIMA*GBEST(:,j+1))

ans =

     1
```

As we can be observed, the DoOU scenario is formulated as a linear inequality function so as the element of the resulting vector corresponding to two terminals of the branch to be larger than 1 (**≥ 1**) [42]-[44]. The minimum output is 1 and presented in Table 27.

Table 27. Iterative Process produced by the BPSO on IEEE-57 bus system continued

(BIMA*GBEST(:,j+1))'
ans =
Columns 1 through 15

1	2	3	2	2	1	2	2	3	2	2	3	3	3	2
Columns 16 through 30														
1	1	3	2	2	2	1	3	3	3	1	1	2	1	1
Columns 31 through 45														
2	2	1	1	1	1	2	2	1	1	1	1	2	2	1
Columns 46 through 60														
1	1	2	2	2	1	3	2	2	2	1	1	2	1	1
Columns 61 through 75														
2	2	2	2	3	3	2	2	1	1	1	1	3	2	2
Columns 76 through 80														
1	2	2	2	2										

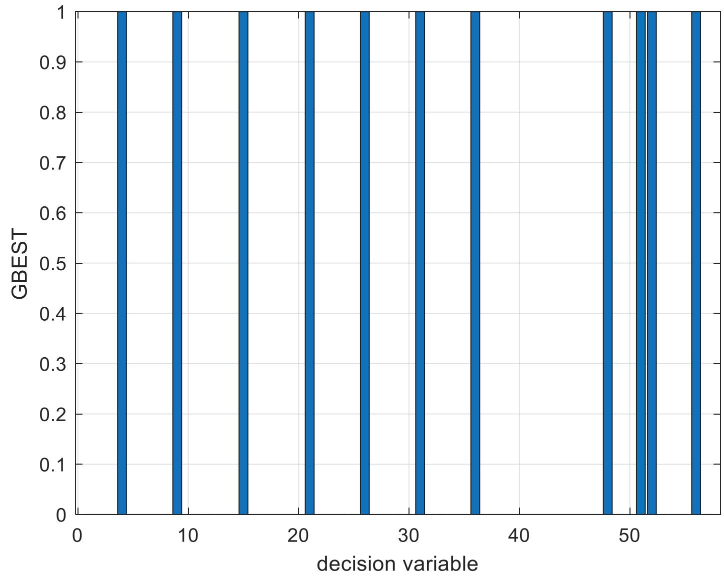

Figure 15. Plot diagram of Placement Result for 57-bus network

Table 28. Optimal Placement Outcome for 57-bus network

PMU locations with partial observability with a depth-of-one degree
4, 9, 15, 21, 26, 31, 36, 48, 51, 52, 56

Table 29 shows the log file produced by the BPSO to give an optimal solution. A PSO has been used to find the global minimum, applied to the 118 bus system [14], [23], [55]. The optima results are shown in Table 30. We confirm the total PMUs number for incomplete observability and their relative sites around the power grid. So, the accelerated BPSO has the capability to execute global search and local search of the solution region [13], [14], [23], [55]. A solution pool of optima points is derived by BPSO and they are illustrated in Table 31.

Table 29. Iterative Process produced by PSO applied to 118 bus system

Iteration	Best particle	Objective function
1	1	70.0000
2	1	70.0000
3	1	68.0000
4	1	56.0000
5	1	51.0000
6	1	51.0000
7	1	49.0000
8	1	47.0000
9	1	40.0000
10	1	40.0000
11	1	38.0000
12	1	34.0000
13	1	34.0000
14	1	34.0000
15	1	30.0000
16	1	30.0000
17	1	27.0000
18	1	24.0000
19	1	24.0000
20	1	24.0000
21	1	24.0000
22	1	24.0000
23	1	24.0000
24	1	23.0000
25	1	19.0000
26	1	18.0000
27	1	18.0000
Elapsed time is 33.531342 seconds.		
PMU =		
Columns 1 through 17		
8 12 17 22 27 34 42 49 54 60 65 70 77 85 92 94 105		
Column 18		
109		

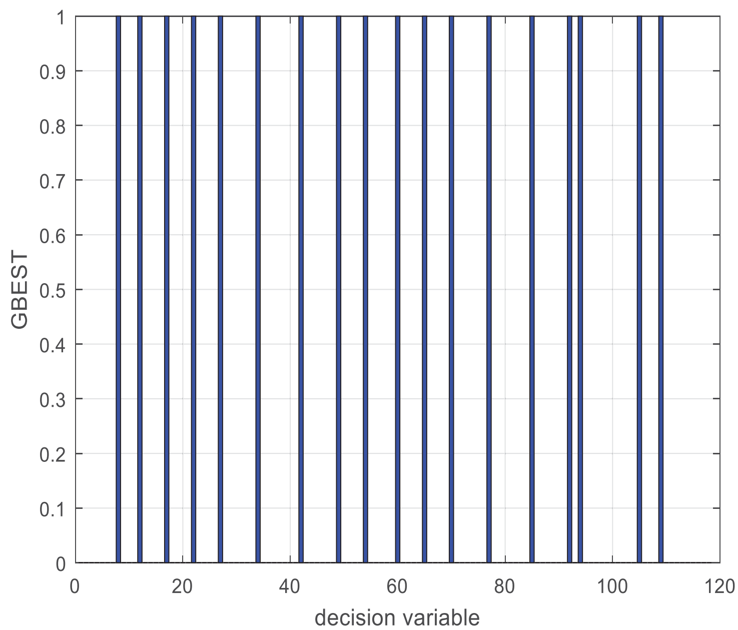

Figure 16. Plot diagram of Placement Result for 118-bus network

Table 30. Optimal Placement Outcome for 118-bus network

PMU locations with partial observability with a depth-of-one degree
8, 12, 17, 22, 27, 34, 42, 49, 54, 60, 65, 70, 77, 85, 92, 94, 105, 109

Table 31. PMU Locations for Incomplete Observability using BPSO

Test System	PMU	PMU location (Bus #)
IEEE 14 bus	2	4, 6
IEEE 30 bus	4	2, 10, 15, 27
IEEE 57 bus	11	4, 9, 15, 21, 26, 31, 36, 48, 49, 52, 56
		4, 9, 15, 21, 26, 31, 36, 48, 51, 52, 56
		4, 9, 15, 21, 26, 31, 36, 48, 50, 52, 56
		6, 10, 15, 20, 24, 29, 32, 37, 41, 48, 54
		4, 10, 15, 21, 24, 29, 32, 37, 41, 48, 55
		5, 10, 15, 20, 24, 29, 32, 37, 41, 48, 53
		5, 10, 15, 20, 24, 29, 32, 37, 41, 48, 54
		5, 10, 15, 20, 24, 29, 32, 37, 41, 48, 55
		4, 10, 15, 21, 24, 29, 32 , 37, 41, 48, 54
		4, 10, 15, 21, 24, 29, 32 , 37, 41, 48, 54
		4, 10, 15, 21, 24, 29, 32, 37, 41, 48, 53
		6, 10, 15, 20, 24, 29, 32, 37, 41, 48, 54
		6, 10, 15, 20, 24, 28, 32, 37, 41, 48, 54
		6, 10, 15, 20, 24, 29, 32, 37, 41, 48, 55

Table 31. PMU Locations for Incomplete Observability using BPSO continued

Test System	PMU	PMU location (Bus #)
IEEE 118 bus	18	8, 12, 17, 21, 27, 34, 40, 49, 54, 61, 68, 70, 77, 85, 92, 96, 105, 111
		8, 12, 17, 21, 27, 34, 40, 49, 54, 61, 68, 70, 77, 82, 86, 92, 105, 111
		8, 12, 17, 21, 27, 37, 45, 49, 54, 62, 65, 70, 77, 85, 92, 96, 105, 111
		8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 68, 71, 77, 80, 85, 92, 105, 110
		8, 12, 17, 21, 27, 37, 45, 49, 54, 61, 65, 70, 77, 85, 92, 96, 106, 110
		8, 12, 17, 21, 27, 34, 37, 49, 54, 61, 68, 70, 77, 85, 92, 95, 106, 110
		8, 12, 17, 21, 27, 37, 44, 49, 54, 61, 69, 70, 77, 85, 92,96, 106, 110
		8, 12, 17, 21, 27, 37, 43, 49, 54, 61, 65, 70, 77, 85, 92, 97, 105, 112
		8, 12, 17, 21, 27, 37, 43, 49, 54, 62, 65, 70, 77, 85 92, 96, 107, 110
		8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 65, 70, 77, 85, 92, 97, 107, 110
		8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 71, 77, 85, 92, 96, 103, 105, 116
		8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 71, 77, 82, 86, 92, 105, 110, 116
		8,12, 17, 22, 27, 34, 37, 49, 54, 61, 68, 71, 77, 85, 92, 96, 105, 110
		8, 12, 17, 21, 27, 37, 43, 49, 54, 61, 69, 72, 77, 82, 85, 92, 103, 105
		8, 12, 17, 21, 27, 34, 41, 49, 54, 62, 65, 70, 77, 80, 85, 92, 105, 110
		8, 12, 17, 21, 27, 37, 44, 49, 54, 61, 65, 72, 75, 80, 85, 92, 103, 105
		8, 12, 17, 21, 27, 34,40, 49, 54, 61, 68, 70, 77, 85, 92, 96, 105, 112
		8, 12, 17, 21, 27, 34, 37, 49, 54, 61, 69, 72, 77, 82, 85, 92, 105, 112
		8, 12, 17, 21, 27, 34, 37, 49, 54, 61, 70, 77, 81, 82, 85, 92, 105, 109
		8, 12, 17, 21, 27, 34, 41, 49, 54, 61, 65, 70, 77, 85, 92, 96, 105, 110
		8, 12, 17, 21, 27, 37, 44, 49, 54, 60, 65, 70, 77, 85, 92, 96, 105, 110

In that case, the objective function has non-strict global minimizer [14]; to contrast with the statement: an optimum point $\vec{x}^* \in \Omega$ is a strict global minimizer; strong global minimizer over feasible region Ω of the problem $min_{\vec{x}\in\Omega} J(\vec{x})$ if $\forall x \in \Omega, (\vec{x} \neq \vec{x}^*) \Rightarrow (f(\vec{x}^*) < f(\vec{x}))$ (30) [13], [45]-[52]. This solution covers the case where nodes being not monitored have been connected with observable adjacent nodes by synchronized measurements as firstly presented in [14], [17], [34]-[41].

GA is implemented for solving mixed integer constrained optimization problems [13]-[22], [61]. GA solver uses a set of starting points called population and iteratively generates better points from the population. The optimal results obtained by the GA solver are illustrated in Table 32 [22], [61].

Table 32 displays the PMU set solution delivered by GA solver included in MATLAB Global Optimization library [62]. Using the trial-and-error process, we gather a solution pool of optima points over the feasible region constituted by the objective function, constraints and binary restriction.

The stopping criteria of the algorithm include the number of generations, the computation limit and the function evaluations within a certain tolerance [13]-[22], [61]. Also, GAs collects a solution pool of optima points; all characterized as globally solutions shown in Table 32 [13]. Each set solution is produced after a trial-and-error effort to find an acceptable constraint; function fitness limits [13].

Table 32. PMU Locations for Incomplete Observability using GA

Test System	PMU	PMU location (Bus #)
IEEE 14 bus	2	4, 6
IEEE 30 bus	4	2, 10, 15, 27
IEEE 57 bus	11	4, 10, 15, 20, 24, 29, 32, 37, 41, 48, 53
		4, 10, 15, 20, 24, 29, 32, 37, 41, 48, 54
		4, 9, 15, 21, 26, 31, 36, 48, 50, 52, 56
		4, 9, 15, 21, 26, 31, 36, 48, 49, 52, 56
		4, 10, 15, 21, 24, 29, 32, 37, 41, 48, 53
		4, 10, 15, 20, 24, 29, 32, 37, 41, 48, 53
		5, 10, 15, 20, 24, 29, 32, 37, 41, 48, 53
		5, 10, 15, 20, 24, 29, 32, 37, 41, 48, 54
		4, 10, 15, 20, 24, 29, 32, 37 , 41, 48, 55
		4, 9, 15, 21, 26, 31, 36, 48, 51, 52, 56
		4, 9, 15, 21, 26, 31, 36, 47, 49, 52, 56
		6, 10, 15, 20, 24, 28, 32, 37, 41, 48, 54
		6, 10, 15, 20, 24, 28, 32, 37, 41, 48, 53
IEEE 118 bus	18	8, 12, 17, 21, 27, 34, 37, 49, 54, 61, 65, 70, 77, 80, 85, 92, 105, 109
		8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 68, 71, 77, 80, 85, 92, 105, 110
		8, 12, 17, 21, 27, 34, 37, 49, 54, 61, 65, 70, 77, 80, 85, 92, 105, 110
		8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 65, 70, 77, 80, 85, 92, 105, 110
		8, 12, 17, 22, 27, 34, 41, 49, 54, 61, 65, 70, 77, 80, 85, 92, 105, 110
		8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 65, 70, 77, 85, 92, 96, 105, 110
		8, 12, 17, 21, 27, 34, 37, 49, 54, 61, 65, 70, 77, 85, 92, 96, 105 110
		8,12, 17, 21, 27, 34, 37, 49, 54, 61, 65, 70, 77, 82, 85, 92, 105, 110
		8, 12, 17, 21, 27, 34, 37, 49, 54, 61, 65 , 70 , 80, 85, 92, 103, 105, 118
		8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 65, 70, 77, 80, 85, 92, 103, 105
		8, 12, 17, 21, 27, 34, 42, 49, 54, 61, 65, 70, 77, 85, 92, 97, 105, 109
		8, 12, 17, 21, 27, 34, 37, 49, 54, 62, 65, 70, 77, 85, 92, 94, 105, 110

Let us examine the performance of GA on 30-, 57- and 118-bus systems. A plot diagram illustrated the PMU sites derived by GA using the 30 bus and 118- systems [61]. The PMU sites are displayed in the table following the plot diagrams. Each individual is a PMU site including in the placement result.

We get each individual and the fitness function in combination with the previous run of the algorithmic model running in the MATLAB platform [61]-[62]. Finally, an optimal solution is reached with the best penalty value as well as a mean penalty value as displayed in Figures 17 & 18.

The optimization is terminated with an average change in the penalty fitness value which was found less than the tolerance related to the function evaluations [61]-[62]. Also, the constraint violation is less than the certain limit [61]-[62]. Each chromosome included in the placement set displays the binary nature of the optimization problem [13], [61]-[62]. An optimal solution is derived within satisfied termination criteria at the end of the optimization [14].

There exist power nodes not being observable; although its neighborhood includes observable network nodes being monitored by synchronized measurements as displayed in the Fig.1. As a benchmark system, we firstly use the IEEE-30 bus system, produced by GA solver [56]-[57]. The GA delivers an optimum point solution by using the tuning operators [22], [62].

As a case study, we use the IEEE-30 bus system as found in [56]-[57]. GA optimizes the constraint binary program and the set solution is $\vec{x} = \{2, 10, 15, 27\}$ under the concept of DoOU. That optimal solution was found in Rohit Babu's work [39]. This is an extra validation for our implementation of partial observability from different algorithmic standpoints. An unobservable node can be found which is adjacent with a neighborhood of observable buses by PMUs posed at a selected power network.

This justifies the fact that a lesser PMUs number is found than the case study of complete observability conditions [39]. The optimal solution displayed in Fig.17 and Table 33. As shown in Fig.17, the fitness function is minimized and each individual is an optimum point [55].

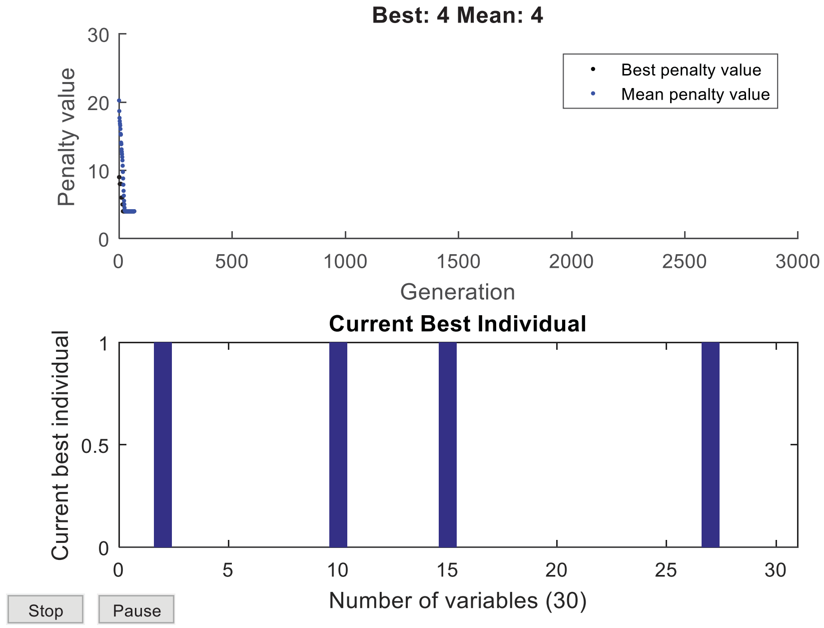

Figure 17. Plot diagram of Placement Result for 30-bus network

Table 33. Optimal Placement Outcome for IEEE-30-bus network

PMU locations with partial observability with a depth-of-one degree
2, 10, 15, 27

Using the 57-and -118 bus system [56], some plot diagrams are produced by GA [22], [62]. The PMU sites are displayed in following tables and the plot diagrams. This set solution is an adequate configuration under the concept of partial observability with degree one [39]. The last iteration for GA executed on the 57 bus system is displayed in Table 34. GA optimizer function is a derivative free optimizer spending relatively a large amount of function evaluations to get an optimum point [14].

We select an appropriate population to avoid a convergence to a local optimum point too soon. Additionally, we properly tune the Elite Count to optimize the best individual of a defined population, which creates the new population using the next generation [14].

Hence, GA gives a global optimal point with high probability. Initially we start the iterative process selecting the population in a double vector where each row represents a vector consisting of bit strings [62]. The fitness function is minimized under tolerances, including a penalty parameter to penalize infeasible individuals. Hence, infeasible individuals are rejected.

By default, the initial population is created by MATLAB's command called CreationFcn [22], [62]. Otherwise, an initial population can be built. The default parameter to build a population in a double vector is the selection uniform.The GA optimizes an initial starting point based on a population of individuals while the fitness function is optimized.

Hence, a measure of how optimal the solution is finally got. The fitness function is minimized within tolerances and includes a penalty term; this methodology penalizes infeasible individuals and refuses those that are infeasible during successive generations in the iterative process [22], [55], [62].

We use a population in a double vector including vectors consisting of individuals in a binary format. With a large population size, the GA returns a global solution point with high probability [19]-[22], [62]. The CreationFcn command creates the initial population for the GA optimizer function for the purpose of starting the iterative process towards optimality [62].

A new population is delivered as an output of crossover scattered operators [19]-[22], [62]. Scattered operator delivers a random binary vector and selects the genes where the vector is a 1 from the first parent, and then genes where the vector is a 0 from the second parent, and combines the genes to form the child [22], [55], [62].

Taking into account these parameters tuned by the user, the GA minimizes the constraint binary program by properly adjusting its constants for the purpose of getting the optimality [22], [62]. The entire procedure is minimized until a specific stopping tolerance criterion is found.

The GA optimizer delivers an optimum point whereas the model's feasibility is satisfied within pre-given tolerances, that is, function and constraint tolerances shown in Table 34.

Table 34 displays the iterative process produced by GA to find optimality using the IEEE-57 as a case study. As observed, GA results in the best possible solution as those found by BBA algorithmic scheme [14]. GA results in an optimal PMUs number lesser than the CO topological observability. GA terminates the whole optimization process and returns an optimal solution within pre-defined tolerance criteria. Those solutions are considered to be non-strict global minima [17], [39].

We consider the power system size; we conclude that a suitable population size is selected taking into account it is equal to a PopulationSize equal to 1000. The last Generation, function evaluations are displayed within a pre-given stall generation. GA solver succeeds to attack the constraint integer models towards a global optimality. The optimization is terminated within the best fitness value without constraint violation [14], [19]-[22], [55], [62].

Both case studies leads to the least PMUs number which agree to those presented in [17] & [39] but the PMUs are placed in different sites within the set configuration solution. The optimal solution is displayed in Fig.18 and Table 35. Then another plot diagram is given using the IEEE-118 bus system. The optimal solution is displayed in Fig.19 and Table 36.

Each individual represent a PMU in the PMU placement set under the concept of DoOU. This optimum point covers that state where an unobservable bus is linked together with a neighborhood consisting by only observable network buses by PMUs [39]. Fig 18 illustrates the best individual representing PMU locations for the DoOU implementation wide-area monitoring scenario [34].

Each individual's fitness function is obtained by combining the results from the previous run of the algorithmic model, which is executed on the MATLAB platform [14], [62]. This paper presents also nonlinear programming algorithms for recognizing placement locations for PMUs in a power network under the concept of complete and incomplete observability.

Table 34. Iterative Process produced by GA tested on the IEEE-57 bus system

		Best	Mean	Stall
Generation	f-count	Penalty	Generations	
90	91000	11	11	47
91	92000	11	11	48
92	93000	11	11	49
93	94000	11	11	50
Optimization terminated: average change in the penalty fitness value less than options.FunctionTolerance				
and constraint violation is less than options.ConstraintTolerance.				
fval =				

Table 34. Iterative Process produced by GA tested on the 57 bus system continued

11
ans =
4 10 15 20 24 29 32 37 41 48 53
Elapsed time is 12.082298 seconds.
output =
problemtype: 'integerconstraints'
rngstate: [1x1 struct]
generations: 93
funccount: 94001
message: 'Optimization terminated: average change in the penalty fitness value less than opti…'
maxconstraint: 0
Set properties:
CreationFcn: @gacreationuniform
CrossoverFcn: @crossoverscattered
Display: 'iter'
EliteCount: 10
FunctionTolerance: 1.0000e-08
PlotFcn: {@gaplotbestf @gaplotbestindiv}
PopulationSize: 1000
SelectionFcn: @selectionstochunif

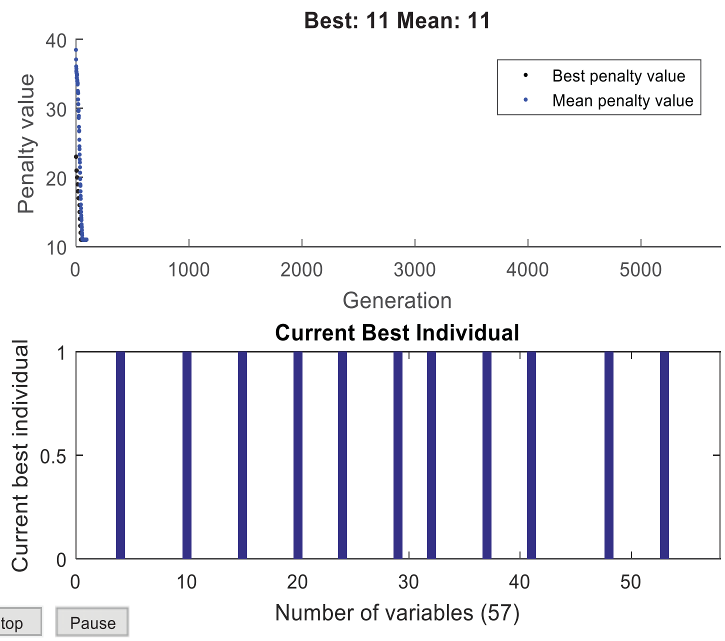

Figure 18. Plot diagram of Placement Result for 57-bus network

Table 35. Optimal Placement Outcome for 118-bus network

PMU locations with partial observability with a depth-of-one degree
4, 10, 15, 20, 24, 29, 32, 37, 41, 48, 53

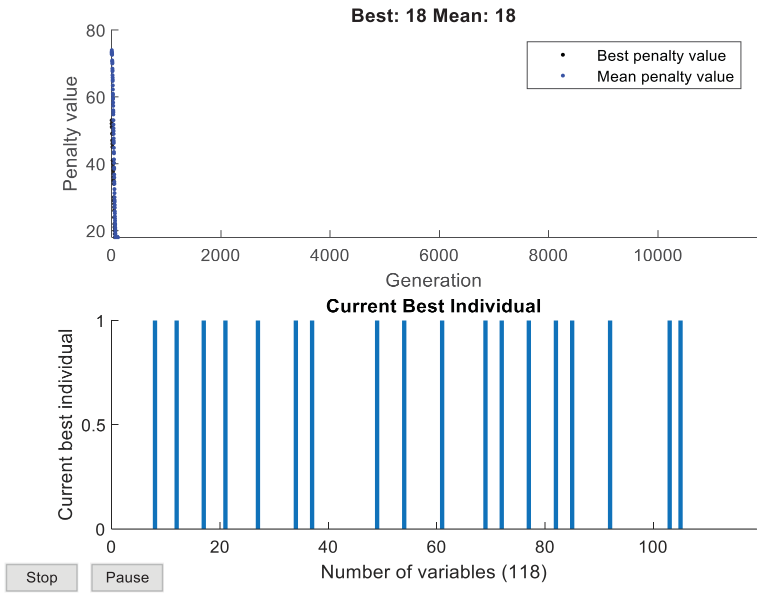

Figure 19. Plot diagram of Placement Result for 118-bus network

Table 36. Optimal Placement Outcome for 118-bus network

PMU locations with partial observability with a depth-of-one degree
8, 12, 17, 21, 27, 34, 37, 49, 54, 61, 69, 72, 77, 82, 85, 92, 103, 105

The model is then transformed into a nonlinear problem with either a quadratic or linear objective function. The minimization model is well optimized based on the FMINCON optimizer routine included in MATLAB optimization library [61]. FMINCON utilizes either SQP methods or IPMs to find an optimal solution.

Optima points are delivered by executing the nonlinear program with FMINCON [61]. As a consequence, all minima point of the objective declared in a quadratic format as follows $J(\vec{x}) = \sum_n x_i^2, \vec{x}|\vec{x} \in [0,1]$ (31) are accepted [45]-[52]. All optima points are located at the same optimal objective function value. They are interpreted as non-strict global optimum points; SQP and IPMs locate non-strict local minimizers at the same optimal objective function value [45]-[52]. SQP and IPMs locate non-strict local minimizers over the feasible set [48]-[49].

An NLP solver needs the differentiation of the objective and constraint functions to start the iterative process [61]. A program is utilized to analytically derive the objective function gradient and the constraint function Jacobian matrix within the MATLAB language [61]. Termination criteria are applied by default, or the parameter settings are adjusted as needed [61]-[62]. Hence, the optimization is terminated in less time avoiding computational burden [61]. Both nonlinear algorithms result in optimal solutions; weak global optima with the same computational complexity [45]-[52].

A weak global minimizer is $if\ J(\vec{x}) \geq J(\overrightarrow{x^*}), \forall\vec{x} \in \Re^n$ (31). So, the objective function delivers non-strict global minima [45]-[52]. IPMs optimize the nonlinear objective function consisting of the same constraint function over an unbounded region on the decision variables [45]-[52]. These nonlinear algorithms locate an alternative solution at the optimal objective cost value [45]-[47].

The SQP and IPMs are executed on the IEEE-57 and 118- bus systems as a benchmark study to prove the validation of the above optima points. Table 37 & 38 and plot diagrams displayed in Fig.20 and Fig.21 display the placement set solution derived by SQP methods. Also, IPMs result in the same least objective function with different location sites displayed in Table 39 & 40 and Fig.22 & Fig.23.

SQPs and IPMs spent function evaluation for the purpose of getting optimality within certain optimality tolerances and criteria [45]-[52]. Both algorithms locate a local minimum point satisfying the constraints counting reasonable function evaluations required to locate it [45]-[52].

The optimization process is considered to be completed because the objective function is non-decreasing in feasible directions. Each optimum point satisfies criteria such as tolerances, constraint violation, relative first-order optimality, optimality Tolerance and constraint Tolerance [45]-[50].

Each solution point is also a global one within a certain stopping criterion. Relative Figures illustrate the adequate convergence properties needed to deliver a locally optimal solution by either SQP or IPMs. The cost function value is the least as the minimization requires. The current point illustrates a binary PMU set solution derived within no constraint violation [46].

Also, the number of function evaluations is shown within an adequate step-size during the descent direction in the direction of a locally optimal solution with First-order optimality [45]-[48]. Although these solutions are locally found, we can characterize them globally [14], [45]-[50]. Nonlinear algorithms adopt penalty methods to accept the step size without rejecting it [45]-[46], [48]-[49], [53].

Using the SQP methods to attack the nonlinear model, the penalty function doesn't reject the unity-step during the iterative process. Hence, the Maratos effect is avoided [45]-[50], [53]. SQP and IPMs spend a reasonable amount of function evaluation to get optimality within certain optimality tolerances and criteria [61]. More details about the performance of the SQP and IPMs methods in NLP solving can be found in [45], [48]-[49]. So, the local minima obtained by SQP and IPMs are acceptable since the constraint function feasibility is found to be satisfied by the simulation results [61]-[63].

Thus, no constraint violation exists, confirmed by the message on the termination of the iterations generated by the optimizer FMINCON for getting the purpose to reach an optimal solution; a locally optimum point [61]. Each optimal solution is locally found by the optimizer function characterized globally after a comparative study with branch-and-bound standard optimality metrics [45]-[52].

Using a suitable MATLAB code which is compatible with the MATLAB environment, we give a solution pool about the partial observability with degree one [17], [39]. Hence, the global optimal solution is attained [14], [45]-[49], [52].

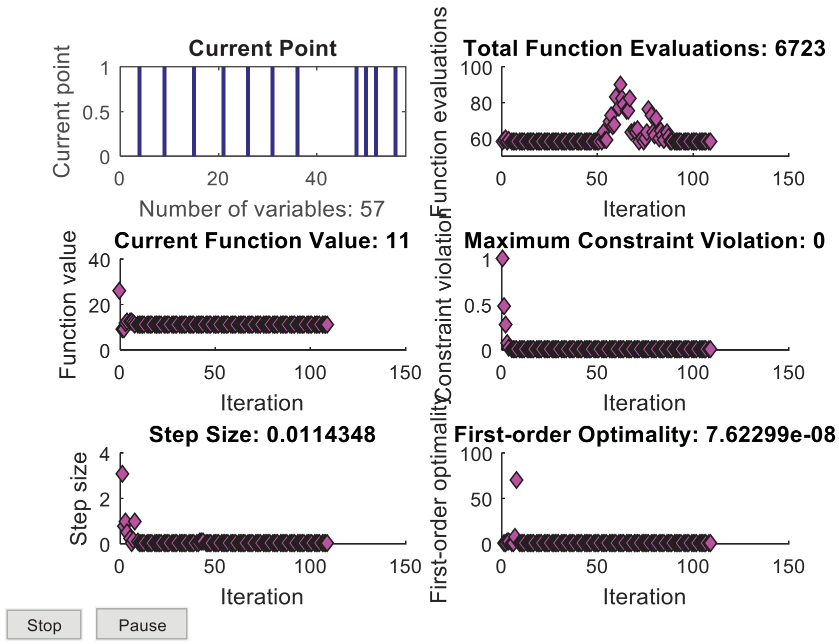

Figure 20 Plot diagram of Placement Result for 57-bus network

Table 37. Optimal Placement Outcome for 57-bus network

PMU locations with partial observability with a depth-of-one degree
4, 9, 15, 21, 26, 31, 36, 48, 50, 52, 56

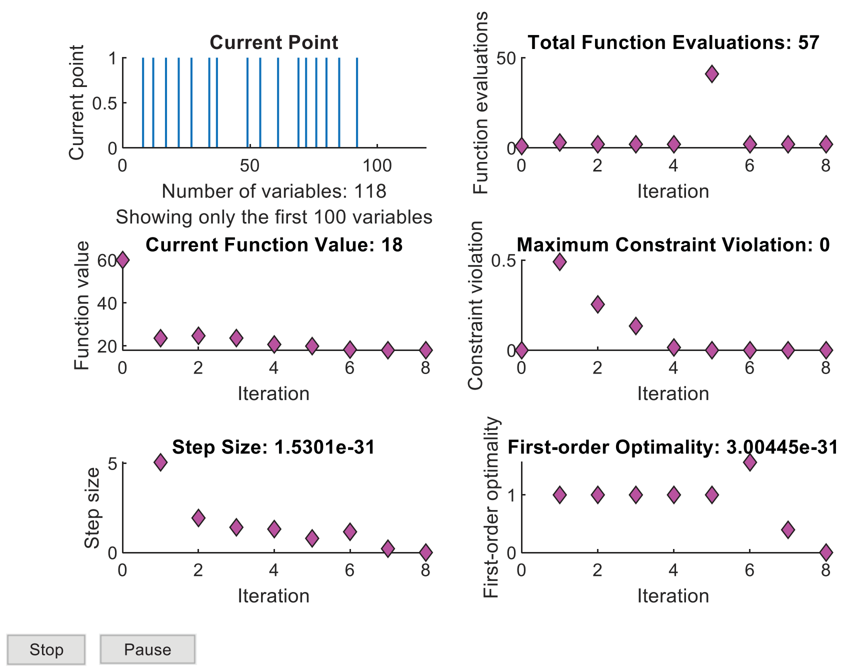

Figure 21. Plot diagram of Placement Result for 118-bus network

Table 38. Optimal Placement Outcome for 118-bus network

PMU locations with partial observability with a depth-of-one degree
8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 69, 72, 76, 80, 85, 92, 103, 105

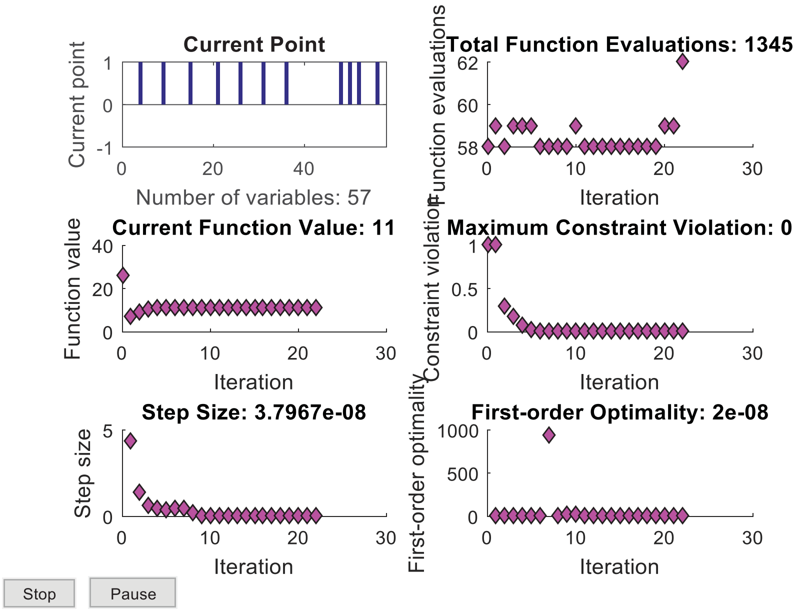

Figure 22. Plot diagram of Placement Result for 57-bus network

Table 39. Optimal Placement Outcome for 57-bus network

PMU locations with partial observability with a depth-of-one degree
4, 9, 15, 21, 26, 31, 36, 48, 50, 52, 56

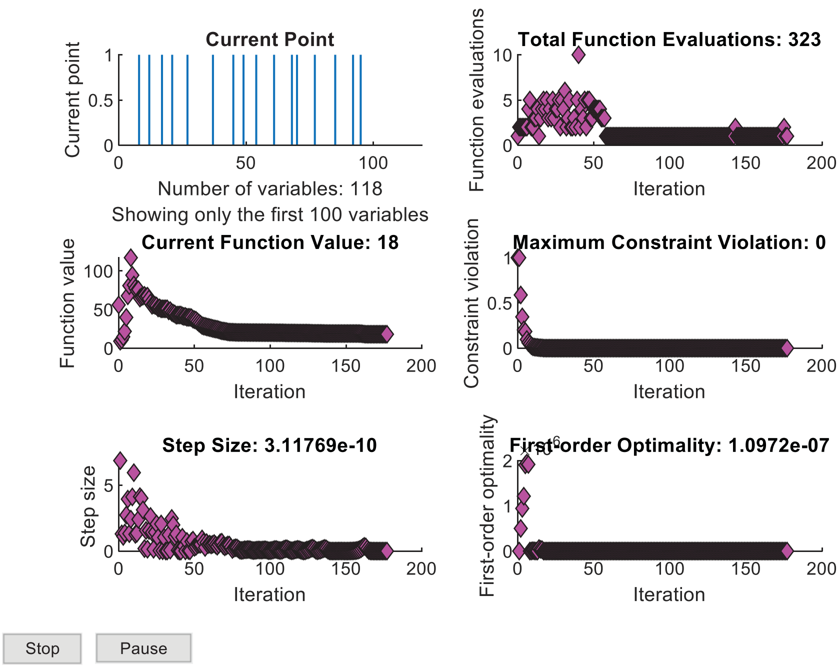

Figure 23. Plot diagram of Placement Result for 118-bus network

Table 40. Optimal Placement Outcome for 118-bus network

PMU locations with partial observability with a depth-of-one degree
8, 12, 17, 21, 27, 37, 45, 49, 54, 61, 68, 70, 77, 85, 92, 95, 105, 112

Table 41. PMU Locations for Incomplete Observability using SQP and IPMs

Test System	PMU	PMU location (Bus #)
IEEE 14 bus	2	4, 6
IEEE 30 bus	4	1, 10, 15, 27
		2, 10, 15, 27
IEEE 57 bus	11	4, 9, 10, 15, 21, 26, 31, 36, 48, 52, 56
		4, 9, 15, 21, 26, 31, 36, 48, 49, 52 56
		4, 9, 15, 21, 26, 31, 36 , 47, 49, 52 56
		4, 9, 15, 21, 26, 31, 36, 46, 49, 52, 56
		4, 9, 15, 21, 26, 31, 36, 48, 49, 52, 56
		4, 9, 15, 21, 26, 31, 36, 48, 51, 52, 56
		4, 9, 15, 21, 26, 31, 36, 48, 50, 52, 56
		4, 10, 15, 20, 24, 29, 32, 37, 41, 48, 54
		4, 10, 15, 21, 24, 29, 32, 37, 41, 48, 54
		4, 10, 15, 21, 24, 29, 32, 37, 41, 48, 55
		4, 10, 15, 21, 24, 29, 32, 37, 41, 48, 53
		4, 10, 15, 20, 24, 29, 32, 37, 41, 48, 53
		5, 10, 15, 20, 24, 29, 32, 37, 41, 48, 54

Table 41. PMU Locations for Incomplete Observability using SQP and IPMs continued

Test System	PMU	PMU location (Bus #)
IEEE 57 bus	11	4, 10, 15, 20, 24, 29, 32, 37, 41, 48, 53
		5, 10, 15, 20, 24, 29, 32, 37, 41, 48, 54
		5, 10, 15, 20, 24, 29, 32, 37, 41, 48, 55
		6, 10, 15, 20, 24, 29, 32, 37, 41, 48, 54
		6, 10, 15, 20, 24, 27, 32, 37, 41, 48, 53
		6, 10, 15, 20, 24, 29, 32, 37, 41, 48, 53
		6, 10, 15, 20, 24, 28, 32, 37, 41, 48, 54
		6, 10, 15, 20, 24, 28, 32, 37, 41, 48, 53
		6, 10, 15, 20, 24, 29, 32, 37, 41, 48, 55
IEEE 118 bus	18	8, 12, 17, 22, 27, 34, 40, 49, 54, 61, 68, 71, 77, 80, 85, 92 105, 111
		8, 12, 17, 22, 27, 34, 40, 49, 54, 62, 65, 71, 77, 80, 85, 92 105, 111
		8, 12, 17, 22, 27, 34, 40, 49, 54, 62, 65, 71, 77, 85, 92, 96 105 111
		8, 12, 17, 22, 27, 34, 40, 49, 54, 61, 65, 71, 77, 80, 85, 92 105, 110
		8, 12, 17, 22, 27, 34, 40, 49, 54, 61, 68, 71, 77, 80, 85, 92 105, 110
		8, 12, 17, 22, 27, 34, 40, 49, 54, 62, 65, 71, 77, 85, 92, 96, 105 110

Test System	PMU	PMU location (Bus #)
IEEE 118 bus	18	8, 12, 17, 22, 27, 34, 40, 49, 54, 61, 68, 71, 77, 85, 92, 96 105, 110
		8, 12, 17, 22, 27, 34, 40, 49, 54, 61, 65, 71, 77, 85, 92, 96 105, 110
		8, 12, 17, 22, 27, 34, 40, 49, 54, 62, 65, 71, 77, 80, 85, 92 105, 110
		8, 12, 17, 22, 27, 34, 40, 49, 54, 61, 65, 71, 77, 80, 85 92, 105, 109
		8, 12, 17, 22, 27, 34, 40, 49, 54, 61, 68, 71, 77, 80, 85 92, 105, 109
		8, 12, 17, 22, 27, 34, 40, 49, 54 , 62, 65, 71, 77, 80, 85, 92 105, 109
		8, 12, 17, 21, 27, 37, 44, 49, 54, 60, 65, 70, 76, 80, 85, 92 103, 105
		8, 12, 17, 21, 27, 37, 44, 49, 54, 60, 65, 70, 80, 85, 92, 103 105, 118
		8, 12, 17, 22, 27, 34, 37, 49, 54, 62, 65, 71, 77, 80, 85, 92 105, 112
		8, 12, 17, 22, 27, 34, 40, 49, 54, 62, 65, 71, 77, 85, 92, 96 105, 112
		8, 12, 17, 22, 27, 34, 37, 49, 54, 61, 71, 77, 80, 85, 92, 105 109, 116

The objective value could be interpreted as a weak global minimizer provided that the results agree with those found in solving the $0/1$ MILP model [49]. A solution pool constituted by a number of optimal solutions is presented in Tables 41. All solutions have the same quantity. Also, we can solve the constraint $0/1$ integer program through open-source integer solvers embedded in OPTI-toolbox [63]. Different ILP solvers are utilized such as SCIP, GLPK, CBC, LPSOLVE and Bonmin [63].

Also, MOSEK can be used for the incomplete observability with degree one [67]. Gurobi optimization engine uses a branch-and-bound algorithm to find an optimal solution to the global optimization problem solving [64]. As can be observed, Gurobi and MOSEK comes across the exact optima points; globally optimal solution found inside a zero-gap tolerance [64]-[67].

Tables 42 & 44 illustrate the optima points for the concept of DoOU using the ILP solvers included in OPTI-toolbox [17], [39], [63]. The constrained $\mathbf{0/1}$integer programming is solved using commercial or open-source integer linear programming (ILP) optimizer functions [61]-[67].

All solvers find a globally optimal solution within a zero-gap tolerance and a relative gap within a pre-given tolerance criterion equal to zero [61]-[65].Tables 42-44 summarizes the simulation results for incomplete observability with degree one [17]. Table 44 illustrates the PMUs number and their placement locations for an implementation of state estimation in real-time produced by Gurobi and MOSEK [36]. Gurobi and MOSEK optimizer engines find the identical PMU nubmers but in different locations within a zero-gap tolerance. This happens because a different branching strategy is followed in order to construct the binary tree and finally to result in an optimal solution [45]-[50].

MOSEK adopts an interior-point method in combination with a primal-dual-simplex algorithm to solve the constraint binary model. As the simulation results shows, primal dual interior point methods have been proved useful to solve the relaxed problem during the iterative process where nodes are explored and infeasible regions are pruned [67].

The LP solver solves the relaxed problems, prunes infeasible regions where objective values are found bigger than the upper bound [45]. An optimal solution has been located satisfying the relative gap tolerance whereas the absolute gap is equal to zero [67]. Hence, a global solution is achieved.

Table 42. Optimal Placement Outcome for Incomplete Observability with degree one

IEEE system	ILP routines for solving the 0-1 Incomplete observability with degree one			
	SCIP	INTLINPROG	GLPK	LPSOLVE
	Optimal PMU locations			
14 bus	4, 6	4, 6	4, 6	4, 6
30 bus	2, 10, 15, 27	2, 10, 15, 27	2, 10, 15, 27	2, 10, 15, 27
57 bus	4, 10, 15, 20, 24, 29 32, 37, 41, 48, 54	4, 10, 15, 21 24, 29, 32, 37 41, 48, 54	4, 9, 15, 21, 26 31, 36, 48, 51 52, 56	4, 9, 15, 21, 26, 31, 36, 48 50, 52, 56
118 bus	8, 12, 17, 21, 27, 37 45, 49, 54, 61, 65, 70 77, 85, 92, 96, 106 110	8, 12, 17, 21 27, 37, 45, 49 54, 61, 69, 70 80, 85, 92, 107 110, 118	8, 12, 17, 21 27, 34, 37, 49 54, 61, 69, 72 77, 80, 85, 92 103, 105	8, 12,17, 21, 27, 37, 43, 49, 54 61, 69, 72, 76, 80, 85, 92 105, 110

Table 43. Optimal Placement Outcome for Incomplete Observability with degree one

IEEE Systems	ILP routines for solving the 0-1 Constraint Integer Program under Incomplete observability with degree one	
	CBC	bonmin
14-bus	4, 6	4, 6
30-bus	2, 10, 15, 27	2, 10, 15, 27
57- bus	4, 9, 14, 15, 21, 26, 31, 36, 49,52, 56	6, 10, 15, 20, 24, 29, 32, 37, 41, 48, 53
118-bus	8, 12, 17, 21, 27, 34, 40, 49, 54 61, 68 70, 77, 85, 92, 96, 105, 111	8, 12, 17, 21, 27, 37, 43, 49, 54, 61, 69 72, 76, 80, 85, 92 105, 110

Table 44. PMU placement locations for DoOU implemented by GUROBI and MOSEK

Test System	PMU	PMU location (Bus #)	
		GUROBI	MOSEK
IEEE 14 bus	2	4, 6	4, 6
IEEE 30 bus	4	2, 10, 15, 27	2, 10, 15, 27
IEEE 57 bus	11	4, 10, 15, 21, 24, 29, 32, 37, 41 48, 53	5, 10, 15, 20, 24, 29, 32, 37, 41 48, 53
IEEE 118 bus	18	8, 12, 17, 21, 27, 37, 45, 49 54 61, 65, 70, 77, 80, 85, 92 105, 110	8, 12, 17, 21, 27, 37, 43, 49, 54, 61 65, 72, 75, 80, 85, 92, 103, 105

7. Study's Impact on the PMU number related to DoOU

Partial observability means the state where the PMUs posed at selected sites by appropriate algorithms cannot give all the needed information for all system buses. Incomplete observability determines the underlying idea in which we measure the distance between unobservable buses from the closest bus in attendance in a local area [17], [39]. The related basis of the idea focuses on a methodical procedure of placing PMUs within a power grid, in stages if it is required [34].

Hence, the regions not being observable to decrease little by little up to point in time allows the power system to be fully observable by a sufficient PMUs number optimally placed within the power grid. We employ the depth-of-one-unobservability in this work. Bigger in quantity depths of unobservability brings about less close to true state of the unobservable power system buses and more uncertainties on margins produced by system state estimation [34].

We deal with DoOU where all of the neighbouring buses of any unobservable bus must be observable. Less unobservability situation results in less accuracy when we execute the state estimation tool in real-time [39].

Discrete, continuous and evolutionary algorithms were executed to get the optimum point under this circumstance. Thus, the capability of monitoring the power network under DoOU was assumed and became real when an appropriate PMUs number was found [17], [39].

Displaying some placement result, observed regions within one unobserved network node are adjacent with power nodes being monitored by lesser PMUs than those in complete. The optimum point is attained minimizing a linear or a quadratic objective function in binary as well as in continuous decision domain.

More than one depth observability leads to lesser accuracy of the state estimation routine. We analyze and study the DoOU concept where a distance of one depth between a power network node not being observed and observed adjacent nodes is taking into account for implementation.

The suggested model is successfully carried out on standard power networks in finding an appropriate PMUs number not satisfied by a full condition of power observability but on occasion where an unobserved network node is adjacent with observable power nodes. Hence, the partial observability with one degree is a topological scenario with robust results. We make use of discrete as well as nonlinear algorithms to strictly show the robustness of our model to detect an appropriate PMUs number. The PMU number and its related solution set at the DoOU can be easily extended to the immediately after lower depth of unobservant regions within a power grid [17].

Lower degree of unobservability results in less accuracy of state estimation routines for the purpose of getting the real state of a power grid [2]-[5]. The robustness of our models to detect optimality is shown even for this situation with a limited PMUs number being present within a power grid.

Experimental results are produced by mathematical and evolutionary algorithms to prove their robustness from different algorithmic schemes. The constraint binary integer model is solved through relaxed problems on explored nodes [13], [45]-[52]. The relaxed problems (LP) are attained by relaxing the binary restriction on the decision variables. When the BBA finds a binary solution during the iterative process, this can be considered as a global optimum point [58]-[67].

Otherwise, the binary tree is constructed with implementations of LP sub-problems towards optimality [49]-[50]. The LP objective value is the upper bound and the root node is separated into two sub problems by suitable branching. Hence, an ILP problem can be reached and solved as a linear program, in the way that solving it with the integrality to remove means that a binary solution will be detected [49]-[50]. Those mathematical models are solved by SQP or IPMs considering a scientific standpoint without being an implementation in real situations as studies claim [9]-[10].

SCIP solves the nonlinear programming model with efficiency towards a globally optimum point. SCIP linearized each polynomial equality constraint, solved the process optimization through a construction of a branch-and-bound tree and finally delivered a globally optimal solution [58]-[60].

As we can see the primal bound is the leader in this optimization process, an equal dual bound has been evaluated that leads to zero-gap toleration [50]. This study gives an inventive model on unambiguous local optimal solutions either in either in binary or continuous domain [47].

We use branch-and-bound algorithms, primal and dual simplex, genetic algorithms and binary particle optimization to attack the combinatorial optimization problem.

Then, we transform the model into a nonlinear problem with either quadratic or linear objective function. SQP and IPMs are used to attack such a kind of optimization model [45]-[49]. Two case studies are presented, analyzed and some benchmark solutions are produced for two cases namely full condition observability and the idea of incomplete observability with degree one.

As we expected, the PMUs number needed for this condition is lesser than those needed to serve the complete observability. This optimal solution serves the condition where an unobservable bus is adjacent to a neighborhood of observable buses by PMUs installed at selected power network nodes.

8. Discussion about the research work with adequate concluding remarks

This paper introduces programming algorithms for recognizing locations for phasor measurement units (PMUs) within a power network under the concept of complete and incomplete observability.

The full observability is accomplished by sensors capable of synchronization through the wide-area monitoring system. Incomplete observability is a basic context because it delivers a systematic approach of installation of synchronized sensors, arranging gradually, to such an extent the parts of an electric graph not being monitored are decreased until the power network will be completely monitored.

Examining the depth of unobservability with degree one, which is a measurement of the distance that separates an unobservant network node from a neighborhood that contains observable buses, is the focus of this research. Our approach to tackling the BIP model involves adopting an evolutionary perspective, followed by transforming the model into a nonlinear program. The nonlinear problem is solved by nonlinear algorithms or a innovative BBA scheme.

The algorithms proposed In this work are well-analyzed on the observability topic and deliver well-accepted optimal solutions for complete and incomplete observability with one degree. Practically the installation of a PMU at a bus or not fully depends on the system requirements, substation availability and the whole budget for massive installation of PMUs in a power grid. Mainly, In this work, depth of unobservability with level one is considered for the OPP problem implementation. The methodology makes use of an edge-vertex incidence matrix representing a graph constituted by branches and nodes.

Also, a square matrix is built based on the dimension of graph' vertices. The methodology of the PMU localization problem based on partial observability is a principal contribution because it gives a systematic way of placing PMUs within a power network, in stages if it is necessary.

In this work, graph theory modeling is employed to examine and find the suitable number and placement of PMUs concerning the DoOU concept. Mathematical, derivative-free, and evolutionary algorithms are utilized to analyze the impact of each approach on achieving convergence to a global solution point.

The mathematical programming model was considered solely synchronized measurements. The algorithmic model is tested on the classical IEEE power systems. MATPOWER gives the line-data connections between the power network nodes to get the edge-vertex and square matrix incidence matrix. Multiplying these matrices, a product matrix is derived.

Based on this matrix, a constraint $0-1$ integer linear program is built. Using an integer solver, the optimum point is derived. Thus, it can be characterized as a global one given that the zero-gap is found and the iterative process is terminated within a satisfied relative gap. The algorithmic model being presented In this work is tested on the classical IEEE power systems.

Hence, the unobservant regions are gradually decreased until the entire power network is fully observable. All the algorithms involving mathematical as well as population-based algorithms solve exactly the partial observability leading to optimal points spending reasonable function evaluations in reasonable runtime.

It calculates a PMU placement set where a minimum distance linear measure between one unobservable bus and its adjacent observable power nodes is a permit situation. The observation of the power grid operation can be evaluated by the system operator, which helps a lot to give an opinion about the current power network state.

The essential idea of this study is a system of placing a limited PMUs number within a power grid. This procedure is implemented in stages if it is required, so regions not being unobservant little by little decrease up to a point in time where the whole power network is fully observable.

Developing mathematical and heuristic algorithms that yield results for the absence of observability with depth-one based on collecting synchronized measurements. Those measurements are gathered by a limited number of PMU devices posed at selected power network nodes. Then, feedback is given in linear estimation routines for the unobservant power regions.

Even so, programming models have been constructed in MATLAB using efficient nonlinear algorithms embedded in optimizer functions included in the MATLAB optimization library or Opti-toolbox. For the purpose of testing the validity of such kinds of computer programming models, the binary -integer-linear program is also solved by the Branch-and-Bound algorithm (BBA).

BBA succeeds in coming across an optimum point within a zero-gap tolerance. Hence, the optimization problem is solved under optimality conditions which lead to solving it globally. The optimization problem is studied through a nonlinear model implementedby a SQP and IPMs.

From the trial efforts of the SQP and IPMs, global convergence has been achieved, in the sense that, from an arbitrary point of view, an optimal solution point will be given. Also, IPMs are proved to be convergent algorithms since they are able to deliver an acceptable optimum point in a runtime comparable with the one elapsed by the SQP methods.

SCIP optimizer function solves the nonlinear model in getting the global optimality. SCIP solves exactly the nonlinear problem giving an optimal solution within a zero-gap tolerance which is the variation between the Primal and Dual bounds.

Additionally, DoOU is declared as a binary polynomial problem and solved by a global nonlinear branch-and-bound algorithm embedded in YALMIP. This global integer routine invokes exterior integer and nonlinear programming solvers to construct the binary tree to find a global optimum point. The NLP solver counts the upper limit whereas the ILP estimates the lower bound. Moreover, the ILP solver performs as an LP solver to construct the branch strategy that leads to a globally optimum point. The upper bound is considered to be the best optimization function value for the minimization problem. Additionally, the lower bound closes the gap. That optimum point is a global solution given that the upper and lower bounds are evaluated to be equal by the optimizer function.

Hence, a global solution point is reached within a zero-gap tolerance. This mathematical methodology allows us to escape local optimal solutions, and it can be identified as the global optimal solution points with a good enough convergence speed.

GAs and BPSOs are also implemented in solving the optimization problem. They are getting the optimality using a metaheuristic standpoint of knowledge. Each algorithm results in non-unique constraint local and global minima simultaneously. The comparative study makes visible that the efficiency of the proposed mathematical and evolutionary algorithms has been proved by the circumstances of DoOU in studying that globally.

References

[1] Phadke A G and Thorp J S 2008 Synchronized Phasor Measurements and Their Applications New York: Springer

[2] Mohamed E. El-Hawary 2020 Advances in Electric Power and Energy: Static State Estimation, Wiley-IEEE Press

[3] Kundu S, Alam M, Kumar B S R, Sankar Thakur S.S. 2023 Power Systems State Estimation Using Complex Synchronized PMU Measurements: Two Novel Non-Iterative Approaches. Arabian Journal for Science and Engineering 48:5, pages 5935-5951.

[4] X. Li, A. Scaglione and T. -H. Chang, "Optimal sensor placement for hybrid state estimation in smart grid, 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, pp. 5253-5257, 2013

[5] X. Li, A. Scaglione and T. -H. Chang, A Framework for Phasor Measurement Placement in Hybrid State Estimation Via Gauss–Newton," in IEEE Transactions on Power Systems, 29, 2, pp. 824-832, March 2014

[6] Korres G.N, 2011 An integer-arithmetic algorithm for observability analysis of systems with SCADA and PMU measurements, Electric Power Systems Research, 81, 7

[7] Korres G. N. and Manousakis N. M., State estimation and bad data processing for systems including PMU and SCADA measurements, Elect. Power Syst. Res., vol. 81, no. 7, pp. 1514-1524, Jul. 2011.

[8] Manousakis N.M, Korres G.N, Aliprantis J.N, Vavourakis G.P and Makrinas G. -C. J., A two-stage state estimator for power systems with PMU and SCADA measurements, 2013 IEEE Grenoble Conference, Grenoble, France, 2013, pp. 1-6

[9] Borkowski D, Wetula A, Kowalski J, Barczentewicz S, Nabielec J, Rogóż M, Szczygieł I Experimental Setup for Harmonic Impedance Measurement in a Real HV Power Grid. Electric Power Components and Systems 47:8, pages 733-742, 2019.

[10] Pal, A. Ayala, G. A. S. Thorp, J. S. Centeno, T, and V. A, 2016. A community-based partitioning approach for phasor measurement unit placement in large systems, Electric Power Compon. Syst., 44, 12, pp. 1314–1329

[11] Johnson T and Moger T 2020 A critical review of methods for Optimal PMU Arrangement of phasor measurement units”, Int. Trans. on Elec. Ener. Sys. 31

[12] Mohanta D.K, Murthy C, D Sinha Roy, D 2016 A Brief Review of Phasor Measurement Units as Sensors for Smart Grid, Elects. Power Components Syst., 44, 411–425

[13] Ahmed M.M, Amjad M, Qureshi M.A, Imran K, Haider K, Z.M, Khan M.O 2022 A Critical Review of State-of-the-Art Optimal PMU Arrangement Techniques”, Energies, 15, 2125

[14] Yang X S 2010 Engineering Optimization: An Introduction with Metaheuristic Applications Hoboken, NJ: John Willey and Sons

[15] Xu B and Abur A 2004 Observability analysis and measurement placement for systems with PMUs IEEE PES Pow. Sys. Conf. and Exp. New York

[16] Theodorakatos N P 2018 Optimal phasor measurement unit placement for numerical observability using a two-phase branch-and-bound algorithm Int. J. of Emer. Elec. Pow. Sys., 19, 3

[17] Gou B 2008 Generalized integer linear programming formulation for optimal PMU placement. IEEE Trans. Pow. Sys. 23

[18] Dua D, Dambhare D S, Gajbhiye R K and Soman S A 2008 Optimal multistage scheduling of PMU arrangement: An ILP approach, IEEE Trans. Pow. Del. 23

[19] Müller H and Castro C. A 2016 Genetic algorithm-based phasor measurement unit placement method considering observability and security criteria IET Gen. Trans. Dist. 10

[20] Koutsoukis N C, Manousakis N M, Georgilakis P S and Korres G N 2013 Numerical observa-bility method for optimal phasor measurement units placement using recursive Tabu search method *IET Gen. Trans. Dist.* 7

[21] Dalali M and Karegar H K 2016 Optimal PMU placement for full observability of the power network with maximum redundancy using modified binary cuckoo optimisation algorithm *IET* Gen. Trans. Dist. 10

[22] Theodorakatos N P 2019 Optimal phasor measurement unit placement for numerical observability using branch-and-bound and a binary-coded genetic algorithm Elec. Pow. Comp. and Sys. 47

[23] Babu R and Bhattacharyya B 2016 Optimal allocation of phasor measurement unit for full observability of the connected power network Int. J. of Elec. Pow. Ene. Sys. 79

[24] Rahman N H and Zobaa A. F 2017 Integrated Mutation Strategy with Modified Binary PSO Algorithm for Optimal PMUs Arrangement, IEEE Trans. on Ind. Infor. 13

[25] Maji T. K. and Acharjee P 2017 Multiple solutions of optimal PMU arrangement using exponential binary PSO algorithm for smart grid applications, IEEE Trans. Ind. Appl., 53, 3

[26] Mishra C, Jones K D, Pal A, and Centeno V 2016 Binary particle swarm optimisation-based optimal substation coverage algorithm for phasor measurement unit installations in practical systems, IET Gener. Transm. Distrib., 10, 2, pp. 1–8

[27] Theodorakatos N.P 2020 A nonlinear well-determined model for power system observability using Interior-Point methods, Measurement. of the Int. Meas. Confed 152

[28] Theodorakatos N.P, Lytras M, Babu R 2020 Towards Smart Energy Grids: A Box-Constrained Nonlinear Underdetermined Model for Power System Observability Using Recursive Quadratic Programming", Energies, 13

[29] Theodorakatos N.P, Manousakis N.M and Korres G.N 2015 A sequential quadratic programming method for contingency constrained phasor measurement unit arrangement, Int Trans Electr Energy Syst, 25, 12, pp. 3185-3211

[30] Theodorakatos N.P, Manousakis N.M and Korres G.N 2015 Optimal Placement of Phasor Measurement Units with linear and non-linear models", Elec. Pow. Comp. and Sys, 43

[31] Korres G.N, Manousakis N.M, Xygkis T.C and Löfberg J. 2015 Optimal phasor measurement unit arrangement for numerical observability in the presence of conventional measurements using semidefinite programming, IET Gener. Transm. Distrib. 9, 15, 2427-2436

[32] Theodorakatos N. P, Lytras M.D., Kantoutsis K, Moschoudis A, and Christos A. Theodoridis . Optimization-Based Optimal PMU Placement for Power State Estimation and Fault Observability, AIP Conf. Proc. *28 September 2023; 2872 (1): 120033. https://doi.org/10.1063/5.0162941*

[33] T.A Alexopoulos, N.M Manousakis and G.N Korres 2016, Fault Location Observability using Phasor Measurements Units via Semidefinite Programming, IEEE Acc. 4

[34] Nuqui, R.F, Phadke, A.G., 2005 Phasor measurement unit placement techniques for complete and incomplete observability, IEEE Trans. Power Del., 20, 4, pp. 2381–2388

[35] Dalawai P. P and Abhyankar A. R., "Placement of PMUs for complete and incomplete observability using search technique," 2013 Annual IEEE India Conference (INDICON), Mumbai, India, 2013, pp. 1-5

[36] Ghosh P. K., Complete and incomplete PMU observablity for hybrid state estimation, 2017 International Conference on Computation of Power, Energy Information and Commuincation (ICCPEIC), Melmaruvathur, India, 2017, pp. 867-870

[37] Sajan K. S. and. Tyagi B, Optimal placement of PMU with optimal branch current phasors for complete and incomplete observability, 2011 IEEE Power and Energy Society General Meeting, Detroit, MI, USA, 2011, pp. 1-5

[38] X. -C. Guo, C. -S. Liao and C. -C. Chu, "Enhanced Optimal PMU Placements With Restricted Observability Propagations," in IEEE Access, vol. 8, pp. 22515-22524, 2020,

[39] Babu, R., Bhattacharyya, B. 2018 An Approach for Optimal Placement of Phasor Measurement Unit for Power Network Observability Considering Various Contingencies, Iran J Sci Technol Trans Electr Eng 42, 161–183

[40] Kumar S, Tyagi B, Kumar V and Chohan S, Incremental PMU Placement Based on Depth of Unobservability, 2018 20th National Power Systems Conference (NPSC), Tiruchirappalli, India, 2018, pp. 1-5

[41] BHONSLE, JYOTI AND JUNGHARE, ANJALI 2016, Optimal placing of PMUs in a constrained grid: an approach, Turkish Journal of Electrical Engineering and Computer Sciences: Vol. 24: No. 5, Article 87

[42] R. Diestel 2017, Graph Theory, Berlin, Germany: Springer-Verlag, 173, pp. 59-172

[43] Narsingh Deo.: Graph Theory with applications to engineering and computer science, (Prentice -Hall Inc, 1974)
[44] Christofides, N**.,** Graph Theory. An Algorithmic Approach', (Academic Press Inc, 1975)
[45] Rao, S.S., 2009 Engineering Optimization: Theory and Practice, 4th ed., John Wiley & Sons
[46] Chinneck J W 2008 Feasibility and Infeasibility in Optimization Int. Ser. in Oper. Res. & Man. Sci
[47] Williams H P 2013 Model Building in Mathematical Programming, New York: Wiley
[48] Luenberger D. G and Ye Y. 2008 Linear and Nonlinear Programming, 3rd ed. Springer
[49] Arora J.S 2016, Introduction to Optimum Design MA, Elsevier Academic Press
[50] Karlof J K 2006 Integer Programming: Theory and Practice Boca Raton: Taylor & Francis Group LLC
[51] M. M. Mano and M. D. Ciletti, 2007 Digital Design", Pearson Prentice-Hall
[52] Williams H P 2013 Model Building in Mathematical Programming New York: Wiley
[53] N. Maratos, 1978, Exact penalty functions for finite dimensional and control optimization problems
[54] Theodorakatos N. P, Lytras M.D., Moschoudis A, Kantoutsis K. Implementation of Optimization-based Algorithms for Maximum Power System Observability using Synchronized Measurements, AIP Conf. Proc. *28 September 2023; 2872 (1): 120006. https://doi.org/10.1063/5.0162930*
[55] Theodorakatos N. P, Lytras M.D., Moschoudis A, Kantoutsis K. Research on Optimization Procedure of PMU Positioning Problem Achieving Maximum Observability Based on Heuristic Algorithms, AIP Conf. Proc. *28 September 2023; 2872 (1): 120032. https://doi.org/10.1063/5.0162934*
[56] Zimmerman R. D., Murillo-Sánchez C. E. and Thomas R. J., 2010 Mat-power: Steady-state operations planning and analysis tools for power systems research and education, IEEE Transactions on power systems, vol. 26, no. 1, pp. 12-19
[57] https://icseg.iti.illinois.edu/power-cases/
[58] Achterberg T. 2009 SCIP: Solving constraint integer programs, Math. Program. Comput. 1 (1), 1–41
[59] Vigerske S, Gleixner, A, 2017 SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework", Optim. Methods Softw
[60] Vigerske S, and Gleixner A, 2018 SCIP: Global optimization of mixed-integer nonlinear programs in a branch-and-cut framework." Optimization Methods and Software 33.3: 563-593.
[61] The MathWorks Inc. Optimization Toolbox for use with MATLAB R®. User's Guide for Mathwork. 2018. Available online: www.mathworks.com
[62] The MathWorks Inc., Global optimization toolbox for use with MATLAB®. User's Guide for Mathwork, 2018. Available online: www.mathworks.com
[63] OPTI Toolbox v2.21 A free MATLAB toolbox for optimization. [Online]. Available: https://github.com/jonathancurrie/OPTI
[64] "Gurobi optimizer reference manual", 2022, [online] Available: http://www.gurobi.com.
[65] https://www.gurobi.com/documentation/9.5/refman/finding_multiple_solutions.html
[66] Theodorakatos N.P, Babu R and Moschoudis A.P 2023 The Branch-and-Bound Algorithm in Optimizing Mathematical Programming Models to Achieve Power Grid Observability. Axioms **2023**, 12*,* 1040. https://doi.org/10.3390/axioms12111040
[67] [Online]. Available : https://www.mosek.com/
[68] Löfberg J., 2004 YALMIP: A toolbox for modeling and optimization in MATLAB", Proc. CACSD Conf., pp. 1-7, [online] Available: http://users.isy.liu.se/johanl/YALMIP/
[69] https://YALMIP.github.io/tutorial/globaloptimization/
[70] https://yalmip.github.io/multi-objective-problems
[71] https://YALMIP.github.io/solver/bmibnb/

[72] Vlachos, DS and Simos, TE 2004, Partitioned Linear Multistep Method for Long Term Integration of the N-Body Problem, Applied Numerical Analysis and Computational Mathematics, 2, 540-546

[73] Vlachos, DS and Xenoulis, AC 1998, Gas detection sensitivity and cluster size, Nanostructured Materials, 10 (8), 1355-1361

[74] Nasiopoulos DK, Sakas, DP and Vlachos, DS 2014, Modeling publications in academic conferences, Procedia-social and behavioral sciences, 147, 467-477

[75] Nasiopoulos DK, Sakas, DP and Vlachos, DS 2013, The contribution of dynamic simulation model of depiction of knowledge, in the leading process of high technology companies, Key Engineering Materials, 543, 406-409

[76] Nasiopoulos DK, Sakas, DP, Vlachos, DS and Mavrogianni, AN 2015, Simulation of generation of new ideas for new product development and IT services, AIP conference proceedings, 1§633 (1), 60-68

[77] Nasiopoulos DK, Sakas, DP, Vlachos, DS and Mavrogianni, AN 2015, Modeling of market segmentation for new IT product development, AIP conference proceedings, 1§633 (1), 51-59

[78] Sakas, DP, Vlachos, DS and Nasiopoulos DK, 2016, Modeling the development of the online conference's services, Library review, 65 (3), 160-184