Journal of Physics: Conference Series

Calculating Global Minimum Points to Binary Polynomial Optimization Problem: Optimizing the Optimal PMU Localization Problem as a Case-Study

To cite this article: Nikolaos P. Theodorakatos *et al* 2024 *J. Phys.: Conf. Ser.* **2701** 012001

View the article online for updates and enhancements.

Calculating Global Minimum Points to Binary Polynomial Optimization Problem: Optimizing the Optimal PMU Localization Problem as a Case-Study

Nikolaos P. Theodorakatos [1*], Angelos P. Moschoudis[1,2] and Rohit Babu[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, University of West Attica, 122 44 Athens, Greece
3 Department of Electrical and Electronics Engineering Alliance University, Anekal, Bengaluru, 562 106, India
*Email: nikos.theo2772002@gmail.com or nikos277@central.ntua.gr

Abstract: State estimation (SE) is an algorithmic function of an energy management system (EMS). SE provides an actual-time monitoring and control of modern electrical power grids. State Estimation can be worked with sufficiency using Phasor Measurement Units optimally placed within a power grid. This paper concerns the implementation of proper algorithms embedded in optimization solvers to the optimal PMU localization problem solving globally.The optimization model is formulated as a $0-1$ nonlinear minimization problem. The problem is transformed to a polyhedron using linearization methods and B&B tree. In this model, we use a linear cost function under polynomial constraints and binary restrictions on the design variables in a symbolic format. This mathematical model is programmed in the YALMIP environment which is fully compatible with MATLAB. The $0-1$ Nonlinear Programming (NLP) model is suitable for getting concisely global optimal solutions. The optimal solution is given by a wrapped optimization engine including a local optimizer routine performing together with a mixed-Integer-Linear Programming routine. The solution is achieved within a zero-gap precisely encountered during the iterative process. This tolerance criterion is a necessity for a successful implementation of the B&B tree because it ensures global optimality with an acceptance relative gap. The minimization model is implemented in a YALMIP code fully compatible with MATLAB in two stages. Initially, an objective function with one term is minimized to discover a number of sensors for wide-area monitoring, control and state estimator applications. Then, an extra product is considered in the objective to suffice maximum reliability for observing the network buses. The numerical minimization models are applied to standard power networks in the direction to be solved globally.

1. Introduction

The essential modifications in the electric power industry demand changes in the operational security and reliability for the purpose of continuous power supply to the customers. This happens due to increasingly gradual demand for electricity. Knowing exactly the state estimator vector, the power control center is able to accomplish this demand using Remote Terminal Units (RTUs) [1]-[3].

Currently, the traditional power grids are transformed into Smart Grids using Phasor Measurement Units (PMUs) to accomplish the appropriate measurements to deliver an optimal input for state estimation, control and power stability applications. Phasor Measurement Unit is a synchrophasor device that is utilized by the utility industry to obtain measurements related to voltage magnitude and phase angle [1]-[3]. This measurement device is able to achieve such a kind of measurement with the aid of Global Positioning System (GPS). A GPS enables time-stamping and the data collected by PMUs are used for monitoring and control activities in a wide-area measuring framework [1]-[3].

Basically, a PMU device is a real-time measuring tool purposefully located in a Smart Grid to catch power network phasors using a frequent time origin to achieve synchronization in collecting the measurements for wide-area monitoring applications [1]-[3]. Recently PMUs are extensively posed in power grids that are transformed this way into Smart Grids. The installation of PMUs makes it able to monitor and suffice the power grid topological or numerical observability [1]-[3], [5]-[7].

Due to the high cost to purchase each PMU device depending on the number of channel capacity for every bus, a least possible PMUs number must be installed in a power system to suffice an adequate power system monitoring [5]-[7]. Observability outcome is an essential tool in analyzing state estimation in real-time. For real time state estimation analysis, there are two main algorithmic approaches, that is, topological analysis and numerical analysis [1]-[3]. The topological model is based on power network connectivity to build the binary matrix and hence, obtainable measurements on zero-injection and power flow measurements are considered [5]-[7].

Therefore, the optimal PMU localization problem is solved by mathematical and heuristic algorithms [17]-[22]. These algorithms find out the proper PMUs number for EMS purposes [1]-[3].

EMS are used to observe with precision the power network state variables that are the voltage phasor of each bus in real time. For that reason, a least PMUs number is suitable for distribution in a smart-grid needed to carry out its State Estimation [1]-[3].

When these measurements are obtainable, the power network is considered completely observable [1]-[3]. In addition, PMU being in work makes various significant utilizations for instance state estimation, power monitoring system, keeping protection methods, power control system and stability applications. The least selection of PMUs in number required for full condition of observability is from $20\%\ to\ 30\%$ of the power network nodes numbers [1]-[7].

The goal is principally to discover the appropriate PMU numbers and their positioning sites to render the condition of the power network fully observable. The PMU localization problem secures power system observability whereas a least PMUs number are posed at selected buses in a power transmission grid [5]-[7]. Hence, it is extremely important for utility industries to implement a practically efficient process optimization to detect a restricted PMUs number being available at Optimal PMU Arrangement sites around the power grid [1]-[3].

The Optimal PMU Arrangement problem belongs is a combinatorial optimization problem where the decision variables take two decision values, namely $YES\ or\ NO$-[11]. The aim is the localization of a PMU at a network bus or not [9]-[11]. With this binary logic, this optimization problem is solved by mathematical or metaheuristic algorithms compatible with its nature [9]-[11].

Therefore, a minimum PMUs number are installed in appropriate sites so as to ensure each network bus to be monitored by at least one time. Various optimization methods have been adopted to give an optimal solution related to the optimal PMU localization problem [5]-[7].

These optimization algorithmic schemes are ranked into two categories, that is, mathematical-based algorithms and heuristic algorithms which properly handle the observability constraints in the direction of optimality [5]-[7]. Several mathematical techniques have been addressed for the solution of a large number of similar technical challeges [71]-[77]. The optimal PMU problem is an intellectually demanding task [5]-[6].

It includes $0-1$ integer linear programming models as well as nonlinear programming approaches with continuous decision variables [17]-[22], [27]-[31]. The first one model gives a global solution whereas the second one discovers local optimality characterized as a global best possible solution [14]. Each algorithm being implemented for the OPP problem solving comes across iteratively an optimal solution which covers the target of complete observation of the power system.

Such an algorithm starts with one estimated point for initialization and finally meets optimality and feasibility within predefined tolerances in unison [16]. The principal programming model is the $0-1$ integer constraint optimization problem using a branch-and-bound algorithm for globally solving [17]-[22]. The pioneering work belongs to Ali Abur's topological observability model [3]. The authors use set theory to eliminate expressions in a binary program [15]-[16].

The optimal PMU localization problem is stated as a constraint binary integer linear program. The optimal localization of PMUs is declared in 0-1 ILP and NLP models [5]-[7]. The ILP model's algorithmic solutions are branch-and-bound algorithms [17]-[22].

In [23], an optimal PMU formulation is presented based on the maximum observability indicator whereas the optimal solutions are satisfied by the state estimator tool [2]-[3]. A branch-and-bound algorithm completed in two phases is utilized to attack the widely recognized and used integer constraint programming model with binary variables [21]-[22]. Many optimization solvers such as intlinprog [62], SCIP [63]-[66], Gurobi [67] are being implemented for the solution of the $0-1$ ILP model in the direction of optimality [14]-[16]. A binary semi-definite programming model is studied in [26] to solve the optimal PMU localization problem based on a numerical observability consideration. Nonlinear algorithms [24] such as interior-point methods [27] and sequential quadratic programming [28]-[30] are suitable to solve the NLP problem [8].

Those algorithms ensure global convergence to a local solution point meaning convergence to a point that satisfies the Karush-Kuhn-Tucker optimality conditions starting from any initial point [8], [24]-[25]. Nonlinear programming models have a feasible set with non-convex structure, leading in many local minimum points if the case study is the minimization [24]-[25].

Using clustering techniques, the nonlinear algorithms are able to encounter a global solution with a high leverage amount of possibility [24]-[25], [44]. Another approach to find a solution for the optimal PMU localization problem is to utilize direct search algorithms [31]. A mathematical Groebner algorithm is suggested to obtain a solution for the WLS optimization problem in [32]. Also, a number of heuristic algorithms are adopted to solve the binary (Boolean) optimization model [8].

Genetic Algorithms [33], Binary Particle Swarm Optimization [34]-[37], Binary Cuckoo Optimization Algorithm [38], Tabu Search [39], Recursive Tabu Search [40], binary gravitational search algorithm [41] and a fuzzy-based modified whale optimization algorithm [42] are adopted to hold with the $0-1$ ILP model's constraints for the purpose of getting optimality [8].

These algorithms give an optimal solution being acceptable for the optimal localization problem solving. Meanwhile, a MILP methodology is adopted in conjunction with a stochastic-based population algorithm in solving the optimal PMU localization problem in [43]. All previous studies define the maximum indicator as a desired-effective optimal solution without showing appropriate solutions. This paper continues a series of newly published articles in this domain to show that this topic is able to be solved with an objective with two criterions (minimize/maximize) [31], [44]-[45].

Generalized Pattern Search Methodology is probably to reach the best possible solutions covering one criterion objective function. This direct search algorithm is proved to be convergent to those optimum points [31]. This study examines the scalability and the ability to reach a goal such as to minimize the PMU numbers delivering a global solution at a single algorithm's run.

Then the objective function is extended with the aim to satisfy two terms, that is, to minimize the PMU cost installation while the reliability is satisfactorily in maximum amount for state estimation purposes [2]-[3]. The optimization terminates within a stopping criterion to find an optimum point.

2. Optimal Sensor Arrangement using a YALMIP branch-and-bound algorithm

The motivation of this study is to minimize the entire programming model with binary products related to the nature of the decision variables. A $0-1$ (MI) Non-Convex LP is proposed, along with perfectly suited branch-and-bound algorithms embedded in standard ILP solvers to build the enumeration tree wrapped together with a local solver, both used to calculate the upper and lower bounds giving a zero-gap tolerance [70]. With this structure, a $0-1$ nonlinear program is studied and finally optimized to deliver the true PMU numbers to suffice system observability [17]-[23].

The minimization model is stated as a $0-1$ nonlinear programming format in which the decision variables are presented in a strictly binary restriction, minimizing a linear cost function under a polynomial constraint function. Such optimization models don't always guarantee global optimality.

This paper will analyze the probable process optimization in getting global optimal solutions produced by a binary (Boolean) polynomial optimization model [11]. We propose the solution of the nonlinear program in a binary framework using the YALMIP program [59]. For simulation purposes, two optimization solvers included in appropriate optimization libraries such as MATLAB library and

YALMIP is a widespread optimization library needed to implement and optimize mathematical models jointly with the utilization of MATLAB. YALMIP toolbox holds up Bi-level programs, second order cone programs, linear programs, quadratic programs, integer programs, nonlinear models and global optimization studies to come across optimality [59], [62]-[70].

We have highlighted the participation of two optimizer routines for revisiting the optimal localization problems of PMUs for installation in a futuristic power system. The minimization model is able to highlight explicit global optimal solutions produced by the collaboration of a local NLP solver with an ILP optimizer function [70]. To implement and solve such kinds of optimization models, YALMIP interfaces commercial or open-source optimization MATLAB codes [59], [62]-[70].

It consists of a linear cost function that is optimized under a polynomial constraint function whereas the design variables are declared as binary using YALMIP [59], [69]-[70]. For that reason, we present some standard numerical experiments and their corresponding numerical results to analyze the impact of this mathematical model related to the determination of a global solution.

This consists of an innovation given that these optimization models don't always ensure global optimal solutions [24]-[25]. To point out our study, this optimal localization problem of PMUs relies on the following two factors to attain an optimal output under guarantee [18], [44]-[45].

A. Minimizing the PMUs number satisfying a number of topological restrictions based on Ohm's and Kirchhoff's mathematical laws that are adequately defined to fulfill a wanted level of power network grid observability.
B. Maximizing the time of monitoring each power network bus directly or indirectly observed by PMUs monitoring devices optimally installed in the power grid.

Remark A is significant to be achieved for the purpose of satisfying the topological observability as defined in previous studies, by a desired PMUs number, put them in appropriate sites around the power grid. Remark B indicates the significance of the achievement of maximum power network observability by using synchronized phasor measurements.

To accomplish that, two product terms involved in the objective function are declared to retain the network observability. Hence, the smallest PMUs in amount are counted. At the same time, the maximum reliability of monitoring the network buses is fulfilled around the modern grid equipped with PMU devices in a least number and optimally at the same time [2]-[3].

The goal of this paper is to improve the reliability of wide area monitoring by making it possible to do and improve the power system protection, power control schemes by using PMUs [2]-[3].

The aim is to decrease the economic load of the utilization of PMUs by industry utilities in installation within the power grid [44]-[45]. Therefore, we attack with efficiency the optimal PMU localization problem under fully observable conditions. Compared with the existing technology that is RTUs, PMUs can improve the degree of correctness of state estimation procedures [1]-[7].

PMU has the capability to calculate voltage phasor magnitude and angles which are state estimation variables to be computed to proceed and calculate power system observability [1]-[3].

A vast PMUs number have already been installed and much more will be installed in power grids around the world in the near future [5]. Thus, the existing power grids are transformed into modern power grids satisfying the continuous demand for energy to the customers of industry utilities with an increased amount of reliability [5].

Given that the PMUs are able to compute the phasor magnitude and angle of a network bus as well as branch current in a straightforward way, a linear state estimation model is able to deliver the power system state with a degree of correctness. This can be achieved due to high statistics rate and PMU measurements collected accurately by an optimal number of these devices around the power grid.

For mathematical programming purposes, we revisit our previous study that introduces a non-convex mathematical model fulfilling the power grid level wanted to an accurate state estimation model [27]-[31]. In this mathematical model, the maximum level of observability is achieved by using clustering techniques in getting global optimality with high leverage [44]-[45].

To attain a compromise in optimization of an objective with two statements at odds, a two-term objective is stated in a linear framework. This statement can be formed in a zero-one polynomial programming framework and it can be optimized by the sum of two terms towards optimality.

Hence to succeed in optimality, we desire to optimize two parameters (minimize/maximize) at the same time by using appropriate optimization software packages. This can be solved by transforming the maximization to an objective term with a negative sign to go to the optimization with a successful run. The objective linear function is optimized subject to a polynomial constraint within binary decision variables trying to compromise conflicting trends with accuracy. We introduce a process via which a non-convex optimization problem with binary decision variables is convexified and transformed into a polyhedron model formulation. The polynomial model is approximated by using a convex polyhedral [8]-[16].

This transformation is able to be solved with the branch-and-bound algorithm embedded in a ILP solver jointly with an NLP solver. For those implementation purposes, a local nonlinear solver and an integer programming solver are used to construct the B&B tree in getting the best possible solution at a root-node which is feasible and optimal in unison. The fulfillment of two optimization solvers is used jointly in a proper optimization software [59], [62]-[70].

The proposed optimization packages gives global optimality after a predefined relative gap to be satisfied within a zero-gap tolerance at the same time. The proposed model is studied and proved by the numerical results being achieved by appropriate algorithms embedded in optimization solvers.

The local nonlinear as well as the integer linear solvers are optional in the *bmibnb* routine included in the YALMIP program [59], [69]-[70]. They are essential to attaining a global optimal solution without spending a considerable amount of calculation time in MATLAB environment [62].

These solvers are jointly used to compute the upper and lower bounds, the difference of them and finally the global solution within an accepted relative gap and a zero absolute gap criteria [8]-[14].

This numerical process optimization is an intellectually demanding topic regarding the complexity of getting it in a global combinatorial optimization framework these days. The methodology is an efficient approach given that a global solution is delivered under assurance without to be necessary any comparison study with other algorithms outcomes published already for the PMU placement.

The main factor leading to success is its capability to reach an optimal global solution spending a relatively small amount of time [63]-[66]. Experimental simulations tested on IEEE power systems illustrate the greater ability, as well as the improved testing, of the proposed binary polynomial model related to standard nonlinear programs used so far in PMU process optimization.

The numerical outcome is delivered by appropriate optimization functions and the YALMIP library for power systems [59], [71]. This study gives remarks being a necessity summarizing in four stages.

1. It is a first pioneering effort to solve nonlinear programming with binary decision variable values with a systematic, an easy and a compact algorithmic scheme towards global optimality.
2. The $0-1$ polynomial problem is transformed into a polyhedral approximation; it delivers a notable solution for a polynomial programming model with binary decisions.
3. Two optimization solvers are concurrently used in MATLAB jointly with YALMIP programs trying to solve the optimization model globally.
4. Its solution can be found using NLP and ILP solvers to count the upper and lower levels of a B&B tree, enumerate the difference of them which finally goes to zero. This optimality and standard tolerance's metric ensure that the current solution is globally.

Considering the new and interesting aspect presented in this study, it is most important to indicate that the optimal PMU arrangement can be solved globally by spending an affordable amount of time considering the optimization model's complexity [15].

This novel proposal is based on using polynomial models with binary decision variables [57]. The complete programming model is stated in a binary polynomial optimization framework [52]. A branch-and-bound algorithm solves the $0-1$ polynomial model towards ε-global optimality [8]-[15].

Our proposal relies on building polyhedral outer approximations of nonconvexities, giving a global optimal solution by solving a number of linear programming relaxations at a given number of explored nodes in the resulting branch-and-bound tree [8]-[14]. More details about b&b algorithm can be found in [7]-[14]. The Branch-and-Bound algorithm is summarized in the following sub-paragrapth [7]-[14].

The output is a desired outcome because the optimization functions being used together evaluate and encounter the cost function at a global minimum objective value [70].

Theory of Branch-and-Bound Algorithm

In this method, the optimization problem is minimized with continuous variables, and the integer variables are relaxed. If a solution point is found to be integer, the BBA is fulfilled as it shows the return output of the integer program. If one of the integer variables is continuous, then one must be solved with two supplementary with the upper bound restriction [8]-[15]:

$$xk \leq [xk] \tag{1}$$

and lower bound constraint:

$$xk \geq [xk] + 1 \tag{2}$$

This optimization procedure of the branching strategy suffices that none of the integer solutions are not being rid of the process. The optimization procedure continues until an integer optimum point has been found. For further branching, if any of the branches present a value of the cost function larger than this upper bound price then the root-node is eliminated [8]-[15].

If a lower value of the cost function is reached than the upper bound value, afterwards the upper bound is replaced. The iterative process goes up to branching until all the root-nodes have been found a value or fathomed. The objective function has the lowest price and corresponds to the integer feasible solution. Hence, the upper bound gives the objective function value [8]-[15].

The optimization process continues to branching procedure until the BBA tree nodes to be evaluated or fathomed [8]-[15]. This solution is considered to be the upper bound of the cost function for the minimization problem [8]-[15].

3. Statement in the Direction of Smart Grids

The major concern of industrial utilities is to learn about the proper PMUs number to be posed in an existing power grid and try to transform it into a Smart Grid based on synchronized measurements.

The principal target is to minimize the PMU in numbers and to achieve a considerable number of times for which a power network node is observed for power network observability and reliability concurrently. Therefore, mathematical programming models are necessary to be solved for the optimal PMU localization problem in the direction of the determination of a global optimal solution.

This study overly solves a novel polynomial model giving the exact PMUs in number which satisfies the linear state estimator for complete observability and maximum network reliability for futuristic power grids. The proposed nonlinear model is easy and straightforward to learn about the optimum point, and thus a global optimality certificate is accomplished.

The proposed mathematical programming approach is adopted to solve a polynomial model with binary variables within an optimizing package fully compatible with MATLAB [62]. With a considerable maximum observability indicator being achieved, each network bus is observed directly or indirectly with the maximum number of times by synchronized measurements. The entire optimization process is terminated with a successful output, where a global solution is given during the B&B tree implementation on the output of the YALMIP model in MATLAB [59], [62].

The programming model is considered to be easy and straightforward in the implementation using a suitable YALMIP programming code completely compatible with the MATLAB environment. Two optimizer tools have been implemented for the proposed $0-1$ polynomial program with the non-convex structure. Hence, the entire procedure in optimization allows us to use the algorithmic scheme to escape local optimality, and it can succeed convergence to a global optimal point even though the minimization model involves more than 100 (>100) decision variables [15].

The entire process is executed by the two optimization solvers put in the same optimization package. The algorithmic scheme being solved by a suitable optimizer routine ends up when the optimality gap goes to zero and a relative gap is satisfied within a predefined tolerance [66]-[70]. This fact means that the two optimizer functions wrapped in the same optimization package perform excellently related to convergence speed, elapsed time and exploring a reasonable number of nodes in the binary tree to reach a global optimal solution within acceptable gap's tolerances [16].

4. Statement of the Optimization Problem

The optimal PMU localization process optimization is to decide the least PMUs number, their corresponding locations and to set in advance the maximum number of times by which a power network bus is monitored either directly or indirectly by the resulting arrangement [39], [44]-[45].

The first effort was succeeded by the authors using evolutionary and nonlinear algorithms with the uncertainty of global optimality in [44]-[45]. This uncertainty is limited by using comparison study with BBA's metrics for evolutionary algorithms [45] or using clustering methods for nonlinear algorithms [44]. Our target is to reformulate a constraint optimization problem by which a joined act of selecting or devoting choices can be done to attain a maximum return on complete observability and maximum level of observability [39].

The nonlinear programming model is declared as an optimization model consisting of a linear objective function under a non-convex constraint function structure that is known to be difficult to be solved with a global optimum point under sufficient precision [8]-[14]. Hence, a mathematical model is stated where the objective function is linear under a polynomial constraint whereas the decision variables are needed to take only binary values $\{0,1\}$ [55]-[58].

Despite this general truth, this study presents such optimization problems being implemented in the YALMIP library [59]. The whole optimization framework interfaces suitable ILP and NLP solvers obtainable in the MATLAB optimization library, optimization commercial or open-sources optimization packages [62]-[70]. This programming model attacks this kind of the $0-1$ nonlinear model, overcoming the difficulty of the determination of global solutions [59], [69]-[70].

This paper studies a nonlinear programming model with binary values, which adopts two optimizer functions to attain a rapid convergence and avoid too soon convergence to a local optimum point.

Then we can solve this programming model with the built-in solver named bmibnb in the YALMIP toolbox to deliver explicit optimally global solutions [59]. YALMIP extends the previous study on a non-convex programming formulation by adding a suitable symbolic syntax to activate binary decision variables in a trial to minimize the $0-1$ nonlinear model [70].

These solutions are benchmark global points for wide-area applications and power system monitoring. Explicit solutions are given by using bmibnb interfacing external solvers to count the high and lower levels in the binary tree construction. The bmibnb develops the binary tree with branching or pruning infeasible areas. The routine terminates the entire recursive process at a given root-node.

The objective function has been found to be the absolute minimum within a zero-gap tolerance. Hence, the desired optimal solution is achieved [8]-[14], [56]-[58], [59], [62]-[70].

The optimization procedure can be considered as a recursive plan to achieve the variation between the high bounds calculated so far, which the objective relative worth of the incumbent solution is at present, and the current lower bound. Due to running calculations of sub-problems, this upper bound of the cost function is achieved at the root-node while the lower bound closes the gap to zero to ensure optimality [8]-[14], [56]-[58], [59], [62]-[70].

This strategy is done repeatedly counting the variation between these levels to reach the zero value of the optimality gap. Hence, a robust optimization problem is studied to calculate the true global optimum point. On that occasion, the optimizer routine chooses to terminate the iterative process whereas a global minimum point has been found.

5. Motivation of this Study

The principal target is to introduce a novel and at the same time a methodology in optimization to produce results which are strongly global optimum points. This process optimization in decision making process is the subject of tuning a procedure to optimize an objective function involving two terms in competition without violating the constraint function at the final root-node [16].

The optimization problem is minimized in two stages. This optimization function includes two statements at odds. Therefore, the two terms objective function are stated as a scalarizing function [48]-[51]. The critical point is to get an optimal solution by optimizing a single-cost function [48]-[51]. In the first step in a process, we minimize the PMUs number and an optimal vector dependent by size of power network is achieved. In the second stage, we optimize a two-product objective function. The objective function includes two competitive terms (minimize/maximize) under restrictions. It is transformed into a single-cost function. Then, this function is minimized using a BBA implementation [8]-[16]. The decision maker problem is stated in the YALMIP optimization toolbox where a $0-1$ nonlinear programming is proposed to discover optimality [52], [68]-[70].

To succeed the optimization, the programming model is built in the YALMIP platform where the decision variables are declared in symbolic binary format [59], [69]-[70]. The context is that we present polynomial constraints that should be defined for a binary restriction of the decision variables.

The strategy is to transform the initial optimization model into a polyhedral approximation and then to examine the non-convexity constraint function and binary decision variables to find a solution.

The proposed model can be solved through the branch-and-bound algorithm embedded in the YALMIP optimization toolbox [59]. This can be implemented by using an ILP solver jointly with an NLP solver, delivering global optimality without to be a necessity a comparison study with other optimality metrics found by previous studies. Therefore, this process optimization arrives at a global optimum point on the output of solving the $0-1$ nonlinear polynomial model [59]-[60], [62]-[70].

The proposed optimization software develops a B&B tree process to reach the best incumbent and feasible solution trying to reduce the dimension of its [70]. The B&B tree's size is keeping as small as possible to find the first feasible current solution and then to the global solution [8]-[14].

6. Contribution of this Study

The OPP problem belongs to a combinatorial optimization problem with a binary logic related to the nature of the decision variable. The problem has an affect on the location of a PMU at a system node or not for adequate power network monitoring. The optimal PMU localization problem is an essential tool for a successful run of the state-estimation model [2]-[3].

This process considers all network buses phasor voltages to be computed by Smart Grid devices [1]-[7]. This can be done using phasor measurement units optimally posed at selected power network buses. Its solution lies in this area to accomplish a minimum PMUs number whereas a maximum reliability amount is accomplished for a successful run of the State Estimator (SE) Processor [5]-[7].

The attempt is to solve the optimal PMU problem and a methodology not wasteful is implemented to find an objective-method producing results related to install a PMU at power system or not using a binary (Boolean) model with logic construction in the observability constraints. Since the solution covers the least number of synchrophasors sensors with maximum indicator remains a hot topic issue.

Hence, this study covers up the lack of relative studies from the side of finding a globally optimal solution. Pioneering works have already discussed the optimization task which is to properly minimize PMU in numbers without giving a sufficient answer about solving it globally or found the upper limit measurement quantity of redundancy required [5]-[7].

Given that the PMU price remains with a price excessively high [5], it is essential to propose an optimal PMU arrangement in an alternative framework than 0-1 integer programming being solved globally. Up to now $0-1$ integer and non-convex programming models are adopted for the optimal PMU localization problem solving [5]-[7]. Non-convex optimization methods can be handled and solved locally. The local optimization methodologies are relatively fast, but they require gradient-based information and descent directions to accomplish a local solution but not the global one [8]-[11].

Even if they are able to detect a globally optimal solution, they cannot confirm it by a mathematical rule [8], [24]-[25]. Branch-and-Bound algorithms can be adopted to solve non-convex optimization, counting lower and upper bounds, giving a gap-zero to demonstrate that a global solution can be found [24]-[25], [58]. In this work, we study a nonlinear program with binary decision variables for solving the optimal PMU placement problem globally. We extend a nonlinear programming approach with continuous variables presented in past studies [27]-[31] into a $0-1$ polynomial model. Beyond that nonlinear approaches with continuous variables detect the desired outcome; our proposal is to strictly declare the decision variables as binary to deliver a global optimum point [62]-[70].

This study is taking into account a global branch-and-bound algorithm running in the YALMIP environment to implement a combinatorial decision-maker and problem solving [59], [63], [69]-[70].

To the best of a fact, two cost terms are used; either as a single term cost function or a two-objective criteria function in an $0-1$ NLP framework for the optimal PMU localization problem. This optimization problem involves the cost and the constraints which function twice differentiable with binary decision variables over the whole search [8]. The initial model is approximated by a polyhedron and is solved by a BBA based-solver in conjunction with an NLP solver interfacing in the main optimizer routine included in the YALMIP environment performing in MATLAB [60], [62].

Our aim is to show that our program runs in MATLAB programming with suitable YALMIP command syntax to attack the optimization problem on a global optimum point [59], [62], [69]-[70].

We introduce an optimization package named $''bmibnb''$ which is a built-in solver in the YALMIP program to the optimal PMU allocation problem solving [59]. BMIBNB utilizes commercial and open-source optimizer routines to figure out the solution within a 0.00 % [59]-[70]

The BMIBNB solver can solve the binary polynomial optimization problem by calling two sub-routines that effectively calculate the lower and upper bounds related to the optimal objective value on a feasible region [70]. A B&B tree process is built and terminates at a root-node with an optimality certificate [14]. Therefore, an optimal solution is achieved in well accepted calculation time.

This optimizer function is able to interface outer approximations including ILP and NLP solvers to deliver a cost-efficient optimal solution. With the uncertainties in getting a global optimal solution related to non-convex formulations, we address this task in a try for the purpose of getting a cure connected with the global optimality [68]-[70]. The foremost idea is to attempt to learn about an efficient solution methodology related to the problem-solving summarizing at the following stages.

1. Our innovation is that this is the first time where a $0-1$ NLP problem is formulated and solved in the direction of global optimality.
2. The nature of the proposed model is a non-convex problem with binary decision variables not always giving an absolute optimal solution from the globalized angle from which is seen.
3. A YALMIP global branch & bound is suggested and can be tested on an optimization problem related to the optimal PMU arrangement task.
4. The iterative process results in a global minimum solution point established by using local and global solvers with a good enough degree of correctness.
5. The globally optimal solution is guaranteed within a zero-gap tolerance at specific calculated root nodes in the implementation of the BBA' tree.

The combination of binary decision variables, a linear cost function and a non-convexity constraint function is NP-hard model in theory but we prove through the simulations that it is easy and straightforward in the implementation. Hence, global optimality is discovered in reasonable running time. Then, the solution algorithm is followed within a s-BBA framework embedded in the SCIP optimizer routine and ensures global optimality within a zero-gap tolerance [63]-[65].

Hence, an optimal solution being is derived by an outer polyhedral approximation. It delivers a solution that is also global for the original binary (Boolean) polynomial optimization model. SCIP optimizer function utilizes an NLP solver which is the IPOPT solver to solve the binary polynomial model [63]-[65]. The optimization framework is summarized as follows [63]-[65]:

1. The optimization model is transformed into a polyhedron and is solved by the SCIP optimizer function which builds a branch-and-bound tree where the Primal and Dual Bounds are calculated to be equal at the given root-node.
2. The optimization is terminated with accuracy because the difference of these bounds gives a zero-tolerance gap. This tolerance achieved means that a global optimal solution is attained. The cost value is considered to be the upper bound when the relative gap tolerance is calculated to be close to 0.00% at the explored root-node of the B&B implementation tree.

This procedure is achieved considering a piece of algorithmic data calculated during the construction of the binary tree within the iterative process. The iterative process is fulfilled when pre-defined stopping and tolerance criteria are met in unison [16]. The log file certificates the global solution being achieved. The whole optimization can be characterized as robust since an optimal solution is attained; also, a global one under strong optimality conditions [8]-[14]. The absolute gap is found to be closed [70].

We implement a minimization problem using a branch-and-bound algorithm. The upper bound is its solution whereas the lower bound is responsible for closing the absolute gap in order to give a strong condition for optimality [70].

Those optimality conditions are satisfied within a zero-gap tolerance and a meaningless percentage relative gap. Therefore, we propose a novel optimization approach to break free of local minima, avoid being trapped into them and to produce a solution within a 0.00 % optimality [14]-[16].

7. Novelty and Assessment Criteria to Achieve Optimality

We study a combinatorial optimization problem related to some binary (Boolean) optimization model with application in the field of synchronized measurements for monitoring the smart grids. The innovation of this study is proved in practice because a polynomial problem with continuous variables is able to attain locally optimal solutions [8]-[12], [27]-[30].

The motivation of this study is to find a solution for the binary (Boolean) minimization problem and notice an optimal point within a zero-gap tolerance; hence to be characterized as a global one [16]. Two targets are aimed at optimizing and they are presented to be solved globally [63]-[65].

1. To properly minimize an objective function involving one term being in a linear form under topological observability constraints while the binary restriction is satisfied on the design variables.
2. To achieve a sufficient optimal and a global solution for a two-product objective function being optimized where these terms perform competitively during the optimization process.

Both algorithmic models are implemented in a strict binary constraint regarding the nature of the arrangement variables. Each mathematical model delivers the desired outcome which is a strong minimum point. This point reflects the PMU required in a suitable number to validate the power system observability. Hence, we bring into clear view the BMIBNB optimizer built-in the YALMIP library. This optimizer function is tested on a binary (Boolean) polynomial problem [70].

In this study, we solve the polynomial problem with binary variables in symbolic format in the YALMIP [59]. The YALMIP global ILP solver invokes external nonlinear and integer solvers and results in a global optimal and feasible solution in unison [70]. The polynomial nonlinear problem is transformed into a polyhedral approximation. Then, a B&B tree is implemented and results in a globally solution; just the same for the initial model [14]-[16].

This is achieved through the solution of the polyhedral approximation by a s-BBA interfacing external general-purpose solvers to derive a globally optimal solution point [8]-[16]. Hence, this study overcomes the uncertainties in getting a global solution of a nonlinear program [8], [24]-[25].

There are three principal optimality metrics adopted in this study, the feasibility, the optimality and the uniqueness or not of a global solution [15]-[16]. The fundamental element in solving the proposed minimization problem is firstly to define if we can find a feasible solution through the iterative process before any optimal minimum point can be detected [8], [16], [24]-[25].

The proposed algorithmic scheme firstly seeks out the best integral solution which is a feasible one, soon after upper solver and lower routines calculations, LP-based domain reduction, and a significant amount of heuristic calculations in getting optimality [8]-[16]. The process in optimization ends when the upper and lower bounds in the B&B tree are totally equal in quantity, so as the solution to be within a 0.00 % optimality. The upper bound is considered to be the solution [70].

8. Preliminary Noteworthy Remarks

The PMU localization problem is presented in this study with the aim to accomplish the global optimality condition to a binary polynomial optimization model. The mathematical model consists of a linear cost function with a polynomial equality constraint function in a binary restriction related to the nature of the decision variables. This study has a major concern of the utilization of a multi-criteria optimization model consisting of two stages:

I. The first stage calculates the adequate PMU in numbers for full condition of observability
II. The second stage calculate those optimal solutions with maximum observability indicator

The major concern is to solve the multi-criteria polynomial optimization problem with binary variables have been ranked accordingly to relating to a b&b approach: (a) transformation of the polynomial model into a polyhedral approximation, (b) solved by global b&b algorithm (c) suitable external nonlinear and integer solvers are interfacing in the main b&b optimizer routine, and (d) a set of optimal solutions are derived satisfying the multi-criteria optimization model [15].

In addition to the above, this work solves the multi-criteria constraint problem into the following types: (a) overall optimization framework consists of a zero-one constraint integer and binary polynomial models (b) classifying suitable b&b and nonlinear solvers, (c) approximation of the binary polynomial model by a polyhedral model, (d) delivering a resulting globally optimal solutions, and (f) the presentation of solutions satisfying the two optimization models either with one criterion or two competitive trends in the objective function [15].

Allow us to draw the optimization framework slated with two optimization models. In this study, we introduce a robust binary (Boolean) optimization model, which is appropriate to a constraint optimization problem with a linear cost function under a non-convex constraint function following the classical 0-1 integer linear program [14]. The numerical result relies on a global branch-and-bound algorithmic function calling external nonlinear as well as integer linear solvers [62], [66]-[70].

We study its robustness to figure out the globally optimal solution reflecting the appropriate number of synchrophasors sensors. This ILP solver relies on a spatial B&B methodology adopting convex envelope approximation procedures for nonlinear programs. External routines are used such as nonlinear and integer solvers to come across the best possible and feasible solution [70].

This branch-and-bound global optimization algorithm requires sub-routines for delivering feasible solutions. YALMIP global b&b process invokes NLP and ILP solvers for computing upper and lower levers of the objective value and an LP solver to solve the relaxation problems. Hence, a global optimum point is detected under warranty [70].

This $0-1$ polynomial model is solved by using a global branch-and-bound (BB) algorithm. The principal ideal is based on a global BBA solver which efficiently invokes two sub-routines to calculate a lower and an upper lever on the optimal objective value [70]. This model is defined on a set consisting of the linear objective function under $0-1$ polynomial constraints within a binary decision investment integrality. BBA plays a crucial role solve the non-convex model given that they relax the polynomial constraint functions, the integrality of the decision variables and build a binary tree in getting optimality.Two optimization procedures are utilized towards optimality [14]-[16]:

- an upper bound can be detected by a local optimization methodology
- a lower bound can be delivered by solving convex relaxations, branching strategies

The first optimizer routine counts the upper bound and the second optimizer routine calculates the lower bound in the B&B tree implementation. The relaxed problems are solved by invoking an LP solver [14]. To reach an optimal solution, we adjust three separate optimization in the BMIBNB routine whereas the branching strategy is developed [70].

We use a local nonlinear routine to calculate the upper bounds by using the command '$bmibnb.uppersolver$' and an integer solver to compute the lower bounds of the objective value by calling the command '$bmibnb.lowersolver$'. Also, the relaxed problems and bound tightening issues are resolved by an LP solver with the command '$bmibnb.lpsolver$' [67]-[70].

A reasonable calculation time is spent solving linear programs based on bound propagation. BMIBNB performs this procedure for such optimization problems [67]-[70]. Hence, we turn it off '$bmibnb.lpreduce$' to 0 [70]**.** The Boolean polynomial problem is solved globally with a pre-defined stopping criteria to be met [16].

The base of such a building can be found in the technology advanced optimizer routines such a SCIP, Intlinprog, Gurobi, GLPK and LPSOLVE optimizer routines with an NLP solver such as FMINCON or IPOPT optimizer function. Finally, an optimal solution is achieved; a global one [14].

9. Binary Linear and a Polynomial Programming Models

An algorithmic scheme is proposed to attain a power observability and electrical measurement amount of redundancy concurrently. Its solution lies in this area by detecting a correct PMUs number reflecting the minimization of the objective function whereas a maximum reliability amount of measuring terms is accomplished for a successful run of a State Estimator (SE) tool [2]-[3].

This combinatorial mathematical model is characterized as a $YES\ or\ NO$ mathematical model which is stated in process optimization as a decision-making and combinatorial problem. Such combinatorial problems are characterized as an optimization problem and decision maker being studied and solved together [8]-[15].

In this work, we analyze concepts such as the implementation, examination and assessment of an objective function in two stages, *Viz.*, as a cost function with one product and as two terms objective function which are minimized in a process optimization. Uncertainties in finding a global solution are unavoidable. A suitable model is executed in an appropriate syntax and must be considered an optimization package. This can be true by involving a one term cost as well as a multi-objective function if trustworthy optimum points must be delivered [15]-[16].

To this effect, we relate an optimization package implemented in the YALMIP in conjunction with MATLAB programming platforms that permits a strategy to solve the problem globally [59], [70].

The utilization of this optimization package assists to escape from local minima points. The a binary polynomial optimization is stated to solve the OPP problem to get a global solution.

Binary polynomial optimization is a numerical optimization task concerning multivariate polynomial functions defined on a search space involving binary-valued variables. The motivation of this study is that we propose a binary polynomial problem that leads to an optimal solution. This optimum point is a global one after the utilization of technology advanced optimizer functions being grouped into ILP and nonlinear programming (NLP) solvers [14]-[15], [62]-[67].

A polynomial optimization model can be shown as a non-convex optimization model which is solved by a branch-and-bound algorithm (BBA). This programming model has a linear cost function, polynomial constraints and binary decision variables [8]-[11], [24], [68]-[70].

To construct the B&B process, an ILP solver is invoked as a lower solver to solve the relaxed problems for branching implementation [14]. The integer linear programming (ILP) solver computes the lower bound related to the objective function value at a given root-node. Meantime, an LP solver is invoked that makes the bound stronger for solving the relaxed problems [14]-[15], [62]-[67].

An NLP solver is invoked as an upper routine to calculate the upper bounds. Hence, the upper bound is estimated by invoking the local solver [70]. A calculation time is spent to follow the length of the branching process to build the b&b tree [8]-[11], [62], [66].

As a consequence of the above used development, the upper bound is a solution point produced at the first root-node of the B&B tree whereas the lower bound closes the gap. All this process optimization is considered as a valuable performance to give the global solution [8]-[16].

10. 0-1 Constraint Integer Linear Programming Problem

Two observability scenarios are examined in the recent bibliography [17], [26]. To evaluate the power network observability, a topological and a numerical model is utilized to successfully execute the state estimation tool based on traditional and synchrophasors measurements [17], [26]. A PMU calculates the voltage phasor at a bus as well as the current phasors emanating from that bus [1]. This measurement is considered to be a straight measurement. Using the knowledge of the voltage phasor at one end point of the line and its current phasor, we proceed to calculate the voltage phasors at the other adjacent nodes through Ohm's rule calculations. Knowing the phasor voltages between two network nodes, we already know the current flow on the line connecting these nodes using Ohm's rule.

If a PMU is posed at every power network bus, the phasor voltage can be calculated by PMU's node and therefore positive feedback is given to the linear state estimation.

In addition, it is not possible to install PMU at each node for realistic power transmission systems [5]-[7]. This demands a communication system and fiber optics at each substation for sending the information to phasor concentrator center (PDC) to run the state estimator tools [2]. Hence, it is difficult enough for full entry of PMUs at a desired amount in Smart Grids [5]-[7].

Meaning desired amount, each network bus must be equipped with a PMU for monitoring its voltage phasor and the currents originating from that bus. Therefore, the minimization of that equipment still remains an adequate process [5]. In the optimal PMU localization problem, design variables are getting involved by choosing a number of a candidate decision's investment to solve it.

This optimization model is declared in a $go-no-go$ integer linear program with binary decision variables [9]. In this application, decision variables are defined to be zero-one values, that is, $x_i \in \{0,1\}, \forall i \in I$ [8]-[16]. With this formulation, the i_{ith} decision is accepted or refused [9], [15].

Hence, the optimal PMU localization problem is based on a binary logic interpreted as $Yes\ or\ No$ decision [9]. The binary integer program consists of a linear cost function under a number of inequality constraints which are characterized as $multiple - choice\ constraints$ [9], [15].

This term is used for these constraints due to the logical nature of them [9]. Initially, an integer linear programming framework with binary decision variables is defined for the optimal PMU implementation in Smart Grids. The objective function is minimized subject to a topological observability function over the entire number of power network buses [17]-[22]. The $0-1$ Boolean constraint integer program is stated in Esq. (3)-(5) as [17]-[22]:

$$\begin{aligned} & minJ(\vec{x}) = w^T\vec{x} \\ & s.t. \quad A\cdot\vec{x} \geq \hat{1} \\ & \qquad\quad \vec{x} \in \{0,1\}^n \end{aligned} \tag{3}$$

Where $x = \left(x_1, x_{2,}, \ldots, x_n\right)^T$ is a binary decision variable vector, whose elements are as [17]-[23]:

$$x_i = \begin{cases} 1\ if\ a\ PMU\ is\ installed\ at\ bus\ i \\ \qquad 0\ otherwise \end{cases} \tag{4}$$

$w = \left(w_1, w_{2,}, \ldots, w_n\right)_{n*1}$ w_i is the cost of the PMU installed at the bus$i$, and $\hat{1}$ is a vector whose entries are all ones. Initially we define $w_i = 1, \forall i \in I$, meaning that all PMUs have the same place in order of arrangement [17]. Based on topological analysis, the binary connectivity matrix is built based on logic theory and the corresponding operators. The elements of the binary connectivity matrix of the power system is defined as follows [17]:

$$a_{ij} = \begin{cases} 1 \quad if\ \ \mathrm{i} = j \text{ or } j \in \mathcal{A}_i \\ \quad 0\ otherwise \end{cases} \tag{5}$$

The binary connectivity matrix is based on the topology structure of a power transmission grid. The data are taken from the MATPOWER package running in MATLAB environment [61].

A. An Aggregated Revisited Formulation for Optimal PMU Model

A multi-criteria decision maker problem is studied in this paper. The study of the multi-objective criteria decision maker model is a methodical procedure which involves uncertainties related to $''\boldsymbol{yes}''$ $\boldsymbol{or}$ $''\boldsymbol{no}''$ decision for solving making problems [9].

Multi-objective mathematical programming is a procedure used to solve the optimum PMU localization problem, giving trade-off globally optimal solutions. A multi-objective optimization problem involves two competing targets, that is, the perfecting in one objective may lead to decline in quality in another objective [8].

We develop an optimization model that integrates all the mathematics parameters defined in the objective to optimize it based on the tuning optimizer function's parameters [15]. Hence, the optimal solution is to hold up the decision maker for the optimal PMU localization problem [15]-[16].

Optimal PMU localization problem is based on two types of cost standards being optimized for the optimal outcome: I) to minimize the PMU cost arrangement II) to maximize the times by which a network bus is observed by a selected PMUs number optimal around the power network either directly or indirectly. In the constraint optimization model, we have two competitive objectives, minimizing the cost of PMUs arrangement output, and maximizing the maximum amount of the redundancy of the measurements. To declare the optimal PMU localization problem to suffice two targets (I) to minimize the objective function value, (II) to reach an optimal solution with maximum observability indicator.

The main issue for a cost function in binary domain variables is to calculate the global minimum point and in a second stage to determine if or not an optimal solution covers two competitive trends in the augmented objective function. A multi-cost function is stated as follows Esq. (6)-(7) [48]-[51]:

$$minimize\{f_1(\vec{x}), f_2(\vec{x})\}\ s.t{:}\ \vec{x}\ \in \Omega\ \vec{x} \in \{0,1\}^n \tag{6}$$

Where Ω is the feasible region constituded by the constraint function and the restriction on the declaration of the decision variable of the objective function [25]. The vector of the cost function is declared using the Esq. (7) [8]-[16], [24]-[25], [48]-[51]:

$$f(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}))^T \tag{7}$$

The multi-objective problem involves more than one cost function that must be minimized or maximized. The goal is to derive a solution which is the best trade-off solution between competitive trends including in the objective function [8]-[16]. A global B&B algorithm optimizes the two-product cost function, to detect those optimal solutions satisfying the optimality metrics [14]-[16]. The final output is a PMU numbers put in appropriate places around the power network [17]-[23].

B. Scalarization Methodology for Multi-Objective Optimization

A two-production objective function is converted into a single-objective function by specifying a predefined weight factor per decision variable in the objective. A simple way to solve the multi-objective function is to scalarize this process and to return a linear combination of these terms whereas the binary restriction holds on the design variable.

Hence, the classical process to solve bi-objective integer programs is to adopt the scalarization by which the problem is stated into a single-cost function [51].

The main target is to minimize the PMU in numbers to provide fully observable conditions whereas the second one performs differently in a manner that prevents agreement with the main objective term in minimization [18], [23], [44]-[45].

This function relies on two targets, that is, to attain full condition of observability whereas a PMU set solution with maximum measurement indicator is delivered by properly optimizing the objective.

The two-cost constraint optimization problem is converted into an optimization problem consisting of a single-cost function being optimized under topological observability and certain restrictions on the arrangement variables [14]-[16].

For a multivariate objective function declared binary-valued, the branch-and-bound algorithm can determine if or not a global minimum point by the simulation run. The resulting optimal vector satisfies the observability constraint either in linear inequality or polynomial equality functions.

We transform this two-criterion objective function into a summation in which each weight factor reflects a placement variable which varies linearly for the purpose of getting the optimality. In the resulting scalarizing cost-function, a weight factor is assigned for each objective function [48]-[53].

To calculate the best measurement outcome, we declare the relative product with a negative sign as we desire to maximize it with the purpose of getting trade-off optimal solutions. Hence, an assessment mathematical model is determined as the constitution of two optimality criteria in the objective function. The integer linear program with binary bounds is formulated using the Esq. (8)-(10):

$$minf(\vec{x}) = \left(w^T - \frac{1}{n} e^T \cdot A\right) \cdot \vec{x} \tag{8}$$

$$A \cdot \vec{x} \geq \vec{1} \tag{9}$$

$$\vec{x} \in \{0,1\}^n \tag{10}$$

The aggregate cost function consists of a linear combination of two objective functions, one reflects to minimize the minimization of synchrophasor devices. The second product is desired to be in the objective to maximize the time by which a power network bus is observed within a topological observability constraint being applied [21].

Therefore, the second product is declared with a negative sign in order for the whole objective to be possible in minimization [8]-[16]. The extra product signifies that each power system node can be monitored by a PMU installed at that node and its incident buses.

The product $e^T A$ is classified by the sum of power system nodes so that the cost functions to be minimized with appropriate weights to each placement variable. Each placement variable has a weight factor analogous to its power system's connectivity node priority.

The size of the objective depends on the dimension of the power system [8]-[16], [24]-[25]. Hence, we multiply the objective with an augmented weight parameter which reflects an appropriate PMUs number to ensure topological observability with a maximum number of times by which each power network is monitored either directly or indirectly [18], [44]-[45].

Minimizing one cost function probably leads to maximizing the amount of times by which a power network bus is observed [31]; hence, it is not always possible to deliver one optimum point satisfying all competitive objective terms in unison [14]-[16], [49]-[51]. Thus, this scalar function is optimized giving trade-off optimal solutions presenting well performance to the maximum observability solution of the optimal localization problem of PMUs in a Smart Grid [18], [23], [35], [41]-[42], [44]-[45].

C. 0-1 Polynomial Optimization Problem

Non-convex nonlinear models are intellectually demanding tasks in solving them because they have inside objectives and constraints in a non-convex format [8], [24]-[25]. The solution of the general non-convex nonlinear problems is an intellectually demanding issue because such problems have an objective function subject to a constraint function, both of them in a non-convex format [24]-[25], [56]-[58]. Similar problems with integer decision variables have the same importance even though these variables are relaxed to be continuous to solve it [8]-[15].

A characteristic of nonlinear programming models is that the solution algorithms are getting trapped in a local minimum point due to the non-convexities of the models [8], [24]-[25]. This can be overcome by using clustering methods [8]-[15].

Although there is no strictly mathematical definition, only empirical rules and stochastic processes can be adopted to detect a global optimum point with high probability [44]. Based on this remark, past studies presented the optimal PMU localization problem as a nonlinear program with continuous decision variables [27]-[31]. The problem is locally solved and is able to return a globally solution with clustering methods [44].

Such problems are stated to have on/off investments that enforce an arrangement variable to be $1\ or\ 0$ meaning that a PMU is installed at a network node or not [15]. This paper follows these studies and declares a polynomial model with binary variables and solves it with appropriate global integer programming solver in the direction of global optimality [70]. This can be achieved by the constraint $0-1$ integer linear program which is the leading one for transforming into an equivalent polynomial constraint model for achieving a global optimum point for both optimization models.

An intellectually demanding task is to declare polynomial equality constraints as well as an inequality constraint function, an optimization problem arising in combinatorial optimization [8]-[16].

The optimal PMU localization problem is proposed as a discrete process in optimization where the decision variables are declared in a binary symbolic format. Its minimum solution point satisfies a set of polynomial equality constraints. Such optimization problems are imposed with $YES\ or\ NO$ decision in the declaration of the design variables [8]-[16]. The aim is to properly handle and finally solve this kind of problem to identify optimal solutions using on/off decisions [15].

Therefore, the target is to minimize a linear cost function subject to a polynomial constraint function whereas the binary restriction is satisfied [14]-[16]. Constraint optimization is declared in a binary polynomial model and it is a widespread formulation that permits us to state the optimal PMU placement in optimization.

The deployment of enforcing $0-1$ decision variables in the YALMIP symbolic syntax is a key characteristic which makes a distinction between the proposed algorithmic optimization scheme from previous nonlinear PMU arrangement approaches. A Boolean (binary) process is used to revisit the $0-1$ constraint integer program by keeping in force polynomial transformations in all inequality constraints of the ILP problem [8]-[16], [27]-[31].

Hence, this proposal declares the decision variables as binary to handle nonlinear equations to discover global optimality for solving this combinatorial optimization problem. The outcome is a high-quality global quantity solution. The binary Boolean polynomial program is declared as Esq. (9)-(11). The optimum agreement of a PMU optimally installed at a power bus can be written as Esq. (11)-(13):

$$minJ(\vec{x}) = \sum_{i=1}^{n} w_i x_i \; w_{i=1}, \forall i \in n \tag{11}$$

$$s.t. \begin{cases} f(\vec{x}) = \hat{0} \\ \vec{x} \in \{0,1\}^n \end{cases} \tag{12}$$

Where ith element determines a polynomial observability constraint for the ith bus [27]-[31], [44]:

$$f_i(\vec{x}) = (1-x_i) \cdot \prod_{j \in n_i} (1-x_j) = 0, \;\; \forall \mathrm{i} \in n \tag{13}$$

An auxiliary binary-valued variable is an extra term incorporated into the polynomial model to reflect *yes or no* decision investment [8]-[9]. Hence, such variables are binary adding to our proposed model with the aim to declare it as a pure Boolean optimization model [15].

Those auxiliary binary decision variables are stated in the declaration of the nonlinear problem and help a lot to find a solution [69]-[70]. The utilization of auxiliary binary variables is to transform a nonlinear problem with continuous variables into an equivalent binary polynomial problem. Also, the utilization of auxiliary binary variables helps a lot to keep the binary (Boolean) tree as small as possible in size [53]-[58]. Hence, a global optimal outcome is achieved as soon as possible [14].

D. An Aggregated Revisited Formulation for Optimal PMU Model

The aim of optimizing PMUs in number needed for power network monitoring is to figure out a vector of decision variables reflecting the least objective function satisfying indisputably a constraint function. Hence, the authors consider some criteria raised in competitive trends, such as minimize in parallel with maximize in an objective function with two products [15]. The resulting model permits us to figure out a trade-off optimal solution in a globally framework using the appropriate optimizer routine to succeed this purpose. Solving the twofold binary Boolean polynomial problem means that two distinct cost term must be satisfied in unison: (I) to achieve an optimal solution which maintain the topological observability and (II) to reach those optimal solution.

In this mathematical model, we declare two competitive objectives, minimizing the PMU cost installation at power network buses, and maximizing the measurement indicator [18]. Such optimization problems include a two-product objective function being optimized with a wrapped optimization package of two optimization solvers of computing the global optimality [62], [66]-[70].

To implement the two-criterion constraint optimization formulation, we transform it into a single-cost function to reach optimality [48]-[51]. We can define a linear cost function consisting of polynomial equality constraints defined on $\{0,1\}$ related to the declaration of the decision variables' nature as Esq. (14)-(15). This objective is composed of the functioning costs of selected PMU at a power network or not and an extra term resulting from maximum level of the summation of times by which each network node is monitored either directly or indirectly Esq. (14)-(15) [17].

$$\min_{x} f(\vec{x}) = \left(w^T - \frac{1}{n} \cdot e^T \cdot A\right) \cdot \vec{x} \tag{14}$$

$$s.t. \begin{cases} f_i(\vec{x}) = (1 - x_i) \cdot \prod_{j \in n}(1 - x_j) = 0, \ \forall \mathrm{i} \in n \\ \vec{x} \in \{0,1\}^n \end{cases} \tag{15}$$

To get a consideration to the first one term, the second one is declared to ensure that the maximum number being observed each network bus will be accomplished in unison. Of course, for minimization purposes, the maximization term is involved with a negative sign in the single-cost function [44]-[45].

Based on the above mentioned, in a second stage the one objective function term is expanded to include a product term related to the number of times which a network bus is monitored by a PMU installed at that bus or in a neighborhood bus. The b-objective function is optimized under the certain polynomial constraint function within a binary decision restriction [48]-[52].

Using a YALMIP global branch & bound algorithm, the polynomial problem is well optimized in the direction of getting the best optimal solution; a global one at a single run. The optimal solution is attained by spending a small number of nodes where the LP relaxations are solved and the branching process be done towards an solution with an optimality criteria [8]-[16], [24]-[25].

The optimal solution satisfies the competitive trends involved in the objective function concurrently. Thus, the optimization model's performance is proven in practice, that is, being able to attain a global solution for the optimal PMU arrangement problem and giving a globally optimal solution having the maximum value of observability information redundancy in the second stage [23].

11. Solving a Boolean Polynomial Problem

In general scope, a nonlinear algorithm such as sequential quadratic programming and interior-point methods used for such nonlinear programming explore the search space seeking out locally optimal solutions [27]-[30]. These mathematical algorithms detect local optimal solutions which result to be characterized as global solutions [8], [24]-[25]. On that occasion, the aim of detecting the true global optimal solution is satisfied using multi-start methodologies with high probability [8], [44].

To overcome this uncertainty, the non-convex optimization problem is declared as a linear cost function that is minimized consisting of a number polynomial equality constraints over decision variables declared in a binary format in the YALMIP platform [59].

The initial model is transformed into a polyhedral approximation being solved by B&B process using a built-in solver in the YALMIP and technology advanced ILP and NLP solvers. Developing the B&B tree, the solvers reach the incumbent solution which is the best feasible solution up until now to prune some areas which don't include any optimum point [70].

The model is solved globally within specific tolerance gaps which ensure the optimal solution [8]-[16]. The iterative process ends up with the best possible solution derived with a powerful integer routine in combination with a local nonlinear algorithm. The polynomial integer program involves products of binary variables declared in a YALMIP syntax command with MATLAB language [59].

The proposed mathematical PMU positioning algorithm turns out well in delivering the global best possible solution related to the best-known PMU number published so far in bibliography [5]-[7].

This best possible PMUs number is put in place around the power grid for suitable power monitoring and state estimation and power control applications [1]-[3], [5]-[7]. To fill the above-mentioned deciding factors, it is adequate to define an objective function that reflects the minimization of sensors capable of synchronization of phasor measurements augmented by a factor related to the maximization of the level of power network observability [18], [23], [44]-[45].

The principal aim is to accomplish the minimization of PMUs and put in suited locations with a purpose to satisfy the state estimation process that ensures the topologically observability [5]-[7]. To accomplish as large as possible the power network computation of times that a bus is being monitored by synchronized measurements, an extra term was added following studies in [44]-[45].

An easy preferred method is followed to attack the two-criterion objective constraint model with a B&B process. We reformulate the cost function with an augmented product multiplying the decision variable. In this mathematical model, two terms involved in the objective function perform competitively towards optimality. This mathematical model is stated in a $0-1$ NLP framework [58].

The proposed $0-1$ polynomial programming model is executed in the YALMIP program whereas each decision variable is given in a binary symbolic format [59]. The aim of this paper is to illustrate that the proposed programming model is effectively solved unambiguously using the YALMIP library in conjunction with MATLAB optimization solvers [59], [62]-[63], [69]-[70].

It aims at minimizing the PMU numbers needed to cover the power system monitoring, and maximizing the amount of times where a network bus is monitored directly or indirectly by a PMU installed at a bus around the power transmission network [18]-[19], [38]-[45]. The YALMIP's bmibnb optimizer function is utilized to solve the suggested binary polynomial model [70].

This can be considered as a novel creation because such programming models don't always ensure global optimal solutions [27]-[31], [44]-[45]. The optimization systematic actions are adequate to accomplish the global solution with accurate standard optimality measures. The global solution is interpreted in the exact PMUs number put in suitable sites around the power grid.

12. Detecting of Global Minima Points for the Optimal PMU placement Problem

We utilize branch-and-bound algorithms, widely used in mixed-integer programming solving in a binary polynomial model. A global process optimization is utilized to detect the global minimum point related to this model. The solution is based on the calculation of the upper and lower bounds of the objective function during the construction of the b&b tree [70].

Hence, an integer linear programming solver is essential for counting the convex relaxations and a nonlinear solver to solve the problem based on locality [56]. For illustration and validation purposes, we use IEEE-bus systems to simulate the binary (Boolean) optimization model for both case studies [61], [71]. In the first phase, we minimize an objective function with one-term to learn about the precise number of synchronized measuring devices to be optimally placed around the power grid.

In the second one, we optimize the augmented function for full condition of observability and maximum number of times being observed on each network bus. The initial non-convex program is transformed into a polyhedron which is solved by using an appropriate B&B algorithm [69]-[70].

Two optimizer functions are performed in conjunction for calculating the upper and lower bounds in the B&B process' calculations. A B&B tree is constructed whereas upper and lower bounds are produced by appropriate local and global solvers embedded in a global nonlinear integer programming routine.The iterative process calculates the variation between these bounds so as to establish the lowest cost value at the tree node in the direction of global solution achievement [69]-[70].

a) Full Condition of Observability Scenario

The objective function in those problems is most frequently a multi-modal cost-function having a considerable amount of local optimal points [9], [15]. Hence, the proposed problem is solved by an appropriate algorithmic scheme avoiding being trapped into the same local solution point. The local optimizer routines FMINCON and IPOPT cannot guarantee to detect a global solution for a non-convex programming problem. Hence, we utilize a reformulation strategy in the specific non-convex (MI) NLP problem. To achieve a global solution, computations must be done in the MATLAB environment using suitable YALMIP toolbox's commands [59], [62], [69]-[70].

b) Studying the OPP by Reformulating in Non-Convex (MI) NLP Model

Reformulation permits us to convert the polynomial problem into a formulation that is easier and more straightforward to be solved by the proposed BMIBNB solver [70]. This is approached by a polyhedron which accomplishes the solution by linearizing the constraints and relaxing the binary decision variables [8]-[16]. Hence, the initial model is more easily solved in a global solution.

Considering that we have already programmed the MATLAB code, we run the BMIBNB branch-and-bound global optimizer function included in the YALMIP toolbox, wrapped with a nonlinear local optimizer function. This function is an upper solver used to implement and solve sub-problems within the global ILP function. FMINCON or IPOPT being available in the MATLAB optimization library and OPTI toolbox respectively can be used for that purpose.Technology advanced ILP solvers are used to solve linear programs and lower solvers [62]-[70].

BMIBNB uses linear relaxations to solve the proposed multivariate binary polynomial model. The optimizer function relies on a s-BBA and adopts a convex envelope act of estimating, that is, approximations to manipulate the nonconvexities [62]-[70]. The relaxation problems are solved using an LP solver that is relatively connected with the nature of the problem [70].

The upper bounds of the B&B tree are detected using a local nonlinear solver such as FMINCON included in MATLAB optimization library [62] or IPOPT included in OPTI-toolbox [66]. The upper and lower bounds can be found by using the NLP and ILP solvers as [59]:

$\mathrm{bmibnb.uppersolver}$	(16)	to interface the local NLP solver
$\mathrm{bmibnb.lowersolver}$	(17)	to invoke the ILP solver
$\mathrm{bmibnb.lpsolver}$	(18)	to invoke an LP solver to tighten the bounds towards a feasible solution

BMIBNB adopts additional cuts to tighten the relaxations, and the involved time required in the lower bound optimizer routines [70]. In the iterative process, whenever BMIBNB comes across a zero-gap with cuts also made by non-convex constraint, it means that the global solution is detected [70].

An optimization is performed with efficiency because it is based on a Bound Tightening solving linear programming relaxations [8]. During the implementation of the B&B tree, propagation procedures are taken into consideration in a way to achieve a fast approximation in using the Bound

Tightening process helps a lot with the whole given strength for convex relaxations being solved as well as to decrease the variable domain [63]-[65]. This is a crucial parameter because the enumerative search is carried out on the variable domain [8]-[16]. The optimization library solves the problem with the built-in solver bmibnb interfacing obtainable routines as local and global solvers to reach a globally optimal solution within a zero-gap and an acceptable relative-gap tolerances [8]-[16].

The global solution is attained through the B&B optimization process where the upper and lower bounds are computed giving a zero-Gap (% 0.00) within a predefined relative gap tolerance. Thus, it delivers with this manner one candidate solution which is the global one in reasonably elapsed time.

The FMINCON routine is stated as the Upper Solver [62] whereas an ILP solver is utilized as Lower Solver as well as an LP Solver to get an optimum point [62], [66]-[67]. Therefore, the B&B process starts to build the binary tree, developing the branching and exploring the nodes [14]-[16]. The principal physical thing in the optimizer routine is to solve the relaxation program [14]-[16].

This one is produced when bounds are obtainable at a given root-node so that the entire framework is constructed by using outer convex envelope approximation approaches [8]-[16], [59], [63], [69]-[70]. The LP relaxations are suitably solved by the LP solver in the direction of optimality. Early heuristic tool is calculated during all of the process in optimization solving to reach the first incumbent solution which is also a feasible one to satisfy the constraint function [47], [64].

The main concept of using dividing heuristics is to incorporate in the B&B tree in getting optimality in a better and fast computational way [54]. They are used to round some fractional variable iteration and thus to revisit the linear approximation problems, succeeding a depth-first-search in the B&B tree [8]-[16]. In this case, dividing heuristics adopt a specific branching regulation which helps a lot to find out a first feasible solution. Thus, heuristic calculations perform helpfully to reach the optimal solution in a fast way [54]. This heuristic implementation is calculated at the given root-node of the B&B tree for necessary achievement for the purpose of getting the true optimality [70].

Therefore, getting an adequate feasible solution at an early stage throughout the iterative process is not only practically advantageous, but likewise assists to go faster the process optimization. A globally optimal solution is met satisfying optimality and feasibility performing in unison.The binary tree terminates its development when a feasible and best possible solution is noticed while some areas are pruned to eliminate infeasibility metrics [8]-[16], [59], [63], [69]-[70].

The main element to implement the proposed optimization model in the YALMIP is to state the design variables using the command $binvar\ with$ the following syntax. The binary nature of it is clear using a symbolic syntax in the YALMIP platform as [59], [69]-[70].

$$x = binvar\ (nbus, 1); \tag{19}$$

Hence, a linear matrix variable with size analogous to the optimization model's dimensions is declared. Therefore, all manner of things can be done related to the binary nature of the optimization model. Once the decision variable is declared in a binary framework, we define the constraints of the proposed programming model using an appropriate syntax fully compatible with the YALMIP platform. We invoke the YALMIP branch-and-bound algorithm with the command [68]-[70]:

$$ops = sdpsettings\ ('solver', 'bmibnb'); \tag{20}$$

$$\mathrm{ops} = \mathrm{sdpsettings}\ (\mathrm{ops}, ' \mathrm{verbose}', 1); \tag{21}$$

We optimize the binary (Boolean) optimization model using the YALMIP routine [59], [68]-[70]:

$$\mathrm{optimize}\ (\mathrm{Constraints}, \mathrm{Objective}, \mathrm{ops}); \tag{22}$$

c) Studying the Benchmark IEEE-14 bus System

We indicate the value of transformation of the initial model into a polyhedral approximation being solved by a B&B tree. The IEEE 14-bus system is used as the first test system to simulate our proposal in YALMIP/MATLAB [61], [71]. Starting with the YALMIP branch-and-bound algorithm, we present an optimum point within an absolute zero-gap tolerance preserving a certificate of global optimality [8]-[16]. We use the IEEE-14 bus system as a first benchmark system to prove our algorithmic scheme's accuracy. The solution is displayed in the log file shown in the Table 1.

FMINCON or IPOPT, both included in commercial or free optimization libraries, are used to produce the Upper bounds [62], [66]-[67]. The nonlinear solver gives the Upper bound to find the first integral solution [70]. On the other hand, the Lower bound is produced by a target integer programming solver. A linear programming solver is used to relax and tight the boundaries [8]-[16].

Meanwhile solving the linear approximation problems by the integer program solver, the Lower bound is produced with sufficient precision [62], [66]-[67], [70].

Starting with a global B&B algorithm, the YALMIP program identifies the equality constraints, some of them as polynomial and some as bilinear as shown in log files produced by the routine. The algorithmic strategy uses improved solutions found by heuristics calculations during the iterative process to come across a proper solution of the underlying $0-1$ NLP model [8]-[15].

Also, a linear relaxation is implemented to illustrate that this heuristic approach delivers global-optimal solutions in a reasonable timetable. Enforcing relaxations of the polynomial and bilinear constraints and binary decision restrictions lead to global optimality despite the non-convexity nature of the optimization model [70]. Powerful optimizer routines help the attempt spent on propagation getting a decision to the optimizer construction itself [14]-[16].

Afterward, the YALMIP BBA solver constructs the binary tree, explores the feasible set, prunes infeasibilities, uses heuristic calculations and delivers a solution. The lower bound is calculated by an ILP optimizer function whereas the upper bound is computed by FMINCON optimizer routine or IPOPT optimizer function behaving well as a local solver in the entire optimization [62].

The performance of optimizing the objective with one criterion is illustrated in Table 1 where the FMINCON routine is used as an upper solver whereas the SCIP solver is invoked as the lower solver [63]-[65]. The branching process ensures a globally optimal solution, since the lower bound can be valid to close the absolute gap [70].

Table 1. Optimization Process: Results of bmibnb routine

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|   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 (bilinear)     1x1|            1 to 1|
|   #9|   Equality constraint (polynomial) 1x1|            1 to 1|
|  #10|   Equality constraint (polynomial) 1x1|            1 to 1|
|  #11|   Equality constraint (polynomial) 1x1|            1 to 1|
|  #12|   Equality constraint (polynomial) 1x1|            1 to 1|
|  #13|   Equality constraint (polynomial) 1x1|            1 to 1|
|  #14|   Equality constraint (polynomial) 1x1|            1 to 1|
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Starting YALMIP global branch & bound.
* Upper solver: ipopt
* Lower solver: SCIP
* LP solver: SCIP
* -Extracting bounds from model
* -Performing root-node bound propagation
* -Calling upper solver
*****************************************************************************
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
*****************************************************************************
 (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:   4.00000E+00     0.00      4.00000E+00    2     1s Solution found by heuristics
```

Table 1. Optimization Process: Results of bmibnb routine (continued)

* Finished. Cost: 4 (lower bound: 4, relative gap 2e-09%)
* Termination with relative gap satisfied
* Timing: 13% spent in upper solver (2 problems solved)
* 7% spent in lower solver (1 problems solved)
* 54% spent in LP-based domain reduction (28 problems solved)
* 1% spent in upper heuristics (1 candidates tried)
Elapsed time is 2.525123 seconds.
ans =
0 1 0 0 0 1 0 1 1 0 0 0 0 0
ans =
2 6 8 9
Linear scalar (real, binary, 14 variables)
Current value: 4
Coefficients range: 1 to 1
SORI =
17

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 [60], [62]. On the other side, the lower bound is obtained through a convex relaxation or duality. The convex relaxations are solved through calling an external integer linear programming solver [60], [62], [66]-[67]. With this manner, the YALMIP BBA detects the global optimum point in reasonable computational time [70].

Numerical problems are calculated by the lower solver and upper solver using also a reasonable number of heuristic calculations as the simulation run illustrated in the final output [64]-[70].

YALMIP BBA solver terminates if the difference between the internally computed UPPER and LOWER bounds on the objective function is less than or equal to Absolute Gap Tolerance. The absolute gap is defined as [15]-[16], [70]:

$$Gap = \mathrm{UPPER} - \mathrm{LOWER} < \in \tag{23}$$

Where $\in$ is the default tuning parameter to return with an optimal solution under warrancy [9]. The relative gap is defined as follows [15]-[16], [70]:

$$\text{relative gap} = (\mathrm{UPPER} - \mathrm{LOWER})/(1 + |\mathrm{UPPER}|) \tag{24}$$

The algorithm terminates with a certificate producing ε-suboptimality [8]. The optimizer routine constantly considers the upper and lower bound on the objective function, the difference between those levels. A gap is calculated and validates the entire optimization process [8]-[16].

As presented by the log file, the lower bound closed the gap ensuring a globally optimality of certificate [14]. Although we declare the relative gap tolerance as a decimal number, the log file produced by the routine displays the gap as a percentage [15]-[16], [70].

That means the relative gap tolerance value is 100 times the calculated relative gap. The solution is an optimum point at the given root-node. The optimum point is met within a zero-gap tolerance and a meaningless relative gap. This solution is not suffering by floating points [15]-[16].

The log file gives as an ouput a pure binary solution. By solving the minimization problem, we consider the upper bound as the optimum point whereas the lower bound is the culprit to close the gap [70]. That optimal solution is strongly characterized as a global one. The optimal result is $\{2,6,8,9\}$ satisfying the one criterion objective function and it is shown in Fig.1.

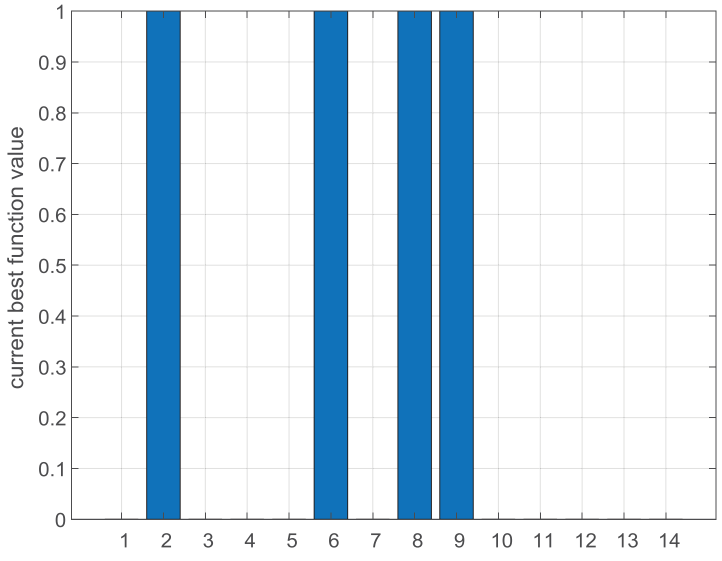

Figure 1. Optimal PMU Arrangement Achieved by bmibnb Optimizer Routine

d) Full Observability and Maximum Amount of Observability

The first instance declares an objective model with one criterion, that is, the determination of an investment responsible to install a PMU at a power grid or not to cover the full scenario of observability. In the second instance one must choose those solutions by minimizing the b-objective function, getting solutions to suffice the competitive trends in the defined objectives [51].

In this paragraph, we show how the augmented function is optimized and two criterions are satisfied at the same time giving a global optimality certificate. A binary (Boolean) model is optimized to be written in symbolic format in the YALMIP program [59], [68]-[70]. We attack this two-criterion constraint optimization problem using a linear combination of the two product terms [15].

We use a negative sign related to the maximization inside the objective function as we desire to maximize this product. Therefore, the aim is to guarantee that the outcome product being derived by the optimization fulfilled those solutions satisfying the aggregate objective [44]-[45].

Our aim of this study was to resolve the optimal PMU localization problem using optimization models with small up to large dimension size [61], [71]. For validation purposes, we use the IPOPT as an alternative local solver to count the upper bound whereas the B&B tree is developed in the direction of getting a first incumbent feasible solution which is the best integral solution [66].

This NLP routine is included in OPTI-toolbox which contains a number of open-source codes such as SCIP, lpsolve, GLPK optimizer routines [66]. As a benchmark system to test the multi-objective function, the 14 bus system is used [61], [71]. The optimization process is illustrated in Table 2.

Journal of Physics: Conference Series **2701** (2024) 012001 doi:10.1088/1742-6596/2701/1/012001

Table 2. Optimization Process: Results of bmibnb routine

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
|   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 (bilinear) 1x1    |          1 to 1|
|   #9|   Equality constraint (polynomial)  1x1|          1 to 1|
|  #10|   Equality constraint (polynomial) 1x1|          1 to 1|
|  #11|   Equality constraint (polynomial) 1x1|          1 to 1|
|  #12|   Equality constraint (polynomial) 1x1|          1 to 1|
|  #13|   Equality constraint (polynomial) 1x1|          1 to 1|
|  #14|   Equality constraint (polynomial) 1x1|          1 to 1|
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Starting YALMIP global branch & bound.
* Upper solver: ipopt
* Lower solver: SCIP
* LP solver: SCIP
* -Extracting bounds from model
* -Performing root-node bound propagation
* -Calling upper solver
******************************************************************************
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
******************************************************************************
(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:   2.64286E+00     0.00     2.64286E+00    2     1s Solution found by heuristics
* Finished.  Cost: 2.6429 (lower bound: 2.6429, relative gap 2.7451e-09%)
* Termination with relative gap satisfied
* Timing: 13% spent in upper solver (2 problems solved)
*            7% spent in lower solver (1 problems solved)
*            56% spent in LP-based domain reduction (28 problems solved)
*            1% spent in upper heuristics (1 candidates tried)
Elapsed time is 2.563467 seconds.

ans =

0   1   0   0   0   1   1   0   1   0   0   0   0   0
```

Table 2. Optimization Process: Results of bmibnb routine (continued)

ans =
2 6 7 9
Linear scalar (real, binary, 14 variables)
Current value: 2.6429
Coefficients range: 0.57143 to 0.85714
SORI =
19

Also, SCIP is used as an LP solver to solve the linear approximations problems at a given B&B tree root-node [66]. As observed, the local ipopt solver is used as an Upper Solver and detects a feasible and a local solution at the same time [70]. This fact results in an upper bound on the obtainable function value [70]. Then a convex approximation is solved. On an occasion that the lower bound is larger than the upper bound achieved so far, this node is pruned [14].

The aim is to tighten the bounds during the branching development so that the relaxation problems can be efficiently solved by spending an amount of nodes in the binary tree [70]. The relative gap between the upper and lower bound is calculated, which presents the most frequent process measurement quantity for the b&b tree.

The difference between those bounds is the optimality gap [8]-[16]. At this root-node, the upper and lower bounds are found to be equal in quantity. The upper bound of the cost function is considered the best possible solution found at the given root-node.

On the other side, the calculation of the lower bound is crucial to minimize the absolute gap giving a global certificate of optimilaty [70]. Therefore, we measure the absolute $\mathrm{UPPER} - \mathrm{LOWER}$ where a percentage relative gap is meaningless [70].

This optimum solution is the global one. This one is attained at a root-node within an absolute Gap $(\%\ 0.00)$ and a meaningless relative gap satisfied at one root-node [70]. Also, less runtime is spent due to the utilization of using heuristic calculations [68]-[70].

Those calculations are required for the optimizer routine to detect the lowest objective function value. The resulting placement vector satisfies the two criteria involved in the objective. The global solution is illustrated in Fig.2. The optimal result $\{2, 6, 7, and\ 9\}$ is the desired outcome to satisfy the topological observability and maximum amount of its [18], [23], [35], [38], [41]-[42], [44]-[45].

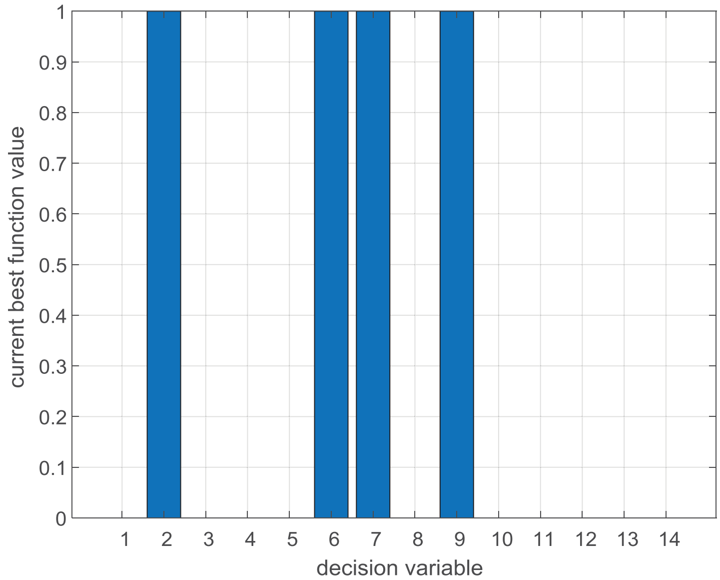

Figure 2. Optimal PMU Arrangement Achieved by bmibnb Optimizer Routine

13.Modeling and Solving the Binary (Boolean) Optimization Problem

Polynomial process optimization is to formulate a multivariate polynomial objective function consisting of a polynomial constraint function to be optimized under some restrictions of the decision variables. In this study, a linear multivariate cost function is optimized under a number of polynomial equality functions within a strict binary definition related to the nature of the decision variables.

The target of global optimization is to discover the generally the most excellent solution, that is, for the frequent issue of minimization, to achieve those design variable values that minimize the cost function on a global solution. As a benchmark case study, we use the optimal PMU allocation problem which is a combinatorial problem in which all design variables are yes/no [9], [15].

A straightforward binary-valued minimization-based problem is proposed to the PMU localization problem solving. Thus, a binary (Boolean) polynomial model comes into view to be a tailored globally optimal solution for such optimization problems [14]. The $0-1$ NLP model chooses a target at the minimization of a linear objective function subject to polynomial is making use of an iterative process which results in optimality. The entire optimization model is implemented by two optimization solvers wrapped in a MATLAB optimization package [62] compatible within the YALMIP program [59].

The optimization solvers are indexed in an upper solver which is a local nonlinear programming (NLP) solver and in a lower integer linear programming (ILP) solver [70]. Meanwhile, an LP optimizer function is used for a suitable branching the branch & bound tree towards optimality.

The desired outcome is given with an zero absolute gap as well as a satisfied relative gap thus, the whole process results in a solution with a global certificate of optimality [59], [66]-[67], [69]-[70].

14. Strategy for Using the YALMIP Global branch-and-bound

This study presents a global branch & bound algorithm, convex relaxations, a novel branching strategy, and upper and lower levels computed by NLP and ILP solvers respectively to reach an optimal solution [70]. Therefore, a novel branch-and-bound algorithm is presented based on an ILP solver performing jointly with an NLP solver in MATLAB [59]-[60], [62]-[70].

The built-in YALMIP solver interfaces an NLP solver with either FMINCON or IPOPT to count the upper bounds and a powerful ILP solver to count the lower bounds and solves the relaxed problems [70]. Those solvers measure the difference between those bounds and finally to deliver an optimal solution within a zero-gap optimality and an acceptable relative gap. The entire optimization process is summarized in the following steps [8]-[14], [56]-[58], [69]-[70].

1. A methodology to compute the lower bounds.
2. A methodology to compute the upper bounds.
3. A branching strategy and exploring nodes.
4. A termination to be done satisfying a zero-gap and a relative gap.

A considerable problem along the branching away procedure is to calculate the upper level problems using a local NLP optimizer routine. In spatial branch-and-bound implementation tree, the NLP problem is solved by the FMINCON or IPOPT optimizer routines using the solution of the convexification at the upper bound of the B&B process tree [8]. Therefore, a local search is implemented to come across the first best integral optimal solution [8], [12]-[13], [24]-[25].

In $''bmibnb''$ solver settings [70], an NLP optimizer routine such as $FMINCON$ [61] or $IPOPT$ embedded in OPTI-toolbox [66] runs at the explored nodes counting the upper bounds of the BBA's process tree giving with this way the upper lever of the cost function value [59], [69]-[70].

This upper bound must be calculated for comparison with a lower bound value which is computed by an ILP solver for the purpose of measuring the gap tolerance [59], [69]-[70]. The gap tolerance is the difference between these bounds and it is an indicator for the algorithm's termination [8]-[16].

Until then, the global branch & bound algorithm prunes the area with infeasible solutions and finally is fulfilled with an optimal solution to be achieved [59], [69]-[70].

This optimization procedure makes a comparison between the upper and lower bounds and results in an optimal solution spending a reasonable time involving calculation time. This time is divided into timing spent in upper solver's calculations and in lower solver's computations [69]-[70].

The heuristics calculation strategy in MINLP solvers is adopted to solve mixed-integer linear nonlinear programming with a non-convex structure. Heuristics is a frequent parameter being considered for algorithmic computations to attain feasible solutions. These solutions are achieved with a good performance indicator spending a reasonable amount of time to the problem solving.

These heuristic calculations are used for such non-convex nonlinear models with binary decisions, also using a NLP as well as ILP solvers in getting the optimality [59], [69]-[70]. The (MI) NLP solvers utilizes a MIP primal heuristic process in optimization by default [46], [63]-[66], [69]-[70].

Using a heuristic approach within the MILP routine, the target is to search an LP feasible solution by solving the LP relaxation to come across a current feasible solution (best integral solution) at the root-node [16], [46], [63]-[66], [69]-[70]. The optimizer routine analyzes if the solution found so far satisfies the nonlinear constraints of the proposed model [15]-[16], [24]-[25], [64].

In an opposite condition, a point is remarked as a reference for using heuristic calculations within the NLP based on a local search process in process optimization [14]-[16], [64]. Also, an amount of time is spent in an LP-domain reduction and a trivial elapsed time related to heuristics computations is spent for the purpose of getting the best possible solution [8]-[14]. The optimal solution is considered to be a global one when the absolute gap tolerance is found to be zero at the given root-node [14]-[16].

15. Modeling Binary (Boolean) Polynomial Problem with bmibnb Solver

The process optimization in a decision maker problem consists of the explanation of the decision context, and declaration of the objective function [8]-[15]. The objective must be optimized under the constraint function and the characterization of the decision variable's nature involved in a decision maker statement. It is a context where decision making faces the optimization problem in which the optimal solution is given by at least two multiple choices [9].

The optimization will resolve the optimal PMU arrangement encountered by the uncertainty of getting the global optimum point and make it true to come across it. Initially, the optimization model is stated as a single-cost constraint optimization model trying to solve it globally [16].

Then, a two-term objective constraint optimization problem is declared to be optimized in a multi-criteria process optimization and model solving [15]. As a result, the optimization technique is an agreement between two conflicting statements to come across optimality satisfying such confecting trends [15]. Those statements are ranked into a minimization product term of the objective and a maximization term getting involved in a scalarization function [8].

The minimization term reflects the least PMUs number whereas the maximization term concerns the upper limit measurement quantity of maximum observability required. In this context, a global optimal solution is achieved in a reasonable quantity involving calculation time being performed proportional to the size of the optimization problem [8]-[16], [56]-[58].

The aim is to make the entire procedure optimization a fully efficient model using optimization tools wrapped in the same optimization MATLAB software code [62]-[70]. The bmibnb solver basically analyzes the linear cost function under a non-convex constraint function whereas the binary nature of decision variables is stated in symbolic format using a suitable syntax included in the YALMIP program. The sdpsettings function enables the interfacing of the built-in YALMIP solver for solving optimization problems [59], [69]-[70].

The function sdpsettings is utilized to interface YALMIP's options with optimizer functions ranking between global ILP and nonlinear solvers [59], [69]-[70]. This function invokes suitable solvers and gives back a MATLAB output defining the minimization of the objective function towards a global optimal solution [59]. The upper bound is the solution within a 0.00% optimality [70].

Our model adopts technology advanced optimizer routines to come across optimality. Hence, and a methodology leading to a global optimal solution is well introduced in this study [62]-[70].

For that reason, a nonlinear framework is adopted involving decision variables in a binary domain to accomplish the desired target which is to be solved globally relatively easy and in a reasonable time scale. Hence, this optimization model is considered to be a pioneer model for the optimal PMU localization problem as firstly defined in [17]-[23].

As a consequence of that, a $0-1$ NLP model is programmed to be optimized by the built-in optimizer function in the YALMIP program [59], [69]-[70]. BMIBNB relaxes the binary integrality as well as the non-convex constraint function giving a polyhedral approximation that is easier to be solved in the direction of global optimality than the initial model given in a binary polynomial optimization format [59], [69]-[70]. BMIBNB solves the polynomial problem to seek out first for a feasible solution to keep with this way the enumeration tree's dimension as small as possible [70].

Then, the incumbent solution results in a global optimum at a given root-node [70]. The optimization solvers are indexed in an upper solver which is a local solver and in a lower integer linear programming (ILP) solver [8]-[15].

Meanwhile, an LP solver is used for a suitable branching of YALMIP global branch & bound tree in getting optimality [70]. This case study is managed to confirm the ability to accomplish aim and the flexible nature of the optimized manner of working object [8]-[16], [59], [63], [69]-[70].

The optimization model is optimized using the FMINCON as a local solver and a standard ILP solver such as SCIP optimizer tool to perform as a global solver to construct the B&B tree [14].

The FMINCON optimizer routine is used as an upper solver and as lower solver is preferred in a commercial package as the Gurobi [67] to achieve an optimal solution in a faster time.

For that purpose, we consider the optimization options stated by the user. These options embedeed in the solver are the declaration of $'bmibnb.lowersolver'$ and $'bmibnb.lpsolver'$ interfacing external ILP solvers and a local NLP solver is used with the option $'bmibnb.uppersolver'$ to count the upper bound [70]. The lower bounds are computed with $'bmibnb.lowersolver'$ and bound tightening is implemented by using a linear programming (LP) solver with the option such as $'bmibnb.lpsolver'$ [58], [69]-[70].

FMINCON calculates the upper bound and an ILP solver is utilized to count the lower bounds and solve the relaxed problems during the development of the B&B tree. As an ILP solver, Gurobi optimizer is selected to come across the lower bound with a satisfied relative gap [14]-[16].

The process optimization is performed where the B&B tree is developed through branch-variable, branching, and an LP propagation methodology [8]-[16].

We use the IEEE-30 bus network to show convergence towards a globally optimal solution either with one criterion or with two-criterion performing competitively regarding to the minimization of the objective function. The optimization model is executed in two stages as shown in Tables 3 and 5.

Tables 3-6 illustrate the performance of the optimization package including the two scenarios being optimized by the YALMIP b&b process. The placement result is shown in Tables 4 & 6. YALMIP global ILP solver detects a globally solution within a zero-gap tolerance as illustrated in Figures 3 & 4.

We use the influence of IPOPT and Gurobi optimizer engines to detect globally optimal solutions spending a trivial amount of runtime [62], [67]. All solvers are executed with similar performance to deliver a globally optimal solution within a zero-gap tolerance spending an acceptable elapsed time.

The optimization problem is solved at one given root-node as illustrated in relative tables. The process optimization is illustrated in Table 3. The best possible sites are shown in Table 4 and Fig.3.

The B&B algorithmic process is terminated when the solution point is delivered within a percentage Gap ($\%\ 0.00$) and a meaningless relative gap. Therefore, an explicit optimal solution is achieved by starting a YALMIP global branch & bound routine [69]-[70].

The determination of the global optimum point is the desired goal, and for the specific optimization problem, a solution is finally detected during the b&b tree implementation towards optimality [16].

We optimize a single-cost function with binary-valued variables under a constraint function and binary decision variables restrictions. We discover the solution point which minimizes the objective function satisfying optimality and feasibility metrics [16].

Optimizing and giving a desired outcome, a least PMU number is located for full conditions of observability purposes as those defined in [17]. In a second stage, the objective is extended to a two-term function (minimize/maximize) to cover the topic of achieving the maximum level of observability. The entire optimization gives solution using *bmibnb* built-in solver in the YALMIP.

This algorithm shceme is implemented based on ILP solvers jointly with an NLP solver fully compatible with the MATLAB environment. We focus on algorithms in the binary domain to declare a $0-1$ nonlinear programming model with the aim to attain a global optimal solution [70].

As observed in the log file illustrated in the Tables 3 & 5, the YALMIP BBA converges to an optimal solution within 1.272357 seconds. The percentage relative gap percentage is calculated as:

$$\text{relative gap} = 100 \times (\text{UPPER} - \text{LOWER})/(1 + |\text{UPPER}|) \tag{25}$$

That gap tolerance is a difference between UPPER and LOWER bounds of the cost function that bmibnb computes in its branch-and-bound algorithm [70]. We implement a minimization problem using a built-in YALMIP global integer solver where the relative gap is meaningless [70].

The solver returns a solution with a 0.00% optimality. A zero-gap tolerance is a necessity and a relative gap also succeeded within a predefined optimality tolerance [70].

Table 3. Optimization Process: Results of bmibnb routine

```
+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
* Starting YALMIP global branch & bound.
* Upper solver: ipopt
* 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:  1.00000E+01   0.00   1.00000E+01   2     2s Solution found by heuristics
* Finished.  Cost: 10 (lower bound: 10, relative gap 9.0909e-10%)
```

Table 3. Optimization Process: Results of bmibnb routine (continued)

* Termination with relative gap satisfied
* Timing: 14% spent in upper solver (2 problems solved)
* 4% spent in lower solver (1 problems solved)
* 16% spent in LP-based domain reduction (60 problems solved)
* 1% spent in upper heuristics (1 candidates tried)
Elapsed time is 1.216967 seconds.
ans =
Columns 1 through 17
1 0 0 0 0 1 1 0 0 1 1 1 0 0 0 0 0
Columns 18 through 30
1 0 0 0 0 1 0 0 1 0 0 0 1
Linear scalar (real, binary, 30 variables)
Current value: 10
Coefficients range: 1 to 1
SORI =
40
ans =
1 6 7 10 11 12 18 23 26 30

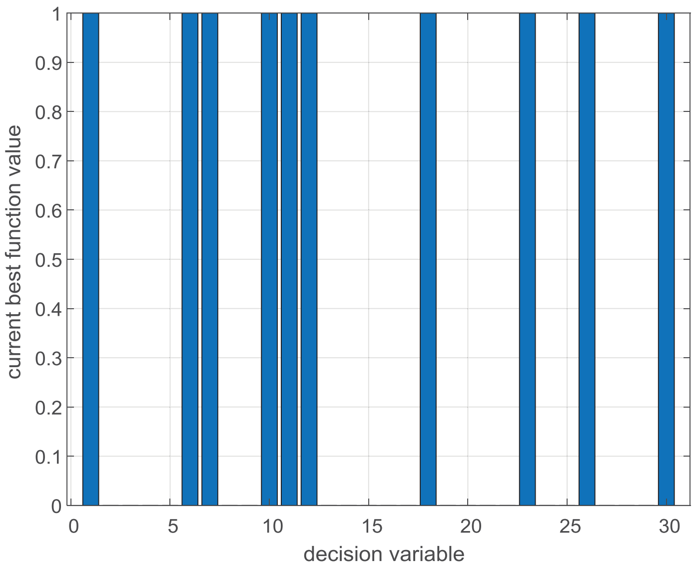

FIGURE 3. Optimal PMU Arrangement Achieved by bmibnb Optimizer Routine

TABLE 4. Optimal PMU Arrangement for the 30-bus power network

PMU Arrangement Locations with Observability Indicator Equal to 35
1, 6, 7, 10, 11, 12, 18, 23, 26, 30

As we observed, the optimal solution is a global one given that the gap is equal to zero. The optimality conditions are finally satisfied within pre-specified stopping and tolerance criteria whereas the optimal global solution is attained without using clustering methods to come across it [16].

YALMIP program's solver can solve the programming model with non-convexities incorporating a powerful global Gurobi solver and a local IPOPT solver [66]. The problem is optimized when certain assessment criteria and the global solution is fulfilled [8]-[16].

The optimal results show that this numerical method is not wasteful and converges to the true optimality. This optimal solution is accepted because the upper and lower levels are found to be equal in number minimizing the gap tolerance with a satisfied relative gap for the two minimization models.

Hence, when no more minimization in the cost function happens during the iterative process, the algorithm ends up with the recognition of the identity of the global optimum point [14]-[15].

So, an unambiguous outcome found to support the effectiveness of the polynomial programming model to come across optimality. The experimental computational results illustrated that the BBA is able to deliver globally optimal points even for large-scale power networks [70].

Then, the single-objective function is expanded in two competitive terms satisfying two criteria: cost minimization and maximum reliability for wide-area and state estimation applications. This mathematical based model is efficiently solved for both case studies, that is, to minimize the cost and to achieve those solutions with maximum observability indicator [18], [44]-[45]. The desirable solution is achieved by interfacing outer solvers running in the MATLAB environment [70].

YALMIP global branch & bound is based on integer linear solvers to solve the lower bounding relaxed problems, and nonlinear routines to calculate the upper bound [70]. Due to branching strategy, exploring nodes, pruning infeasible regions by this optimizer function, the optimal solution is different with the same least cost function value. We set the option bmibnb ‘lpreduce’ to zero [70] so as to definitely reduce the computational time spent to solve the optimization globally [18], [44]-[45].

Table 5. Optimization Process: Results of bmibnb routine

+++
* Starting YALMIP global branch & bound.
* Upper solver: ipopt
* Lower solver: GUROBI
* LP solver: GUROBI
* -Extracting bounds from model
* -Preforming 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: 8.26667E+00 0.00 8.26667E+00 2 1s Solution found by heuristics
* Finished. Cost: 8.2667 (lower bound: 8.2667, relative gap 1.0791e-09%)
* Termination with relative gap satisfied
* Timing: 12% spent in upper solver (2 problems solved)
* 6% spent in lower solver (1 problems solved)

Table 5. Optimization Process: Results of bmibnb routine (continued)

*	15% 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: 0.1015	
solvertime: 1.1635	
info: 'Successfully solved (BMIBNB)'	
problem: 0	
Elapsed time is 1.272357 seconds.	
ans =	
8.2667	
ans =	
Columns 1 through 16	
0 1 0 1 0 1 0 0 1 1 0 1 0 0 1 0	
Columns 17 through 30	
0 1 0 0 0 0 0 0 1 0 1 0 0 0	
ans =	
2 4 6 9 10 12 15 18 25 27	
Linear scalar (real, binary, 30 variables)	
Current value: 8.2667	
Coefficients range: 0.73333 to 0.93333	
SORI =	
52	

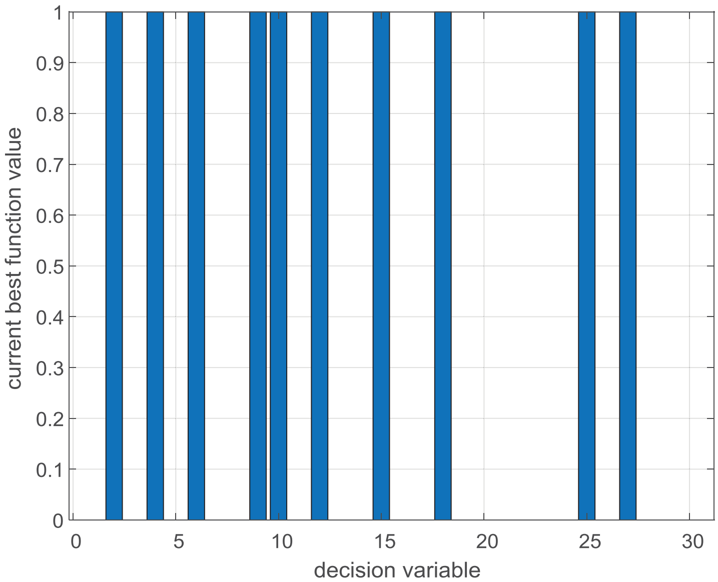

FIGURE 4. Optimal PMU Arrangement Achieved by bmibnb Optimizer Routine

TABLE 6. Optimal PMU Arrangement for the 30-bus Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 52
2, 4, 6, 9, 10, 12, 15, 18, 25, 27

16. Implementation of binary polynomial problem using medium-sized power networks

Relaxing the polynomial constraints as well as the binary integrality of the decision variables doesn't always importantly decrease the complexity of the solution of the proposed mathematical model. The not easy task is involved with the number of the decision variables linking together to formulate each constraint getting involved in the optimization process [8]-[16].

The main target of the relaxation of the polynomial optimization model is to make the programming model easier to be solved and for which is easier to attain a global optimal solution spending one root-node [56]-[58]. This numerical procedure is a hybrid B&B algorithm which performs with an outer approximation NLP solver to the optimization problem solving on global point.

The $'bmibnb'$ solver relies on a spatial branch-and-bound algorithm (s-BBA) which adopts convex approximations for nonlinear calculations during the construction of the binary tree [59], [63], [69]-[70]. Proper simulation results are generated in MATLAB using technology advanced optimization solvers with a sufficient and adequate optimization way [14], [61]-[62], [68]-[70].

Hence, the optimal solution is collected by performing the $0-1$ nonlinear model using powerful NLP as well as ILP standard solvers [66]-[70]. For that reason, the proposal model is tested on optimization models including from a few variables up some thousands of decision variables.

After the declaration of the Boolean polynomial optimization model, medium-sized power networks are used for the optimization task. Their data are taken by using the MATPOWER software package [61]. The optimization model is programmed in MATLAB & YALMIP platforms [62], [68].

The YALMIP global branch & bound adopts strategies integrated in its implementation to transform the original model into a polyhedral approximation [70]. Hence, a number of convex sub-regions are created for the purpose of getting the best possible solution satisfying the feasibility of the new one model. So, this solution is considered to be the same for the initial model [16].

The optimization process gives tighter variable bounds being achieved by considering the constraint function and the current bounds in the manner by which the B&B tree is developing [70]. Local domain reductions are noticed at the incumbent root-node of the B&B tree [69]-[70].

Domain propagation is a programming methodology used by mixed-integer linear solvers [14]-[16]. This branching methodology is performed at every root-node of the B&B tree in tightening the local domains of each decision variable [14]-[15], [46]. Therefore, a strong branching is developed with the help of propagation embedded in an ILP solver to come across the optimality [46].

Hence, a powerful branching is developed using domain propagation methodology which is considerably advantageous in our (MI) NLP model with the help of an ILP solver and an NLP solver.

The whole optimization package has the ability to recognize the global optimum point being covered by a warranty that the optimality gap goes to zero when the upper and lower bounds are totally equal and the relative gap is satisfied. The lower bound minimizes the suboptimality criteria, thus the solution is calculated within an absolute gap. Hence, the zero-gap is achieved [70].

The 57-and 118-bus systems are used for simulating the binary (Boolean) polynomial optimization model. The second and third experimental tests are larger polynomial problems. The process optimization is performed using the 57-, 118-bus systems to come across the optimality [61], [71].

As observed, a handle constraint mechanism adopted in an ILP solver is implemented to tighten the bound called domain propagation [8]-[14]. The best possible outcome satisfies the competitive trends involved in the objective function. Heuristic calculations happen accelerating the entire optimization and recognizing the incumbent solution with the lowest objective function value. This lower value is considered to be the global solution within an absolute Gap (% 0.00) as shown in Table 7.

During the iterative process, the branching and nodes are being explored for the purpose of getting a global optimum point by an ILP optimizer function. For that purpose, the LP relaxations and convex approximations are necessary to be computed to come across the optimality [8]-[16].

The optimizer functions are used for solving the relaxation problems, and nonlinear solvers for the upper bound calculations [8]-[14], [56]-[58], [70]. The local solver is used for calculating upper bounds [70]. Also, a lower solver which is an ILP optimizer routine is used for counting the lower solver and solving the relaxed problems to develop the binary enumaration tree [70].

This process in optimization can be viewed as a method done repeatedly to attain a comparison between the difference of the current upper bound, which is the cost incumbent value (best integral solution), and the lower bound at present, which is the least of the lower bounds of the running for selection sub-problems [8]-[14], [56]-[58], [70]. The whole framework is terminated at a root-node giving the global solution optimality where optimality criteria are met. The PMU localization sites are shown in Figures 5 and 6 and shown in relative Tables.

Table 9 illustrates the procedure optimization and tested on a 118-bus system for the purpose of finding the global optimum point within a zero-gap tolerance and meaningless relative gap, [59]-[70].

The solution is the upper bound for the minimization problem while the lower bound is culprit to close the gap [68]. Hence, the difference of them is equal to zero giving an absolute gap equal to zero [70]. Hence, an optimality certificate has been achieved and validated the entire optimization process.

Table 7. Optimization Process: Results of bmibnb routine

++
* Starting YALMIP global branch & bound.
* Upper solver: fmincon
* Lower solver: SCIP
* LP solver: SCIP
* -Extracting bounds from model
* -Performing 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

Table 7. Optimization Process: Results of bmibnb routine (continued)

Node	Upper	Gap (%)	Lower	Open	Time
1:	1.70000E+01	0.00	1.70000E+01	2	6s Solution found by heuristics
* Finished. Cost: 17 (lower bound: 17, relative gap 5.5555e-10%)					
* Termination with relative gap satisfied					
* Timing: 4% spent in upper solver (2 problems solved)					
* 19% spent in lower solver (1 problems solved)					
* 101% spent in LP-based domain reduction (114 problems solved)					
* 1% spent in upper heuristics (1 candidates tried)					
Elapsed time is 6.167039 seconds.					
sol =					
struct with fields:					
yalmipversion: '20210331'					
matlabversion: '9.4.0.813654 (R2018a)'					
yalmiptime: 0.1864					
solvertime: 32.2366					
info: 'Successfully solved (BMIBNB)'					
problem: 0					
Columns 33 through 48					
0 0 0 1 0 0 1 0 1 0 0 0 1 1 1 0					
Columns 49 through 57					
0 1 0 1 0 1 0 0 0					
Linear scalar (real, binary, 57 variables)					
Current value: 17					
Coefficients range: 1 to 1					
SORI =					
61					
ans =					
Columns 1 through 16					
2 6 12 19 22 25 27 32 36 39 41 45 46 47 50 52					
Column 17					
54					

Table 7. Optimization Process: Results of bmibnb routine (continued)

Nonlinear scalar (real, models 'come across', 1 variable)
Current value: 2
Coefficients range: 1 to 1
Nonlinear scalar (real, models 'milpsubsref', 1 variable)
Current value: 1
Coefficients range: 1 to 1

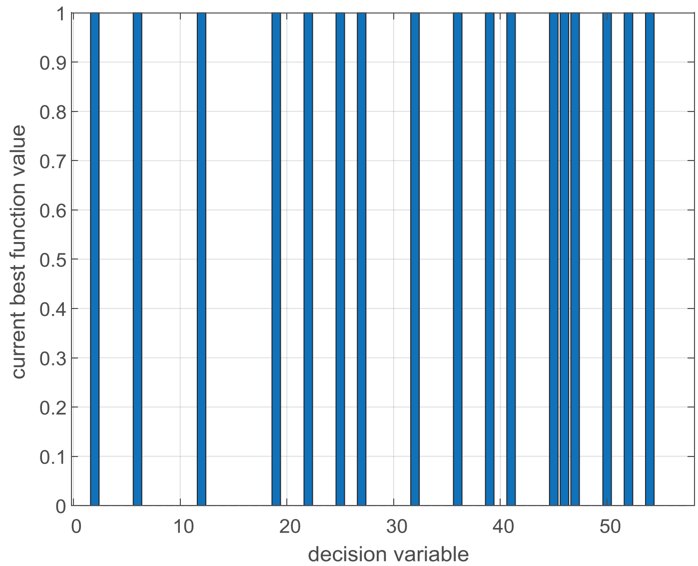

Figure 5. Optimal PMU Arrangement Achieved by bmibnb Optimizer Routine

Table 8. Optimal PMU Arrangement for the 57-bus Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 61
2, 6, 12, 19, 22, 25, 27, 32, 36, 39, 41, 45, 46, 47, 50, 52, 54

Table 9. Optimization Process: Results of bmibnb routine

+++
* Starting YALMIP global branch & bound.
* Upper solver: fmincon
* Lower solver: INTLINPROG
* LP solver: INTLINPROG
* -Extracting bounds from model
* -Performing root-node bound propagation
* -Calling upper solver (no solution found)
* -Branch-variables: 118
* -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

Table 9. Optimization Process: Results of bmibnb routine (continued)

1 : 3.20000E+01 0.00 3.20000E+01 2 32s Solution found by heuristics
* Finished. Cost: 32 (lower bound: 32, relative gap 3.0302e-10%)
* Termination with relative gap satisfied
* Timing: 18% spent in upper solver (2 problems solved)
* 4% spent in lower solver (1 problems solved)
* 14% spent in LP-based domain reduction (236 problems solved)
* 1% spent in upper heuristics (1 candidates tried)
sol =
struct with fields:
yalmipversion: '20210331'
matlabversion: '9.4.0.813654 (R2018a)'
yalmiptime: 0.1864
solvertime: 32.2366
info: 'Successfully solved (BMIBNB)'
problem: 0
Elapsed time is 32.424994 seconds.
ans =
Columns 1 through 16
1 0 0 0 1 0 0 0 0 1 0 1 0 0 1 0
Columns 17 through 32
1 0 0 1 0 0 1 0 0 0 0 1 0 1 0 0
Columns 33 through 48
0 0 0 1 0 0 0 1 0 0 0 1 0 1 0 0
Columns 49 through 64
1 0 0 1 0 0 0 1 0 0 0 0 0 1 0 1
Columns 65 through 80
0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 1
Columns 81 through 96
0 0 0 0 1 0 1 0 0 1 0 0 0 1 0 0
Columns 97 through 112

Table 9. Optimization Process: Results of bmibnb routine (continued)

0 0 0 0 0 1 0 0 1 0 0 0 0 1 0 0
Columns 113 through 118
0 1 0 1 0 0
Linear scalar (real, binary, 118 variables)
Current value: 32
Coefficients range: 1 to 1
ans =
32
SORI =
152
ans =
Columns 1 through 16
1 5 10 12 15 17 20 23 28 30 36 40 44 46 49 52
Columns 17 through 32
56 62 64 71 75 77 80 85 87 90 94 102 105 110 114 116

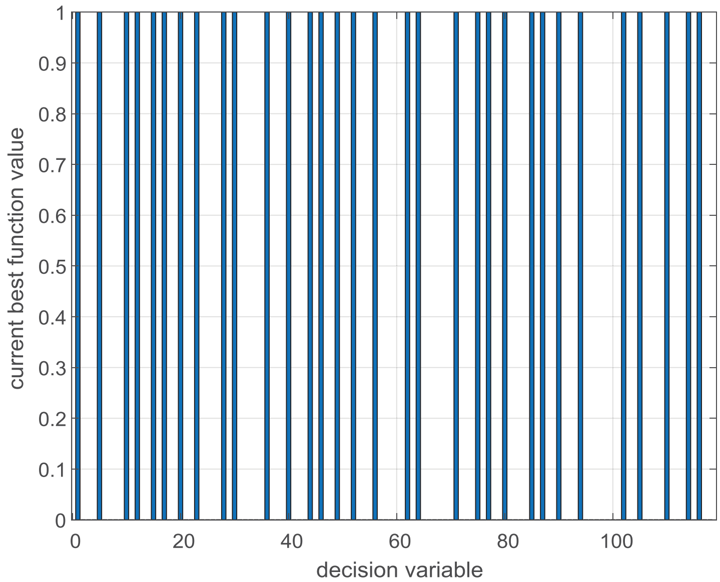

FIGURE 6. Optimal PMU Arrangement Achieved by bmibnb Optimizer Routine

TABLE 10. Optimal PMU Arrangement for the 118-bus Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 152
1, 5, 10, 12, 15, 17, 20, 23, 28, 30, 36, 40, 44, 46, 49, 52, 56, 62, 64, 71, 75, 77, 80, 85, 87, 90, 94 102, 105, 110, 114, 116

After getting an optimal solution satisfying one criterion objective function, the second instance declares a two-criterion model. The iterative process is executed again and trade-off globally optimal solutions are derived having the maximum observability indicator [18], [23], [44]-[45].

Multi-criteria optimization model is executed on a 57-bus and 118-bus system and a desired outcome is achieved as illustrated in Table 11 & 13 [61], [71]. The aim is at minimizing an objective function with the best outcome with a maximum System Observability Index (SORI) [18].

This indicator is related to the aggregate count by which a power network is monitored either directly or indirectly by a resulting PMU arrangement [44]-[45]. bmibnb optimizer interfaces SCIP and INTLINPROG routine and an NLP routine for counting the lower and upper bounds respectively as shown in Tables 11 & 13.

The LP solver does the branching rules using suitable LP relaxations and heuristic calculations being necessary to end the iterative process with success getting an optimal solution [70].

The optimization terminates with a globally optimal solution at a given root-node [70]. The solution is achieved within optimality criteria 0.00 %. This criteria means no better solution can be found therefore, a globally optimal solution has been found. To present the algorithmic scheme and its results, we illustrate the log file produced by the bmibnb optimizer routine in Tables 11 and 13.

The process optimization is shown in Tables 11 and 13 whereas the placement sites are shown in Tables 12 and 14 with the plot-diagram shown in Fig 7 and 8.The PMU positioning sites are illustrated in Figures 7 & 8 and tabulated in Tables 12 &14 for the 57 bus and 118 bus systems [61], [71].

As observed, the entire optimization process ends up at one given root delivering a globally optimal solution in an affordable elapsed time [8]-[15], [62]-[70]. According to the B&B process, gap tolerance results in zero satisfying that the polyhedral approximation is efficiently solved.

That solution satisfies the initial model which is declared as a nonlinear program with binary decision variables. As shown by the resulting outcome, the YALMIP returns an optimal solution within a gap equal to be zero (% 0.00) [58], [62]-[67], [69]-[70]. Also, a meaningless relative gap is reported. There, any better solution doesn't exist. Hence, this one is a global solution [14].

Table 11. Optimization Process with Maximum Observability

++
* Starting YALMIP global branch & bound.
* Upper solver : fmincon
* Lower solver : SCIP
* LP solver : SCIP
* -Extracting bounds from model
* -Performing 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: 1.57368E+01 0.00 1.57368E+01 2 9s Solution found by heuristics
* Finished. Cost: 15.7368 (lower bound: 15.7368, relative gap 5.9749e-10%)

Table 11. Optimization Process with Maximum Observability (continued)

* Termination with relative gap satisfied
* Timing: 4% spent in upper solver (2 problems solved)
* 38% spent in lower solver (1 problems solved)
* 71% spent in LP-based domain reduction (114 problems solved)
* 1% spent in upper heuristics (1 candidates tried)
Elapsed time is 9.425901 seconds.
ans =
Columns 1 through 16
1 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0
Columns 17 through 32
0 0 0 1 0 0 0 1 0 0 0 1 0 1 0 1
Columns 33 through 48
0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 0
Columns 49 through 57
0 1 0 0 1 0 0 0 1
Linear scalar (real, binary, 57 variables
Current value: 15.7368
Coefficients range: 0.87719 to 0.96491
SORI =
72
ans =
Columns 1 through 16
1 4 6 9 15 20 24 28 30 32 36 38 41 47 50 53
Column 17
57
Nonlinear scalar (real, models 'come across', 1 variable)
Current value: 1
Coefficients range: 1 to 1
Nonlinear scalar (real, models 'milpsubsref', 1 variable)
Current value: 1
Coefficients range: 1 to 1

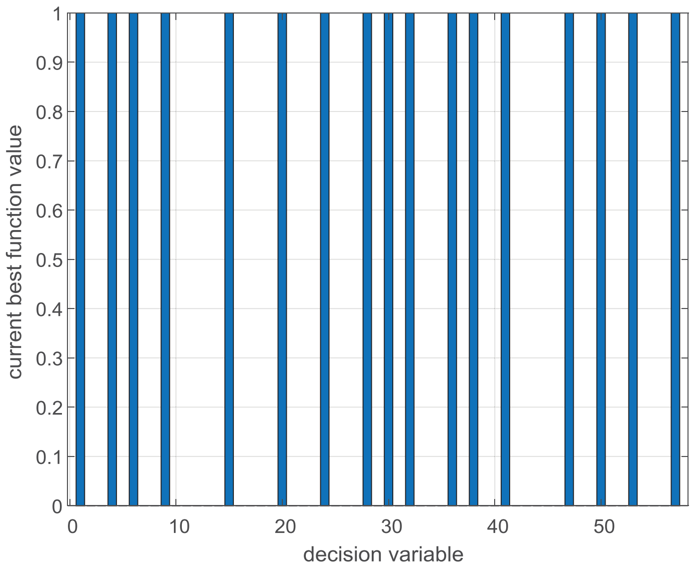

FIGURE 7. Optimal PMU Arrangement Achieved by bmibnb Optimizer Routine

TABLE 12. Optimal PMU Arrangement for the 57-bus Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 72
1, 4, 6, 9, 15, 20, 24, 28, 30, 32, 36, 38, 41, 47, 50, 53, 57

The optimal result is found to be the desired outcome having a $SORI\ indicator\ equal\ to\ 164$ for the 118 bus system [18], [44]-[45]. Thus, this solution satisfies the two-term objective function. Table 15 illustrates the entire optimization's performance metrics using standard ILP solvers and FMINCON or Ipopt as a local solver.

The capability of YALMIP global branch & bound algorithm is illustrated by the fact of using NLP solvers such as FMINCON or IPOPT as upper solvers and ILP solvers for instance GLPK, SCIP, Intlinprog, LPSOLVE and Gurobi as lower solvers [62]-[70].

Then, we turn the initial $0-1$ ILP model presented in Esq. (3)-(5) into an augmented ILP model with measurement level of observability issues presented in Esq. (11)-(13). Based on the above logic statement, a zero-one constraint integer linear programming model must be optimized in two phases.

Table 13. Optimization Process with Maximum Observability

++
* Starting YALMIP global branch & bound.
* Upper solver: fmincon
* Lower solver: INTLINPROG
* LP solver: INTLINPROG
* -Extracting bounds from model
* -Performing root-node bound propagation
* -Calling upper solver (no solution found)
* -Branch-variables: 118
* -More root-node bound-propagation
* -Performing LP-based bound-propagation
* -And some more root-node bound-propagation
* Starting the b&b process

Table 13. Optimization Process with Maximum Observability (continued)

Node	Upper	Gap (%)	Lower	Open	Time
1:	3.06102E+01	0.00	3.06102E+01	2	38s Solution found by heuristics

```
* Finished.  Cost: 30.6102 (lower bound: 30.6102, relative gap 3.1634e-10%)
* Termination with relative gap satisfied
* Timing: 10% spent in upper solver (2 problems solved)
*          5% spent in lower solver (1 problems solved)
*         17% spent in LP-based domain reduction (236 problems solved)
*          1% spent in upper heuristics (1 candidates tried)
Elapsed time is 39.010560 seconds.
ans =

  Columns 1 through 16

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

  Columns 17 through 32

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

  Columns 33 through 48

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

  Columns 49 through 64

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

  Columns 65 through 80

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

  Columns 81 through 96

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

  Columns 97 through 112

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

  Columns 113 through 118

    0   1   0   0   0   0

Linear scalar (real, binary, 118 variables)
Current value: 30.6102
Coefficients range: 0.91525 to 0.98305

SORI =
```

Table 13. Optimization Process with Maximum Observability (continued)

```
 164

ans =

  Columns 1 through 16

   3   5   9   12   15   17   21   23   28   30   34   37   40   45   49   53

  Columns 17 through 32

   56   62   64   68   71   75   77   80   85   86   91   94   101   105   110   114
```

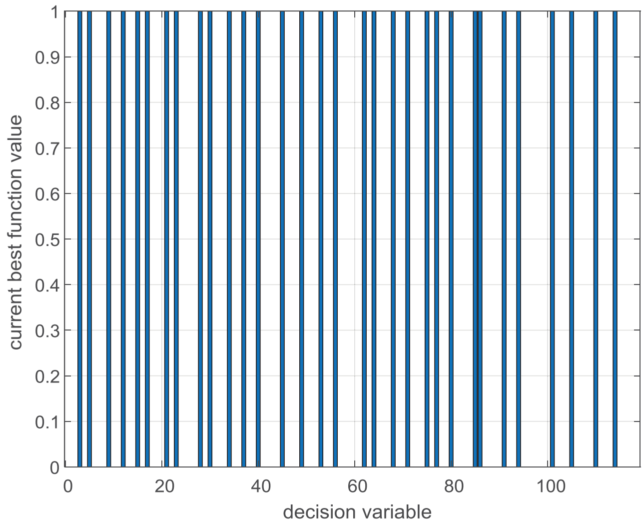

FIGURE 8. Optimal PMU Arrangement Based on the Maximum Observability

TABLE 14. Optimal PMU Arrangement for the 118-bus Power Network

PMU Arrangement Locations Maximum Observability Indicator Equal to 164
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, 91, 94 101, 105, 110, 114

TABLE 15. Best PMU Arrangement for Standard Power Systems

Intlinprog and Fmincon Optimizer Routines to the binary (Boolean) Polynomial Problem		
IEEE bus system	Single-Cost Function	Aggregate-Cost Function
14-bus	2, 8, 10, 13	2, 6, 7, 9
30-bus	3, 5, 8, 10, 11, 12, 18, 23, 25, 29	2, 4, 6, 9, 10, 12, 15, 18, 25, 27
57-bus	3, 6, 12, 15, 19, 22, 25, 27, 32, 36, 39 41, 44 47, 50, 52, 54	1, 4, 6, 9, 15, 20, 24, 28, 30, 32, 36 38, 39, 41 46, 50, 53
118-bus	2, 5, 10, 12, 15, 17, 20, 23, 28, 30, 36 40 44, 46, 49, 52, 56, 62, 64, 71, 75, 77, 80, 85, 87, 90, 94, 101, 105, 110, 114, 116	3, 5, 9, 12, 15, 17, 21, 25, 29, 34, 37 40, 45 49, 52, 56, 62, 64, 68, 70, 71 78, 85 86, 89, 92, 96, 100, 105, 110 114, 118

TABLE 15. Best PMU Arrangement for Standard Power Systems (continued)

SCIP and Fmincon Optimizer Routines to the Binary (Boolean) Optimization Problem		
IEEE bus system	Single-Cost Function	Aggregate-Cost Function
14-bus	2, 7, 11, 13	2, 6, 7, 9
30-bus	1, 2, 6, 9, 10, 12, 15, 18, 25, 27	2, 4, 6, 9, 10, 12, 15, 19, 25, 27
57-bus	1, 4, 9, 20, 24, 25, 28, 29, 32, 36, 38, 39, 41 44, 46, 51, 54	1, 4, 6, 9, 15, 20, 24, 28, 31, 32, 36 38, 41 46 50, 53, 57
118-bus	1, 7, 9, 11, 12, 17, 21, 25, 28, 34, 37, 41 45, 49 52, 56, 62, 64, 71, 72, 75, 77, 80, 85, 87 90, 94, 101, 105, 110, 114, 116	3, 5, 9, 12, 15, 17, 21, 25, 28, 34, 37 40, 45 49, 52, 56, 62, 64, 68, 70, 71 78, 85, 86, 89 92, 96, 100, 105, 110 114, 118
GLPK and Fmincon Optimizer Routines to the Binary (Boolean) Optimization Problem		
IEEE bus system	Single-Cost Function	Aggregate-Cost Function
14-bus	2, 7, 11, 13	2, 6, 7, 9
30-bus	3, 5, 8, 9, 10, 12, 19, 23, 26, 29	2, 4, 6, 9, 10, 12, 15, 20, 25, 27
57-bus	1, 6, 13, 15, 19, 22, 25, 27, 32, 36, 39, 41, 44 47, 51, 52, 54	1, 4, 6, 9, 15, 20, 24, 28, 30, 32, 36 38, 39 41 47, 51, 53
118-bus	2, 5, 10, 11, 12, 17, 21, 24, 25, 29, 34, 37 40 45 49, 52, 56, 62, 63, 68, 73, 75, 77, 80, 85 86, 90 94, 101, 105, 110, 114	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, 101, 105 110, 115
lpsolve and Fmincon Optimizer Routines to the Binary (Boolean) Optimization Problem		
IEEE bus system	Single-Cost Function	Aggregate-Cost Function
14-bus	2, 8, 10, 13	2, 6, 7, 9
30-bus	1, 2, 6, 10, 11, 12, 15, 19, 25, 27	2, 4, 6, 9, 10, 12, 15, 18, 25, 27
57-bus	1, 2, 6, 10, 19, 22, 25, 26, 29, 32, 36, 39, 41 44 46, 49, 54	1, 4, 6, 9, 15, 20, 24, 25, 28, 32, 36 38, 39, 41 46, 51, 53
118-bus	1, 5, 10, 11, 12, 17, 21, 25, 29, 34, 37, 40 45 49, 52, 56, 62, 63, 68, 70, 71, 75, 77, 80, 85, 87, 91, 94, 101, 105, 110, 114	3, 5, 9, 12, 15, 17, 21,25, 29, 34, 37 40, 45, 49 53, 56, 62, 64, 68, 70, 71 78, 85 86, 89 92, 96 100, 105, 110 114, 118
Gurobi and Fmincon Optimizer Routines to the Binary (Boolean) Optimization Problem		
IEEE bus system	Single-Cost Function	Aggregate-Cost Function
14-bus	2, 8, 10, 13	2, 6, 7, 9
30-bus	1, 5, 8, 10, 11, 12, 19, 23, 26, 29	2, 4, 6, 9, 10, 12, 15, 19, 25, 27
57-bus	1, 4, 6, 13, 20, 22, 25, 27, 29, 32, 36, 41, 45 47, 51, 54, 57	1, 4, 6, 9, 15, 20, 24, 28, 31, 32, 36 38, 41 46, 51, 53, 57
118-bus	3, 5, 10, 12, 15, 17, 21, 23, 29, 30, 36, 40 44, 46, 49, 52, 56, 62, 64, 71, 75, 77, 80, 85 87 90, 94 101, 105, 110, 115, 116	3, 5, 9, 12, 15, 17, 21, 25, 29, 34, 37 40, 45, 49, 52, 56, 62, 64, 68, 70, 71 76, 78, 85, 86 89, 92, 96, 100 105, 110, 114

In the first one, we seek out a global solution meaning that a least number of PMU are needed to cover the full condition of topological observability [17]. The second phase covers the scenario where a resulting PMU placement is an adequate configuration to cover a maximum observability condition [18], [44]-[45]. The 118-bus system is used as a benchmark system, and the GLPK and SCIP routines are used for optimizing both optimization models Esq. (3)-(5) and Esq. (11)-(13).

Two instances are studied. The first one minimizes an objective function with the aim to find the least PMUs number while the second one finds solution with 0.00 %. This solution is the least PMUs number where the network observability is found to have the maximum value [44]-[45].

Tables 16 and 19 illustrates the iterative process being executed for the purpose of getting an optimal solution for the $0-1$ ILP model. The message window is displayed to a log file. The MIP log has the appearance displayed in Tables 16 & 18 [66].

MPL shows the advancement of GLPK routine while an optimization run is performed. GLPK gives the number of iterations and the objective function value being derived and the status messages which gives the infeasibilities during the revised simplex implementation [66].

Table 16. Optimization Process: Results of GLPK routine

GLPK Simplex Optimizer, v4.48
118 rows, 118 columns, 476 non-zeros
0 : obj = 0.000000000e+000 infeas = 1.180e+002 (0)
* 148: obj = 3.600000000e+001 infeas = 0.000e+000 (0)
* 172: obj = 3.200000000e+001 infeas = 0.000e+000 (0)
OPTIMAL SOLUTION FOUND
Integer optimization begins...
+ 172: mip = not found yet >= -inf (1; 0)
+ 190: >>>>> 3.200000000e+001 >= 3.200000000e+001 0.0% (3; 0)
+ 190: mip = 3.200000000e+001 >= tree is empty 0.0% (0; 5)
INTEGER OPTIMAL SOLUTION FOUND

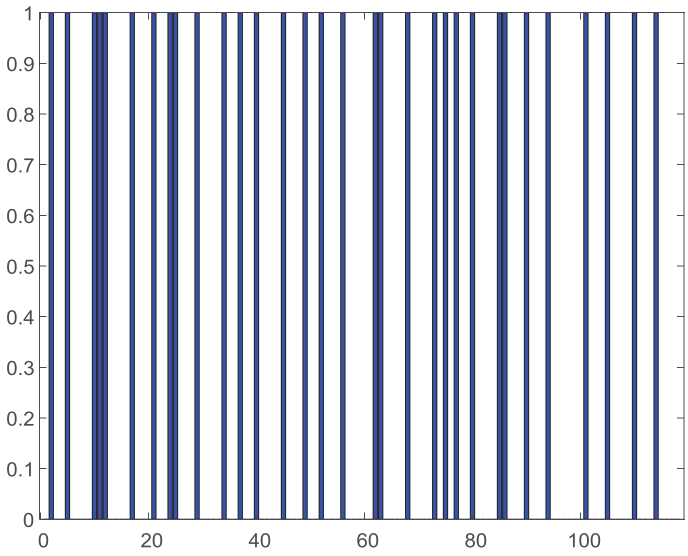

FIGURE 9. Optimal PMU Arrangement

TABLE 17. Optimal PMU Arrangement for the 118-bus Power Network

PMU Arrangement Locations Maximum Observability Indicator Equal to 156
2, 5, 10, 11, 12, 17, 21, 24, 25, 29, 34, 37, 40, 45, 49, 52, 56, 62, 63, 68, 73, 75, 77, 80, 85, 86, 90 94, 101, 105, 110, 114

Table 18. Optimization Process: Results of GLPK routine

GLPK Simplex Optimizer, v4.48
118 rows, 118 columns, 476 non-zeros
0 : obj = 0.000000000e+000 infeas = 1.180e+002 (0)
* 148: obj = 3.446610169e+001 infeas = 0.000e+000 (0)
* 187: obj = 3.061016949e+001 infeas = 0.000e+000 (0)
OPTIMAL SOLUTION FOUND
Integer optimization begins...
+ 187: mip = not found yet >= -inf (1; 0)
+ 189: >>>>> 3.061016949e+001 >= 3.061016949e+001 0.0% (1; 0)
+ 189: mip = 3.061016949e+001 >= tree is empty 0.0% (0; 1)
INTEGER OPTIMAL SOLUTION FOUND

$\text{pecentage relative gap} = 100 * |\text{best integer feasible solution found so far} - \text{best bound}|/(1e-10 + |\text{best bound}|)$; Termination criteria: $\text{Gap} = 0$; [66]

The PMU locations are displayed in Fig. 10 and Table 19.

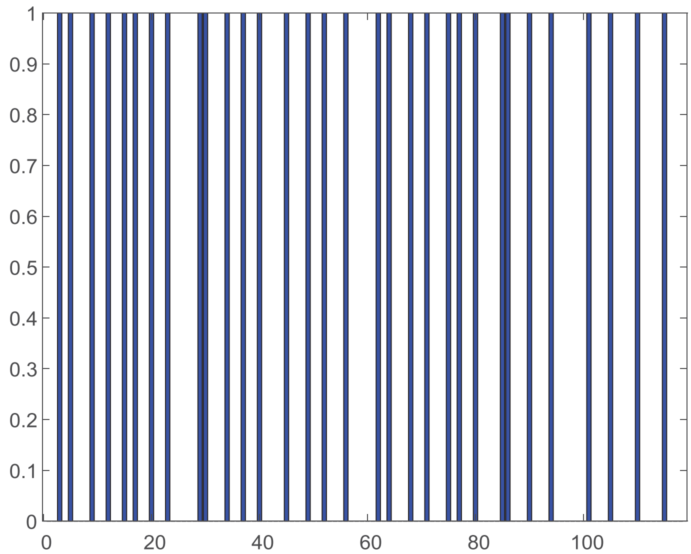

FIGURE 10. A PMU Placement Set with Maximum Indicator of Observability

TABLE 19. Optimal PMU Arrangement for the 118-bus Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 164
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, 101, 105, 110, 115

GLPK gives the data related to the initial linear programming relaxation while the b&b search shows the node count, the second column gives the best integer feasible solution delivered so far. The third column gives the best bound inside a relative gap at the particular step in the process [66].

As shown in Tables 16 and 18, GLPK explores and prunes infeasible regions, seeking out for the best feasible solution and finally gets the best possible solution within a zero-gap tolerance [66].

The B&B tree is implemented by GLPK routine. This optimizer function is an essential process to locate promising feasible solution points in the space of optimality. We can observe that the optimum point has been achieved since the difference between the upper and lower bounds goes to zero so the optimal solution can be characterized as Globally Optimal by far [62].

Hence, the tree is empty, and ends up with a solution with 0.00 %. An integer optimal solution has been found as the status message displays. An optimal solution is achieved by counting the optimality gap equal to zero which means that no better optimum point can be found by the ILP solution solver [8]-[16], [66].

SCIP executes the $0-1$ ILP model, counting constantly the primal and dual bounds during the iterative process. The Primal bound is found to be equal with the Dual Bound and it is culript for the mininimization of the gap [63]-[66]. The gap is minimized so the certificate of optimality has been ensured [64]. The optimal results are shown in Tables 20 & 22. Table 20 shows the log file produced by SCIP. SCIP discovers a zero difference between those bounds. The Primal bound is the global solution for the minimization model and culpit for this is the Dual bound which closes the gap [65].

Table 20. Optimization Process: Results of SCIP routine

time	node	left	LP iter	LP it/n	mem	mdpt	frac	vars	cons	cols	rows	cuts	confs	strbr	dual bound	primal bound	gap
t 0.1s	1	0	0	-	443k	0	-	91	65	0	0	0	0	0	--	1.000000e+002	Inf
b 0.1s	1	0	0	-	507k	0	-	91	65	91	65	0	0	0	--	3.800000e+001	Inf
* 0.1s	1	0	52	-	520k	0	-	91	65	91	65	0	0	0	3.200000e+001	3.200000e+001	0.00
0.1s	1	0	52	-	520k	0	-	91	65	91	65	0	0	0	3.200000e+001	3.200000e+001	0.00

SCIP Status: problem is solved [optimal solution found]
Solving Time (sec): 0.14
Solving Nodes: 1
Primal Bound: +3.20000000000000e+001 (4 solutions)
Dual Bound: +3.20000000000000e+001
Gap: 0.00

The PMU locations are displayed in Fig.11 and in Table 21. This placement solution has the maximum observability indicator [44]-[45]. Table 21 illustrates the PMU positioning sites being derived by optimizing the one-criterion objective function. Table 22 illustrates the process optimization for the multi-objective model using SCIP optimizer function [63]-[66].

As the B&B tree has been constructed, a suitable number of nodes are explored where an LP relaxation is solved at each root-node [8]-[16]. The Primal Bound is the desired cost value for the minimization problem. Finally, an optimality and feasibility criteria has been found. Table 22 produces the log file while Table 23 shows the PMU placement with a maximum observability indicator [59], [62]-[67], [69]-[70].

The Dual Bound closes the gap giving global optimality. In this instance, the SCIP optimizer routine is considered to have delivered a solution with a certificate of 0.00 % optimality [66]. Two competitive trends are satisfied by a global branch & bound in unison. A global optimum point is delivered within a 0.00 % relative gap.

Hence, an enumeration tree is considered in the direction of getting the first incumbent solution that satisfies the observability constraints, reducing the dimension of the B&B tree and finally giving the global one solution.The global optimality is fulfilled with a zero-gap and a relative gap within a pre-specified tolerance tuned by the implementation of an optimization model in the MATLAB environment [62]. SCIP terminates with a desired outcome when the gap between the upper and lower bounds is equal to zero.

More exactly, the primal bound is the upper bound for the minimization problems. On the other hand, the dual cost bound is the lower bound for such kind of optimization problems. The Primal bound is the solution for the minimization problem whereas the Dual Bound is responsible to close the absolute gap [57]-[58]. The solution is the upper bound for the objective function related to the minimization problems whereas the lower bound minimizes the optimality criteria to zero [63]-[65].

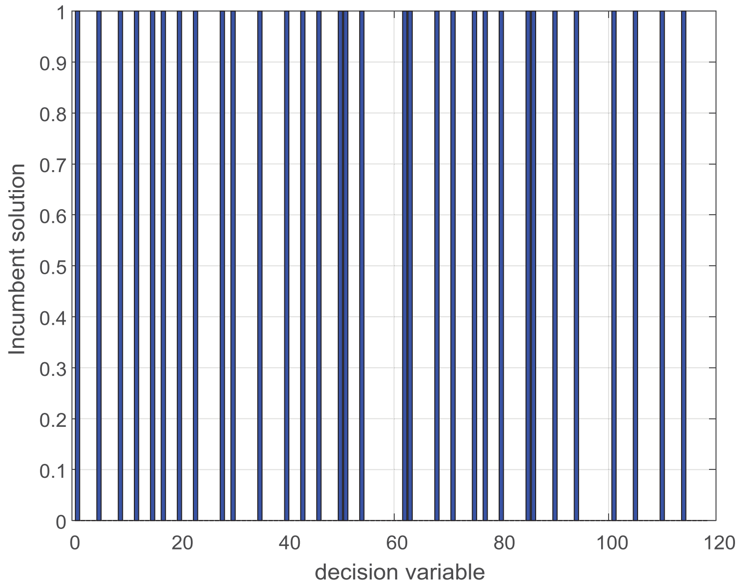

FIGURE 11. Optimal PMU Arrangement

TABLE 21. Optimal PMU Arrangement for the 118-bus Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 150
1, 5, 9, 12, 15, 17, 20, 23, 28, 30, 35, 40, 43, 46, 50, 51, 54, 62, 63, 68, 71, 75, 77, 80, 85, 86, 90 94, 101, 105, 110, 114

SCIP adopts only the interior point optimizer routin Ipopt to linearize and solve the non-convexity constraint function [66]. The discrepancy between the primal and dual bounds is zero which is inside the predefined criterion to determine a global optimal solution [64]. The Primal Bound is the solution whereas the Dual Bound is the culprit to close the gap. Hence, the optimizer solver returns a solution with 0.00 % optimality [63]-[65].

Table 22. Optimization Process: Results of SCIP routine

time	node	left	LP iter	LP it/n	mem	mdpt	frac	vars	cons	cols	rows	cuts	confs	strbr	dual bound	primal bound	gap
t 0.1s	1	0	0	-	443k	0	-	91	65	0	0	0	0	0	--	9.638136e+001	Inf
b 0.1s	1	0	0	-	507k	0	-	91	65	91	65	0	0	0	--	3.428814e+001	Inf
* 0.1s	1	0	47	-	520k	0	-	91	65	91	65	0	0	0	3.061017e+001	3.061017e+001	0.00
0.1s	1	0	47	-	520k	0	-	91	65	91	65	0	0	0	3.061017e+001	3.061017e+001	0.00
SCIP Status: problem is solved [optimal solution found]																	
Solving Time (sec): 0.12																	
Solving Nodes: 1																	
Primal Bound: +3.06101694915254e+001 (4 solutions)																	
Dual Bound: +3.06101694915254e+001																	
Gap: 0.00																	

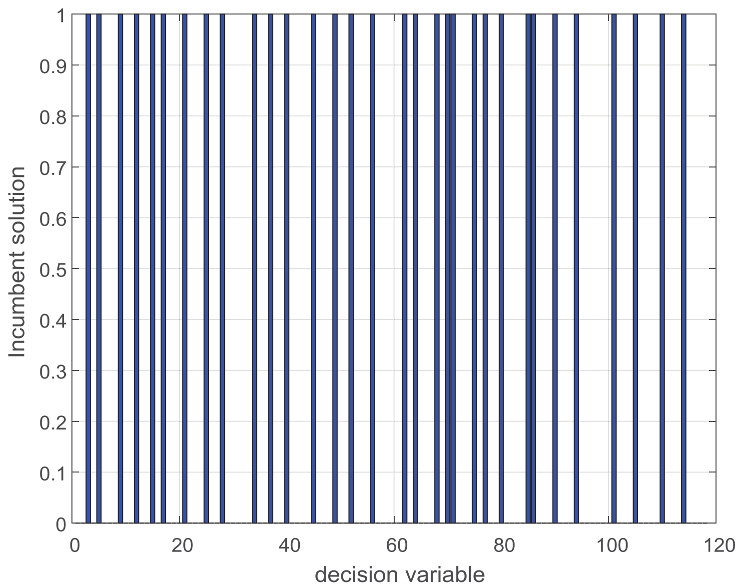

FIGURE 12. Optimal PMU Arrangement

TABLE 23. Optimal PMU Arrangement for the 118-bus Power Networks

PMU Arrangement Locations with Maximum Observability Indicator Equal to 164
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, 101, 105, 110, 114

Table 24 illustrates the optimal solutions derived by solving the $0-1$ integer linear program. For instance, GLPK, SCIP, Intlinprog, LPSOLVE and Gurobi solves the binary integer program [66].

TABLE 24. BEST PMU Arrangement for Standard Power Systems

Intlinprog routine to the binary (Boolean) Constraint Integer Problem		
IEEE bus system	Single-Cost Function	Aggregate-Cost Function
14-bus	2, 8, 10, 13	2, 6, 7, 9
30-bus	1, 5, 8, 10, 11, 12, 19, 23, 26, 29	2, 4, 6, 9, 10, 12, 15, 20, 25, 27
57-bus	1, 4, 9, 20, 23, 27, 29, 30, 32, 36, 38, 41 45, 46, 50, 54, 57	1, 4, 6, 9, 15, 20, 24, 28, 30, 32, 36, 38, 41, 46 51, 53, 57
118-bus	2, 5, 10, 12, 15, 17, 21, 25, 29, 34, 37, 41 ,45, 49, 53, 56, 62, 64, 72, 73, 75, 77, 80, 85, 87, 91, 94, 101, 105, 110, 114, 116	3, 5, 9, 12, 15, 17, 21, 25, 29, 34, 37, 40, 45 49, 53, 56, 62, 64, 68, 70, 71, 75, 77, 80, 85, 86 91, 94, 101, 105, 110, 114
SCIP routine to the binary (Boolean) Constraint Integer Problem		
IEEE bus system	Single-Cost Function	Aggregate-Cost Function
14-bus	2, 7, 11, 13	2, 6, 7, 9
30-bus	1, 2, 6, 9, 10, 12, 15, 18, 25, 27	2, 4, 6, 9, 10, 12, 15, 18, 25, 27
57-bus	1, 4, 6, 9, 20, 24, 25, 28, 32, 36, 38, 39 41, 44, 46, 51, 53	1, 4, 6, 9, 15, 20, 24, 25, 28, 32, 36, 38, 39, 41 46, 50, 53
118-bus	1, 5, 9, 12, 15, 17, 20, 23, 28, 30, 35, 40 43, 46, 50, 51, 54, 62, 63, 68, 71, 75, 77 80, 85, 86, 90, 94, 101, 105, 110, 114	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, 101, 105, 110, 114

TABLE 24. BEST PMU Arrangement for Standard Power Systems (continued)

GLPK routine to the binary (Boolean) Constraint Integer Problem		
IEEE bus system	Single-Cost Function	Aggregate-Cost Function
14-bus	2, 6, 7, 9	2, 6, 7, 9
30-bus	1, 5, 8, 10, 11, 12, 15, 18, 25, 27	2, 4, 6, 9, 10, 12, 15, 18, 25, 27
57-bus	1, 3, 6, 13, 19, 22, 25, 27, 32, 36, 39, 41 44, 47 51, 52, 55	1, 4, 6, 9, 15, 20, 24, 25, 28, 32, 36, 38, 39, 41 46, 50, 53
118-bus	2, 5, 10, 11, 12, 17, 21, 24, 25, 29 34, 37 40, 45, 49, 52, 56, 62, 63 68, 73, 75, 77 80, 85, 86, 90, 94 101, 105, 110, 114	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, 101, 105, 110, 115
lpsolve routine to the binary (Boolean) Constraint Integer Problem		
IEEE bus system	Single-Cost Function	Aggregate-Cost Function
14-bus	2, 6, 8, 9	2, 6, 7, 9
30-bus	1, 2, 6, 9, 10, 12, 15, 18, 25, 27	2, 4, 6, 9, 10, 12, 15, 20, 25, 27
57-bus	1, 4, 6, 9, 15, 20, 24, 25, 28, 32, 36, 38 39, 41, 46, 51, 53	1, 4, 6, 9, 15, 20, 24, 25, 28, 32, 36, 38, 39 41, 46, 50, 53
118-bus	1, 5, 9, 11, 12, 17, 20, 23, 25, 28, 34, 37 40, 45, 49, 52, 56, 62, 63, 68, 71, 75, 77 80, 85, 86, 90, 94, 101, 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 91, 94, 101, 105, 110, 114
Gurobi routine to the binary (Boolean) Constraint Integer Problem		
IEEE bus system	Single-Cost Function	Aggregate-Cost Function
14-bus	2, 7, 10, 13	2, 6, 7, 9
30-bus	1, 5, 6, 9, 10, 12, 15, 19, 25, 27	2, 4, 6, 9, 10, 12, 15, 19, 25, 27
57-bus	2, 6, 12, 19, 22, 25, 27, 29, 32, 36, 39, 41 45 46, 49, 51, 54	1, 4, 6, 9, 15, 20, 24, 25, 28, 32, 36, 38, 39, 41 47, 50, 53
118-bus	1, 5, 9, 12, 15, 17, 21, 23, 28, 30, 34, 37 40 45, 49, 52, 56, 62, 63, 68, 71, 75, 77 80, 85 86, 91, 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, 91 94, 102, 105, 110, 114

The $0-1$ polynomial constraint model can be solved by using a s-BBA implemented in the SCIP optimizer routine [63]-[65]. The SCIP optimizer routine uses a spatial branch-and-bound algorithm to solve (MI) nonlinear programs [63]-[65]. SCIP needs an external LP routine to implement LP relaxations, which are required to be given at compilation computational time [63]-[65].

SCIP uses Soplex by default for that purpose [63]-[65]. SCIP performs well the process optimization in the YALMIP jointly with MATLAB platform [62], develops the B&B tree, pruning infeasible regions and solving the problem with reasonable LP relaxations at a number of nodes calculated for this optimality purpose [63]-[66].

The aim is to linearize the constraint function and binary restriction decision variable. The constraint function and performance operations are considered at a given B&B tree node. The enumeration B&B tree is composed by nodes and leaves [8]-[14], [63]-[66].

The leaves are needed in the process optimization. The B&B tree is developed by branching procedures on fractional integer variables or variables in non-convex constraints being violated, using primal heuristics and reformulation through presolving [8]-[14], [63]-[66]. The strategy adopted in optimization constructs linear programming problems by relaxing the binary restriction of the decision variables and convexifying the non-convexity form of the constraint function [63]-[65].

The LP relaxation procedure consists of relaxing the binary restriction $\vec{x} \epsilon \{0, 1\}$, transforming the non-convexities by convexification and linearization of nonlinear functions which are also convexity functions [8]-[16]. For the specific problem, the presolve procedure is performed as a short process towards getting a certificate of optimality [8]-[14], [63]-[70].

The primal bound gives the solution for the minimization problem within a suboptimality criterion (%) whereas the dual bound is culprit to minimize this criteria and the solution to be the global one [63]-[65]. SCIP terminates the optimization process and displays the optimal solution within a zero-gap optimality [63]-[65]. SCIP produces optimal results for the minimization of one and two criteria in two successive phases. The optimal solution is found to be inside a zero relative gap. The percentage relative gap is defined as follows [63]-[65]:

$$\mathit{percentage}\ \text{relative gap} = 100 * |\text{primal} - \text{dual}|/\text{MIN}(|\text{dual}|, |\text{primal}|) \quad (26)$$

Tables 25 & 29 illustrate the number of nodes of the particular run of the s-BBA embedded in the SCIP optimizer routine [63]-[65]. Then, the whole optimization process terminates with the optimal solution and the Dual Bound closes the gap [62]-[69]. With the aspect to the simulation results, it has been illustrated that the Primal bound of the B&B tree for minimization purposes is considered to be the solution of the objective function within a sub-optimality criteria. The Dual bound closes the gap. The upper bound on the cost function value is the optimal.

SCIP optimizer overcomes the phenomenon of trapping at a local minimum point giving the global optimal solution within an absolute Gap-tolerance equal to zero [59]-[60], [62]-[70].

Table 25 presents the log file produced by the SCIP optimizer routine to display the desired outcome. An amount of nodes are explored where the LP relaxations are solved, the discrepancy between the dual bound and primal bound is constantly calculated and finally an optimal solution is calculated with a zero-gap tolerance [64]-[65].

Tables 25 & 29 & 31 & 33 illustrate the efficiency of the global BBA to come across an optimal solution within a zero-gap tolerance. Therefore, a notable output is its capability to succeed at a fact convergence rate for achieving global optimum point. A B&B enumeration tree has been built where limited number of nodes are explored to solve this combinatorial problem [8]-[16]. Table 26 illustrates the PMU positioning sites whereas those placement sites are dislpayed in Fig.13.

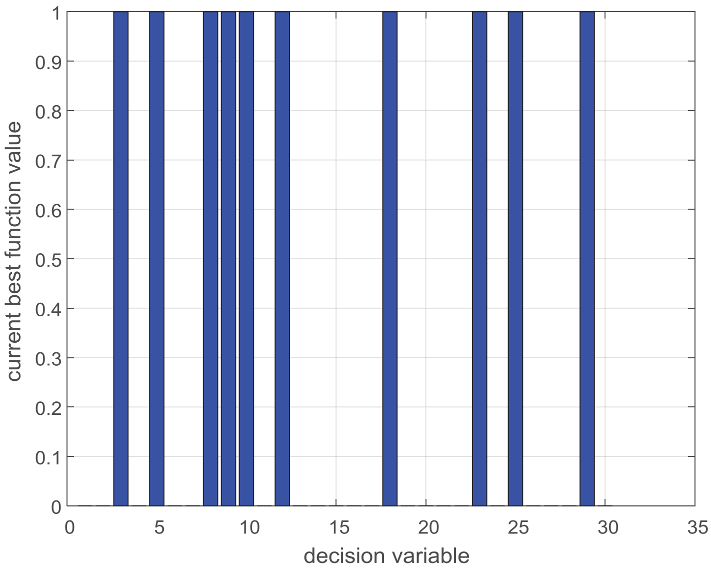

FIGURE 13. Optimal PMU Arrangement Achieved by SCIP Optimizer

Table 25. Optimization Process: Results of SCIP optimizer routine

```
time | node | left |LP iter|LP it/n| mem |mdpt |frac |vars |cons |cols |rows |cuts |confs|strbr | dual bound | primal bound | gap
T 0.7s |   1 |   0|    0 |    - |3623k|  0 |  -  | 422 | 630 | 422 | 886|  0 |  0 |  0 |  --  |2.000000e+001 |  Inf
  0.7s |   1|   0|   85 |    - |3619k|  0 | 42 | 422 | 630 | 422 | 886 |  0 |  0 |  0 |8.250000e+000 |2.000000e+001 | 142.42
U 0.8s|  1 |   0|   85 |    - |3642k|  0 | 42 | 422 | 630 | 422 | 886 |  0 |  0 |  0 |8.250000e+000 |1.000000e+001 | 21.21
  0.9s |   1 |  0| 104 |   -|4185k|  0 |  - | 422 | 631 | 422 | 972 | 86  |  0 |  0 |1.000000e+001 |1.000000e+001 |  0.00
  0.9s |   1 |  0| 104 |   -|4185k|  0 |  - | 422 | 631 | 422 | 972 | 86  |  0 |  0 |1.000000e+001 |1.000000e+001 |  0.00
```

SCIP Status: problem is solved [optimal solution found]
Solving Time (sec): 0.86
Solving Nodes: 1

Primal Bound: +1.00000000000000e+001 (2 solutions)
Dual Bound: +1.00000000000000e+001
Gap: 0.00
Elapsed time is 0.938705 seconds.

sol =

struct with fields:

yalmipversion: '20210331'

matlabversion: '9.4.0.813654 (R2018a)'

yalmiptime: 0.3288

solvertime: 0.6072

info: 'Successfully solved (SCIP-NL)'

problem: 0

ans =

```
  Columns 1 through 14

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

  Columns 15 through 28
```

Table 25. Optimization Process: Results of SCIP optimizer routine (continued)

```
    0    0    0    1    0    0    0    0    1    0    1    0    0    0

  Columns 29 through 30

    1    0
```

Linear scalar (real, binary, 30 variables)
Current value: 10
Coefficients range: 1 to 1

```
SORI =

    39

ans =

    3    5    8    9   10   12   18   23   25   29
```

percentage relative gap $= 100 * |\text{primal} - \text{dual}|/\text{MIN}(|\text{dual}|, |\text{primal}|)$; termination criteria: Gap $= 0$ or Solving Time=0.86 s ; [63]-[66].

BBA adopts a tree search strategy to unquestioningly count all solutions that can exist to the given minimization model, spreading pruning rules to get rid of regions of the search space that cannot give a high quality solution point [8]-[14]. SCIP utilizes three strategies to build the binary tree such as search and branching strategies and rules to prune infeasible regions to find a feasible and optimum point at the same time [63]-[66].

We conclude that if the binary tree is small in size, the solving process is going to be fast [8]-[16]. That enumeration tree is solved without computation complexity as the algorithmic scheme's experimental outcome illustrates. That optimal result consists of a desired outcome in the direction of getting a feasible and global solution in a robust manner without computation burden [8]-[14]. An optimal solution is produced without the optimization to be computational heavy in the process. The solving time is 0.86 s as found jn the above log output [63]-[66].

Furthermore, the objective function is found to be equal at the lowest value estimated by the branch-and-bound tree iterated in the log file for the pure optimization [15]. The simulations validate the proposed program related to algorithms' results and the involved calculation time [70].

Table 25 displays the results where a simulation run is considered as successful because the Primal and Dual Bounds are found to be equal. We understand that in the status log file, the primal and dual bound are equal, hence, the optimization problem has been solved [63]-[66]. Therefore, SCIP leads quickly towards a solution within a 0.00 % optimality criterion [8]. SCIP results in a solution equal to the Primal Bound for the minimization problem where the Dual Bound closes the gap. In such a case, it is considered that the algorithm produces a certificate of optimality [8]-[14].

TABLE 26. Optimal PMU Arrangement for the 30-bus Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 39
3, 5, 8, 9, 10, 12, 18, 23, 25, 29

A notable remark is that we count the measuring times by which a power network node is monitored either directly by a PMU at that node or indirectly by PMUs selected at adjacency buses. Thus, a novel framework for the nonlinear model is proposed in the SCIP for achieving global optimality avoiding being trapped into a local solution point [63]-[65].

The objective function with two products results in a single-function for optimizing in a nonlinear framework with binary decision variables. This single-cost function is stated with a weight element by which each decision variable has its priority in the multi-criteria optimization shown in Table 27.

TABLE 27. Measuring Times by Which a Power Network Bus is Monitored

Columns 1 through 17																
3	5	3	5	3	8	3	3	4	7	2	6	2	3	5	3	3

Columns 18 through 30												
3	3	3	3	4	3	4	4	2	5	4	3	3

The weights are shown in Table 28. This table signifies the weights reflecting the significance of each variable for the purpose of getting that solution with maximum observability indicator [18].

TABLE 28. Weights Defined in the Multi-Objective Function

Columns 1 through 8							
0.9000	0.8333	0.9000	0.8333	0.9000	0.7333	0.9000	0.9000

Columns 9 through 16							
0.8667	0.7667	0.9333	0.8000	0.9333	0.9000	0.8333	0.9000

Columns 17 through 24							
0.9000	0.9000	0.9000	0.9000	0.9000	0.8667	0.9000	0.8667

Columns 25 through 30					
0.8667	0.9333	0.8333	0.8667	0.9000	0.9000

Table 29. Optimization Process: Results of SCIP optimizer routine

```
time | node | left |LP iter|LP it/n| mem |mdpt |frac |vars |cons |cols |rows |cuts |confs|strbr| dual bound   | primal bound | gap
T 0.8s|   1 |    0 |    0 |      - |3625k | 0 |   - | 422 | 630 | 422 | 886 |   0   |   0 |   0 |     --              |1.720000e+001 |   Inf
  0.8s |    1 |    0 |  112|    - |3622k |   0 | 60 | 422 | 630 | 422 | 886 |   0   |   0 |   0 |6.833333e+000 |1.720000e+001  | 151.71
U 0.9s|   1 |    0 |  112 |   - |3647k|   0 | 60 | 422 | 630 | 422 | 886 |   0   |   0 |   0 |6.833333e+000 |8.266667e+000  | 20.98
  0.9s|     1 |    0 |  169 |   - |4018k|   0 |   - | 422 | 631 | 422 |1030 | 144 |   0 |   0 |8.266667e+000 |8.266667e+000 |   0.00
  0.9s|     1 |    0 |  169 |   - |4018k|   0 |   - | 422 | 631 | 422 |1030 | 144 |   0 |   0 |8.266667e+000 |8.266667e+000 |   0.00
SCIP Status: problem is solved [optimal solution found]
Solving Time (sec): 0.92
Solving Nodes: 1
Primal Bound: +8.26666666666667e+000 (2 solutions)
Dual Bound:    +8.26666666666667e+000
Gap: 0.00
Elapsed time is 1.798749 seconds.
ans =
  Columns 1 through 14

   0    1    0    1    0    1    0    0    1    1    0    1    0    0
  Columns 15 through 28
   1    0    0    1    0    0    0    0    0    0    1    0    1    0
  Columns 29 through 30
   0    0
Linear scalar (real, binary, 30 variables)
Current value: 8.2667
Coefficients range: 0.73333 to 0.93333
SORI = 52
```

$\text{percentage relative gap} = 100 * |\text{primal} - \text{dual}|/\text{MIN}(|\text{dual}|, |\text{primal}|)$; Termination criteria: $\text{Gap} = 0$ and Solving Time=1.798749s; [63]-[66].

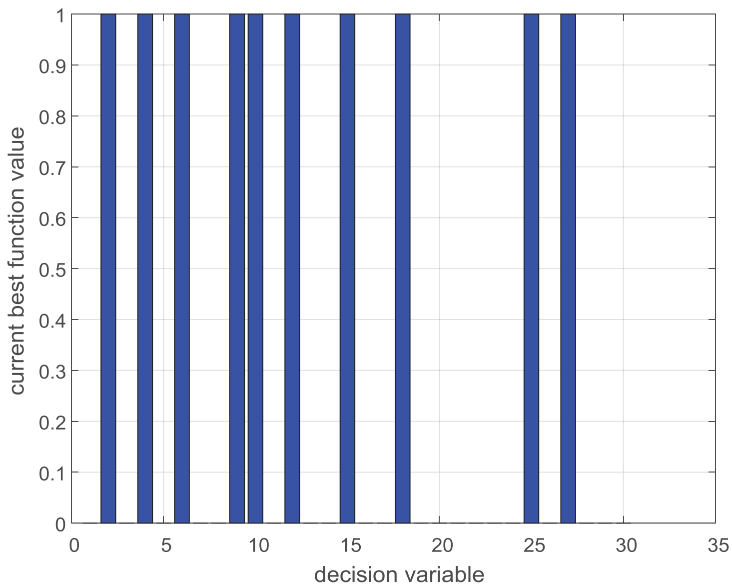

FIGURE 14. Optimal PMU Arrangement Achieved by SCIP Optimizer Routine

TABLE 30. Optimal PMU Arrangement for the 118 Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 52
2, 4, 6, 9, 10, 12, 15, 18, 25, 27

The two-criterion constraint nonlinear program is implemented with SCIP optimizer and the implementation of it is given in Tables 31& 33. After beginning the optimization process, SCIP will primarily take a few rounds of the presolving process and after that the s-BBA will explore the region constituted by feasible solutions in the direction of getting the optimality [63]-[65].

The presolving process helps a lot to reformulate the initial model. Then, we solve the proposed model with SCIP optimizer routine to ensure globally optimality [63]-[65].

Tables 31&33 illustrate the number of nodes spent during the binary tree implementation, the elapsed time and the primal and dual bounds. The difference of them goes to zero ensuring that a global optimal solution is achieved [8]-[16]. The optimal solutions are illustrated in Tables 31&33.

As can be observed, the linear programming relaxations are solved at specific nodes without being time consuming during the iterative process [8]-[16]. Therefore, the optimal solution is reached by solving a small number of LP relaxations at the root nodes exploring the optimizer function [65].

The Primal Bound is the solution for the minimization while the lower bound closes the gap declaring that the global certificate of optimality has been achieved [63]-[65].

The Primal and Dual Bounds are computed, the relative gap between those levels and an optimality is validated when the algorithm terminates the entire process to a global ϵ-optimality. [64]. The relative gap was found equal to zero giving the global optimality [14]-[15].

The $0-1$ polynomial problem has been optimized globally in a fast computational way as shown in the log output of Table 31 [70]. Table 31 shows the entire optimization process to a global ϵ-optimality. The Primal Bound is the solution on the objective function and the globality is guaranteed because the Dual Bound has posed a lower limit in the whole process.

TABLE 31. Optimization Process: Results of SCIP optimizer routine

time	node	left	LP iter	LP it/n	mem	mdpt	frac	vars	cons	cols	rows	cuts	confs	strbr	dual bound	primal bound	gap
T 2.2s	1	0	0	-	26M	0	-	3780	4489	3780	7029	0	0	0	--	8.600000e+001	Inf
2.3s	1	0	362	-	26M	0	205	3780	4489	3780	7029	0	0	0	2.500000e+001	8.600000e+001	244.00
U 2.6s	1	0	362	-	27M	0	205	3780	4489	3780	7029	0	0	0	2.500000e+001	3.300000e+001	32.00
2.8s	1	0	906	-	28M	0	211	3780	4490	3780	7427	398	0	0	3.092857e+001	3.300000e+001	6.70
2.9s	1	0	906	-	27M	0	211	3780	4490	3780	6488	398	0	0	3.092857e+001	3.300000e+001	6.70
3.1s	1	0	1358	-	28M	0	167	3780	4490	3780	7066	976	0	0	3.173333e+001	3.300000e+001	3.99
3.1s	1	0	1358	-	26M	0	167	3780	4490	3780	3331	976	0	0	3.173333e+001	3.300000e+001	3.99
3.4s	1	0	1642	-	20M	0	54	3780	1752	3780	3072	1319	0	0	3.200000e+001	3.300000e+001	3.13
3.4s	1	0	1642	-	20M	0	54	3780	1752	3780	2932	1319	0	0	3.200000e+001	3.300000e+001	3.13
3.6s	1	0	1759	-	22M	0	80	3780	1709	3780	2935	1397	0	0	3.200000e+001	3.300000e+001	3.13
3.8s	1	0	1785	-	25M	0	36	3780	1709	3780	3001	1463	0	0	3.200000e+001	3.300000e+001	3.13
4.2s	1	0	1823	-	33M	0	62	3780	1709	3780	3015	1477	0	0	3.200000e+001	3.300000e+001	3.13
4.5s	1	0	1825	-	38M	0	0	3780	1709	3780	3039	1501	0	0	3.200000e+001	3.300000e+001	3.13
* 4.5s	1	0	1825	-	38M	0	-	3780	1709	3780	3039	1501	0	0	3.200000e+001	3.200000e+001	0.00

SCIP Status: problem is solved [optimal solution found]
Solving Time (sec): 4.54
Solving Nodes: 1
Primal Bound: +3.20000000000000e+001 (3 solutions)
Dual Bound: +3.20000000000000e+001
Gap: 0.00
Elapsed time is 16.228007 seconds.

percentage relative gap $= 100 * |\mathrm{primal} - \mathrm{dual}|/\mathrm{MIN}(|\mathrm{dual}|, |\mathrm{primal}|)$; Termination criteria: $\mathrm{Gap} = 0$ and Solving Time=4.54 s ;.

SCIP attends to produce a small sized branch-and-bound tree without producing many branching decisions. Therefore, a powerful branching regulation is performed that leads to optimality [14]. The solver determines the Primal and Dual bounds where it terminates with the best solution [64].

Thus, an optimal solution is attained within a zero-gap tolerance; a global solution was discovered within a 0.00 % optimality criterion. SCIP successfully closes the optimality gap. Gap zero means that no better possible solution can be found by far [70]. Therefore, SCIP returns a global optimal solution and the problem has been solved exactly avoiding being trapped into a local solution or a sub-optimal solution [63]-[65].

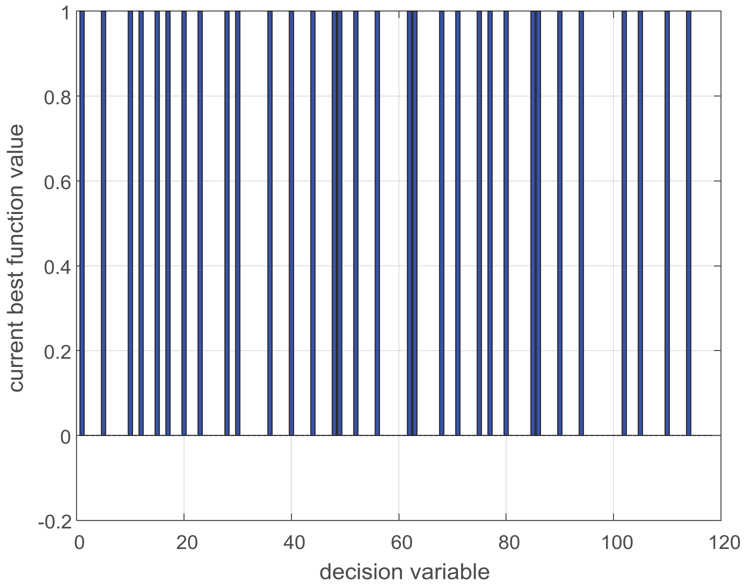

FIGURE 15. Optimal PMU Arrangement Achieved by SCIP Optimizer Routine

As observed, for the second case study, SCIP delivers optimal results with equal quality and quantity as those found in [23]. The optimal sites are shown in Table 32. An outcome is produced by the SCIP and it is a feasible optimal solution for the initial model at the same time [63]-[66].

TABLE 32. Optimal PMU Arrangement for the 118-bus Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 154
1, 5, 10, 12, 15, 17, 20, 23, 28, 30, 36, 40, 44, 48, 49, 52, 56, 62, 63, 68, 71, 75, 77, 80, 85, 86, 90 94, 102, 105, 110, 114

The optimization process is efficiently executed and shown in the log file illustrated in the Table 33. During the iterative process, the Primal and Dual bounds are found to be equal in quantity [70].

Soplex is an LP solver suitable to solve linear programming relaxations towards optimality. SoPlex solves the relaxed problems to find an optimum solution point. The difference of those bounds delivers the global certificate of optimality [63]-[66].

For the proposed minimization problem, the Primal bound is the best possible solution whereas the Dual bound is culprit to close the absolute gap. Hence, the optimality is achieved within a 0.00% gap tolerance [45], [63]-[66]. That stopping tolerance is an adequate criterion because it says that no better solution can be found by the overall optimization.

The optimal solution has a maximum observability indicator, that is, equal to $SORI = 164$. The elapsed computational time is reasonably spent considering the dimension of the optimization problem, the transformation of the polynomial problem into a polyhedral approximation and its computational complexity [13].

Table 33 presents the log file produced by SCIP where the optimization process is displayed. The optimal solution is illustrated in Fig.16 and the PMU locations are shown in Table 34.

TABLE 33. Optimization Process: Results of SCIP optimizer routine

time	node	left	LP iter	LP it/n	mem	mdpt	frac	vars	cons	cols	rows	cuts	confs	strbr	dual bound	primal bound	gap
T 2.4s	1	0	0	-	26M	0	-	3780	4489	3780	7029	0	0	0	--	8.300000e+001	Inf
2.6s	1	0	344	-	26M	0	202	3780	4489	3780	7029	0	0	0	2.392373e+001	8.300000e+001	246.94
U 3.0s	1	0	344	-	27M	0	202	3780	4489	3780	7029	0	0	0	2.392373e+001	3.063559e+001	28.06
3.3s	1	0	670	-	28M	0	149	3780	4490	3780	7489	460	0	0	2.944504e+001	3.063559e+001	4.04
3.3s	1	0	670	-	27M	0	149	3780	4490	3780	6509	460	0	0	2.944504e+001	3.063559e+001	4.04
3.6s	1	0	920	-	28M	0	65	3780	4490	3780	6836	787	0	0	3.051332e+001	3.063559e+001	0.40
3.6s	1	0	920	-	25M	0	65	3780	4490	3780	2574	787	0	0	3.051332e+001	3.063559e+001	0.40
3.8s	1	0	977	-	18M	0	27	3780	1372	3780	2063	871	0	0	3.061017e+001	3.063559e+001	0.08
3.8s	1	0	977	-	17M	0	27	3780	1372	3780	1831	871	0	0	3.061017e+001	3.063559e+001	0.08
3.9s	1	0	997	-	18M	0	35	3780	1158	3780	1786	888	0	0	3.061017e+001	3.063559e+001	0.08
4.0s	1	0	1015	-	19M	0	32	3780	1158	3780	1831	933	0	0	3.061017e+001	3.063559e+001	0.08
4.0s	1	0	1021	-	20M	0	33	3780	1158	3780	1839	941	0	0	3.061017e+001	3.063559e+001	0.08
4.2s	1	0	1060	-	22M	0	43	3780	1158	3780	1848	950	0	0	3.061017e+001	3.063559e+001	0.08
4.3s	1	0	1082	-	24M	0	41	3780	1158	3780	1871	973	0	0	3.061017e+001	3.063559e+001	0.08
4.4s	1	2	1251	-	24M	0	41	3780	1158	3780	1871	973	0	17	3.061017e+001	3.063559e+001	0.08

transformed objective value is always integral (scale: 0.00847457627118644)

Presolving Time: 2.36

time	node	left	LP iter	LP it/n	mem	mdpt	frac	vars	cons	cols	rows	cuts	confs	strbr	dual bound	primal bound	gap
4.7s	1	0	1404	-	10M	0	10	291	529	291	478	0	0	17	3.061017e+001	3.063559e+001	0.08
4.7s	1	0	1404	-	10M	0	10	291	529	291	471	0	0	17	3.061017e+001	3.063559e+001	0.08
4.7s	1	0	1437	-	10M	0	29	291	500	291	504	44	0	17	3.061017e+001	3.063559e+001	0.08
E 4.7s	1	0	1437	-	10M	0	-	291	500	291	504	44	0	34	3.061017e+001	3.061017e+001	0.00

SCIP status: problem is solved [optimal solution found]

Solving Time (sec): 4.74

Solving Nodes: 1 (total of 2 nodes in 2 runs)

Primal Bound: +3.06101694915254e+001 (3 solutions)

Dual Bound: +3.06101694915254e+001

Gap: 0.00

Elapsed time is 32.601301

$\text{percentage relative gap} = 100 * |\text{primal} - \text{dual}|/\text{MIN}(|\text{dual}|, |\text{primal}|)$; Termination criteria: $\text{Gap} = 0$ and Solving Time=4.74 s ;

TABLE 34. Optimal PMU Arrangement for the 118-bus Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 164
3, 5, 9, 12, 15, 17, 21, 23, 29, 30, 34, 37, 40, 45, 49, 52, 56, 62, 64, 68, 71, 75, 77, 80, 85, 86, 90, 94 101, 105, 110, 115

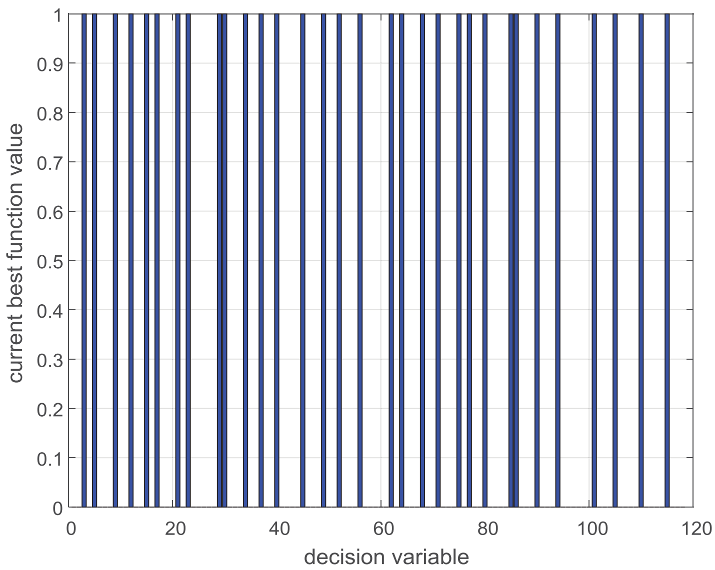

FIGURE 16. Optimal PMU Arrangement achieved by SCIP Optimizer Routine

SCIP optimization function performs the binary polynomial optimization model for power systems sized by 14 to 300 decision variables reflecting the number of power network buses within a power transmission grid as shown in Table 35 [71]. Hence, the $0-1$ polynomial constraint optimization problem is easy to be solved by spending a trivial quantity involving calculation time. This optimum point can be characterized as a high-quality optimal solution [18], [35]-[38], [41], [42], [44]-[45].

SCIP solver explores the feasible region and gives the desired outcome at one root-node. The algorithm is characterized as a convergent method. The B&B algorithm explores a number of nodes where the relaxed problems are solved towards getting the best possible and feasible solution at the same time. Hence, an optimal solution was found with the dimension of the B&B tree keeping as small as possible. Once the MATLAB code is written, we execute this program to find optimality [62].

It is guaranteed that it is a global one with high probability. The optimum points are displayed in Tables 35-37. The numerical results are found to be a global optimal solution for power network size from small to medium size. The proposed arrangement of PMUs in smart grids was tested on IEEE-300 bus systems. This power network reflects a large-scale optimization problem [24]-[25].

BMIBNB solver supports integer, binary and continuous variables to execute an optimization program [70]. BMIBNB performs non-convex optimization problems, and come across the optimal solution. The BMIBNB optimizer function is performed with efficiency on the IEEE-300 bus system [61], [71]. It detects the global optimal solution for both case studies spending a trivial amount of time considering the dimension size of the optimization problem as illustrated in Tables 38 & 40.

It is confirmed that it can be validated for larger power transmission networks in size [61]. The best resulting arrangement was found to be an economical solution satisfying the two criteria determined for the proposed nonlinear scheme; Optimal Arrangement of PMUs and maximum number of times by which a network bus is monitored either directly or indirectly [18], [44]-[45].

The large-scale optimization model is optimized by the bmibnb function embedded in YALMIP toolbox [70]. Such optimization problem corresponds to simulating the IEEE-300 bus system [61], [71]. The optimization problem includes 300 decision variables with equal number of equality constraints where a binary restriction of the decision variables is defined [61], [71].

TABLE 35. Best PMU Arrangement for Standard Power Systems

SCIP Optimizer Routine to the Binary (Boolean) Optimization Problem		
IEEE bus system	**Single-Cost Function**	**Aggregate-Cost Function**
14-bus	2, 7, 10, 13	2, 6, 7, 9
30-bus	3, 5, 8, 9, 10, 12, 18, 23, 25, 29	2, 4, 6, 9, 10, 12, 15, 18, 25, 27
57-bus	1, 6, 9, 15, 18, 21, 24, 25, 28, 32, 36, 38, 41 46, 50, 53, 57	1, 4, 6, 9, 15, 20, 24, 28, 31, 32, 36, 38, 41 47, 50, 53, 57
118-bus	1, 5, 10, 12, 15, 17, 20, 23, 28, 30, 36 40 44, 48, 49, 52, 56, 62, 63, 68, 71 75, 77 80 85, 86, 90, 94, 102, 105, 110, 114	3, 5, 9, 12, 15, 17, 21, 23, 29, 30, 34, 37, 40 45 49, 52, 56, 62, 64, 68, 71, 75, 77, 80, 85 86, 90 94, 101, 105, 110, 115
300-bus	1, 2, 3, 11, 15, 17, 19, 22, 23, 27, 33, 37 38, 43, 48, 49, 53, 54, 55, 59, 60, 64, 68 69, 71, 73, 76, 80, 86, 89, 93, 96, 98, 99 101, 109, 111, 112, 113, 116, 118, 122 125 132, 135, 139, 141, 152, 157, 160 163, 168 183, 187, 189, 190, 193 196 200, 204, 208 210, 211 215, 216, 217 219, 221, 223, 228 232, 233, 235, 237 240, 242, 251, 265, 267 268, 269, 270 272, 273, 274, 276, 294	1, 2, 3, 11, 12, 15, 17, 19, 22, 23, 25, 27 29 33, 37, 38, 43, 48, 49, 53, 54, 55, 58 59, 60 62 64, 65, 68, 71, 79, 83, 85, 86 88, 89 93 98 99, 101, 103, 109, 111 112, 113 116, 118 119 124, 132, 135, 138 143, 145 152, 157 163, 167 173, 177, 183 187, 189 190, 193, 196, 202, 204 208, 210 211, 213 216, 217, 219 224, 225, 228 267 268, 269 270, 272, 273, 274, 276, 294

TABLE 36. Best Optimal Solution Satisfying One Objective-Criterion Objective Function

IEEE bus system	Best Objective Function Value	Maximum Measurement Indicator
14-bus	4	16
30-bus	10	39
57-bus	17	70
118-bus	32	154
300-bus	87	393

TABLE 37. Best Optimal Solution Satisfying Two Objective-Criterion Objective Function

IEEE bus system	Best Objective Function Value	Maximum Measurement Indicator
14-bus	4	19
30-bus	10	52
57-bus	17	72
118-bus	32	164
300-bus	87	432

As observed, the BBA solved efficiently the polynomial model in the direction of global optimality. The optimum point is the global solution; hence the tolerance gap is zero meaning that it can't be a better solution than this found so far [8]-[15], [63]-[65].

Gurobi and Intlinprog are used as lower bounds solvers to count the lower bound and take care of branching as linear programming solvers in the optimization [62], [67]. We reduce the elapsed time by setting 'bmibnb.lpreduce' to zero [68], [70].

We consider the IEEE-300 bus system for implementation to show the optimizer routine's efficiency to get the optimality [61], [71]. Both case studies are illustrated in Tables 38 & 40 whereas the optimal PMU positioning sites are shown in Fig 17 and 18 using Gurobi as an ILP solver. Tables 38 & 40 illustrate the entire process in optimization using the Gurobi solver [67]. The optimal results are obtained by testing the proposed model on the IEEE-300 bus network [61], [71].

Table 38. Optimization Process: Results of bmibnb routine

++
* 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: 300
* -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 : 8.70000E+01 0.00 8.70000E+01 2 225s Solution found by heuristics
* Finished. Cost: 87 (lower bound: 87, relative gap 1.1364e-10%)
* Termination with relative gap satisfied
* Timing: 22% spent in upper solver (2 problems solved)
* 1% spent in lower solver (1 problems solved)
* 6% spent in LP-based domain reduction (600 problems solved)
* 1% spent in upper heuristics (1 candidates tried)
sol =
struct with fields:
yalmipversion: '20210331'
matlabversion: '9.4.0.813654 (R2018a)'
yalmiptime: 0.7623
solvertime: 225.5867
info: 'Successfully solved (BMIBNB)'
problem: 0
Elapsed time is 226.423003 seconds.
ans =

Journal of Physics: Conference Series **2701** (2024) 012001 doi:10.1088/1742-6596/2701/1/012001

Table 38. Optimization Process: Results of bmibnb routine (continued)

Columns 1 through 16															
1	1	1	0	0	0	0	0	0	0	1	0	1	0	1	0
Columns 17 through 32															
0	0	0	0	0	1	1	0	1	0	1	0	0	1	0	0
Columns 33 through 48															
1	0	0	0	1	1	0	0	1	0	1	0	0	0	0	1
Columns 49 through 64															
1	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1
Columns 65 through 80															
0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
Columns 81 through 96															
0	0	0	0	0	1	0	1	0	0	0	0	1	0	0	1
Columns 97 through 112															
0	1	1	0	1	0	0	0	0	0	0	0	1	0	1	1
Columns 113 through 128															
1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	1
Columns 129 through 144															
0	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0
Columns 145 through 160															
1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
Columns 161 through 176															
0	0	0	1	0	0	0	1	0	0	1	0	0	0	0	0
Columns 177 through 192															
0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0

Table 38. Optimization Process: Results of bmibnb routine (continued)

Columns 193 through 208

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

Columns 209 through 224

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

Columns 225 through 240

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

Columns 241 through 256

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

Columns 257 through 272

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

Columns 273 through 288

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

Columns 289 through 300

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

Linear scalar (real, binary, 300 variables)
Current value: 87
Coefficients range: 1 to 1

SORI =

372

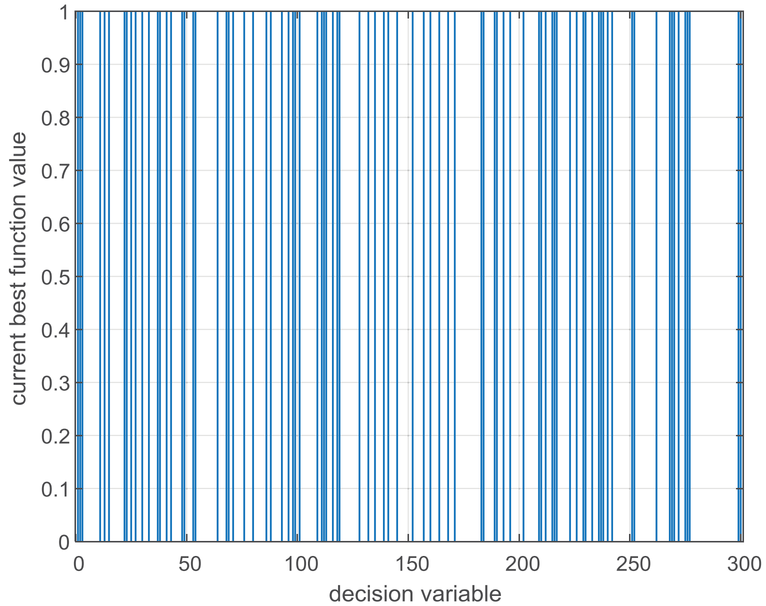

FIGURE 17. Optimal PMU Arrangement achieved by BMIBNB/GUROBI/FMINCON/ Routines

TABLE 39. Optimal PMU Arrangement for the 300-bus Power Network

PMU Arrangement Locations with Observability Indicator Equal to 372
1, 2, 3, 11, 13, 15, 22, 23, 25, 27, 30, 33, 37, 38, 41, 43, 48, 49, 53, 54, 64, 68, 69, 71, 76, 80, 86 88, 93, 96, 98, 99, 101, 109, 111, 112, 113, 116, 118, 119, 128, 132, 135, 139, 141, 145, 152, 157 160, 164, 168, 171, 183, 184, 189, 190, 193, 196, 202, 209, 210, 212, 215, 216, 217, 223, 226 229, 230, 233, 236, 237, 238, 240, 242, 251, 252, 262, 268, 269, 270, 272, 275, 276, 277, 299, 300

Table 40 illustrated the experimental result for testing the two-criterion objective function being optimized under conflicting trends (minimize/maximize). To present the algorithmic idea, we illustrate the log file for the minimization problem in Table 40. The solver returns a solution within a 0.00% criterion. Thus, a validation is given for minimizing a large-scale optimization problem [61].

Table 40. Optimization Process: Results of bmibnb routine

+++
* 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: 300
* -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 : 8.55600E+01 0.00 8.55600E+01 2 231s Solution found by heuristics

Journal of Physics: Conference Series **2701** (2024) 012001 doi:10.1088/1742-6596/2701/1/012001

Table 40. Optimization Process: Results of bmibnb routine (continued)

* Finished. Cost: 85.56 (lower bound: 85.56, relative gap 1.1553e-10%)
* Termination with relative gap satisfied
* Timing: 22% spent in upper solver (2 problems solved)
*   1% spent in lower solver (1 problems solved)
*   7% spent in LP-based domain reduction (600 problems solved)
*   1% spent in upper heuristics (1 candidates tried)
sol =
struct with fields:
yalmipversion: '20210331'
matlabversion: '9.4.0.813654 (R2018a)'
yalmiptime: 0.3034
solvertime: 231.0496
info: 'Successfully solved (BMIBNB)'
problem: 0
Elapsed time is 231.359541 seconds.
ans =
Columns 1 through 16
1 1 1 0 0 0 0 0 0 0 1 1 0 0 1 0
Columns 17 through 32
1 0 1 0 0 1 1 0 1 0 1 0 1 0 0 0
Columns 33 through 48
1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1
Columns 49 through 64
1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 1
Columns 65 through 80
1 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0
Columns 81 through 96
0 0 1 0 1 1 0 1 1 0 0 0 1 0 0 0
Columns 97 through 112

Table 40. Optimization Process: Results of bmibnb routine (continued)

0	1	1	0	1	0	1	0	0	0	0	0	1	0	1	1
Columns 113 through 128															
1	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0
Columns 129 through 144															
0	0	0	1	1	0	0	0	0	1	0	0	0	0	1	0
Columns 145 through 160															
1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
Columns 161 through 176															
0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0
Columns 177 through 192															
1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0
Columns 193 through 208															
1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
Columns 209 through 224															
0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	1
Columns 225 through 240															
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Columns 241 through 256															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Columns 257 through 272															
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1
Columns 273 through 288															
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Columns 289 through 300															
0	0	0	0	0	1	0	0	0	0	0	0				

Hence, the original model is transformed into a polyhedral approximation results in an optimal solution within a zero-gap and meanginless percenatage relative gap [15]. Log files produced by the optimizer routine shows the optimality [14]. Figures 18 and 19 show each one diagram derived by the optimization process whereas the optimal results are tabulated in 41, 43 & 45.

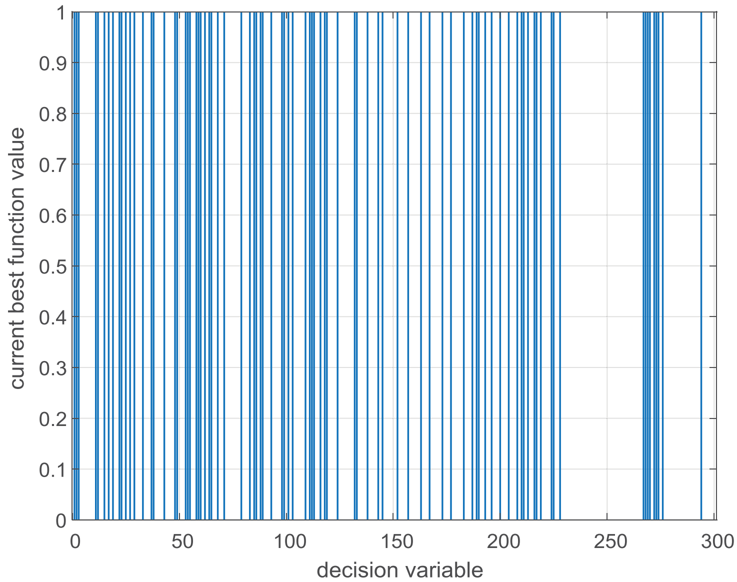

FIGURE 18. Optimal PMU Arrangement achieved by BMIBNB/GUROBI/FMINCON/ Routine

TABLE 41. Optimal PMU Arrangement for 300-bus Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 432
1, 2, 3, 11, 12, 15, 17, 19, 22, 23, 25, 27, 29, 33, 37, 38, 43, 48, 49, 53, 54, 55, 58, 59, 60, 62
64, 65, 68, 71, 79, 83, 85, 86, 88, 89, 93, 98, 99, 101, 103, 109, 111, 112, 113, 116, 118, 119
124, 132, 133, 138, 143, 145, 152, 157, 163, 167, 173, 177, 183, 187, 189, 190, 193, 196, 200, 204
208, 210, 211, 213, 216, 217, 219, 224, 225, 228, 267, 268, 269, 270, 272, 273, 274, 276, 294

Tables 42 & 44 shows the whole optimization is derived by the Intlinprog solver included in the MATLAB optimization library [62]. The upper bound on the cost function is the global solution due to the fact that the lower bound closes the gap. Therefore, the solution is achieved within a Gap (%) equal to zero. The optimizer function is invoked within the MATLAB optimization library which interacts with the YALMIP library [58], [67]-[69].

The ILP solver counts the lower bounds and an LP solver is utilized to solve the relaxed problems [8]-[16]. The local nonlinear solver is the FMINCON solver or the IPOPT optimizer routine embedded in the YALMIP BBA solver to come across the upper bounds [8]-[16]. Using NLP and ILP functions in optimizing the proposed model, it is proven the efficiency of the entire procedure by delivering trade-off optimal solutions. Such optimal solutions are given in [44]-[45] covering the concept of conflicting trends (minimize/maximize) [15]-[16].

Journal of Physics: Conference Series **2701** (2024) 012001 doi:10.1088/1742-6596/2701/1/012001

Table 42. Optimization Process: Results of bmibnb routine

* Starting YALMIP global branch & bound.
* Upper solver: fmincon
* Lower solver: INTLINPROG
* LP solver: INTLINPROG
* -Extracting bounds from model
* -Performing root-node bound propagation
* -Calling upper solver (no solution found)
* -Branch-variables: 300
* -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: 8.70000E+01 0.00 8.70000E+01 2 320s Solution found by heuristics
* Finished. Cost: 87 (lower bound: 87, relative gap 1.1361e-10%)
* Termination with relative gap satisfied
* Timing: 27% spent in upper solver (2 problems solved)
* 2% spent in lower solver (1 problems solved)
* 9% spent in LP-based domain reduction (600 problems solved)
* 1% spent in upper heuristics (1 candidates tried)
sol =
struct with fields:
yalmipversion: '20210331'
matlabversion: '9.4.0.813654 (R2018a)'
yalmiptime: 0.2423
solvertime: 263.7147
info: 'Successfully solved (BMIBNB)'
problem: 0
Elapsed time is 263.955948 seconds.
ans =
Columns 1 through 16
1 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0
Columns 17 through 32
0 0 1 0 0 0 1 0 1 0 1 0 1 0 0 0
Columns 33 through 48
1 0 0 0 1 0 0 0 0 0 1 0 0 0 0 1
Columns 49 through 64
1 0 0 0 1 1 0 0 0 0 0 0 0 1 0 1
Columns 65 through 80

Table 42. Optimization Process: Results of bmibnb routine (continued)

0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	1
Columns 81 through 96															
0	0	0	0	0	1	0	1	0	0	0	1	1	0	0	1
Columns 97 through 112															
0	1	1	0	1	0	0	0	0	0	0	0	1	0	1	1
Columns 113 through 128															
1	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0
Columns 129 through 144															
0	0	0	1	0	0	1	0	0	0	1	0	1	0	0	0
Columns 145 through 160															
0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
Columns 161 through 176															
0	0	0	1	0	0	1	1	0	0	1	0	0	0	0	0
Columns 177 through 192															
0	0	0	0	0	0	1	1	0	0	0	0	1	1	0	0
Columns 193 through 208															
1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
Columns 209 through 224															
1	1	0	1	0	0	1	1	1	0	0	0	0	0	1	0
Columns 225 through 240															
0	1	0	0	1	1	0	0	1	0	0	1	1	1	0	0
Columns 241 through 256															
0	1	0	0	0	0	0	0	0	0	1	1	1	0	0	1
Columns 257 through 272															
0	0	0	0	0	1	0	0	1	0	0	1	1	1	0	1

Table 42. Optimization Process: Results of bmibnb routine (continued)

Columns 273 through 288
0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
Columns 289 through 300
0 0 0 0 0 0 0 0 0 0 1 1
Linear scalar (real, binary, 300 variables)
Current value: 87
Coefficients range: 1 to 1
SORI =
371

Fig.19 illustrates the plot diagram of the entire optimization problem whereas the exact PMU positioning sites are shown in Table 43 without the optimal result to be affected by floating-points phainomenon.

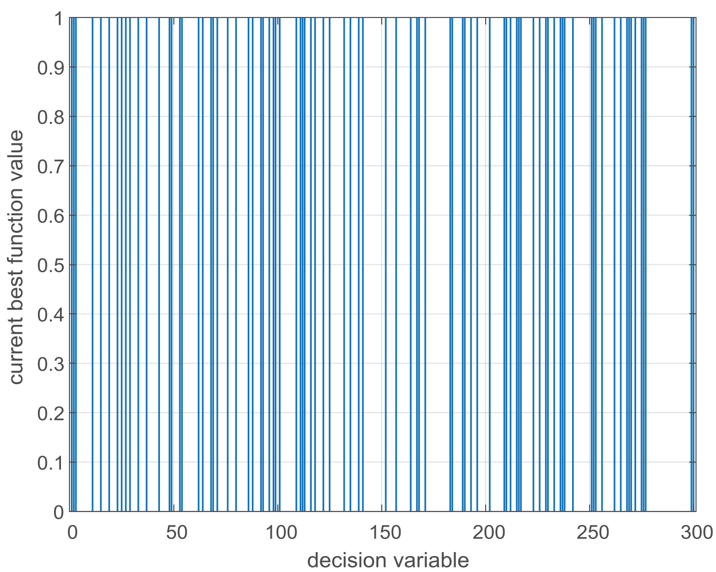

FIGURE 19. Optimal PMU Arrangement achieved by BMIBNB/ INTLINPROG /FMINCON

Table 44 illustrates that the optimal solution with maximum indicator of observability for a large-scale optimization problem reflects the IEEE-300 bus system [61], [71]. The process optimization illustrates that a B&B process is within a globalized strategy and converges to a global solution which is the upper bound of the B&B tree given at a root-node [70].

The upper bound is utilized to find the objective function for the minimization problem [14] whereas the lower bound minimizes the absolute gap delivering a global optimum solution [8]. This solution is achieved within a zero-absolute gap and a meaningless percentage relative gap [70].

TABLE 43. Optimal PMU Arrangement for the 300-bus Power Network

PMU Arrangement Locations with Observability Indicator Equal to 371
1, 2, 3, 11, 15, 19, 23, 25, 27, 29, 33, 37, 43, 48, 49, 53, 71, 76, 80, 86, 88, 92, 93, 96, 98, 99, 101 109, 111, 112, 113, 116, 118, 122, 125, 132, 135, 139, 141, 152, 157, 164, 167, 168, 171, 183, 184 189, 190, 193, 196, 202, 209, 210, 212, 215, 216, 217, 223, 226, 229, 230, 233, 236, 237, 238, 242 251, 252, 253, 256, 262, 265, 268, 269, 270, 272, 275, 276, 277, 299, 300

Table 44. Optimization Process: Results of bmibnb

++
* Starting YALMIP global branch & bound.
* Upper solver: fmincon
* Lower solver: INTLINPROG
* LP solver: INTLINPROG
* -Extracting bounds from model
* -Performing root-node bound propagation
* -Calling upper solver (no solution found)
* -Branch-variables: 300
* -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: 8.55600E+01 0.00 8.55600E+01 2 309s Solution found by heuristics
* Finished. Cost: 85.56 (lower bound: 85.56, relative gap 1.1553e-10%)
* Termination with relative gap satisfied
* Timing: 22% spent in upper solver (2 problems solved)
* 2% spent in lower solver (1 problems solved)
* 9% spent in LP-based domain reduction (600 problems solved)
* 1% spent in upper heuristics (1 candidates tried)
Elapsed time is 241.958038 seconds.
sol =
struct with fields:
yalmipversion: '20210331'
matlabversion: '9.4.0.813654 (R2018a)'
yalmiptime: 0.2492
solvertime: 241.7048
info: 'Successfully solved (BMIBNB)'
problem: 0
ans =
Columns 1 through 16
1 1 1 0 0 0 0 0 0 0 1 1 0 0 1 0
Columns 17 through 32
1 0 1 0 0 1 1 0 1 0 1 0 1 0 0 0

Table 44. Optimization Process: Results of bmibnb (continued)

Columns 33 through 48															
1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1
Columns 49 through 64															
1	0	0	0	1	1	1	0	0	1	1	1	0	1	0	1
Columns 65 through 80															
1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	0
Columns 81 through 96															
0	0	1	0	1	1	0	1	1	0	0	0	1	0	0	0
Columns 97 through 112															
0	1	1	0	1	0	1	0	0	0	0	0	1	0	1	1
Columns 113 through 128															
1	0	0	1	0	1	1	0	0	0	0	1	0	0	0	0
Columns 129 through 144															
0	0	0	1	0	0	1	0	0	1	0	0	0	0	1	0
Columns 145 through 160															
1	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0
Columns 161 through 176															
0	0	1	0	0	0	1	0	0	0	0	0	1	0	0	0
Columns 177 through 192															
1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0
Columns 193 through 208															
1	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1
Columns 209 through 224															
0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	1

Table 44. Optimization Process: Results of bmibnb (continued)

Columns 225 through 240															
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Columns 241 through 256															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Columns 257 through 272															
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1
Columns 273 through 288															
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Columns 289 through 300															
0	0	0	0	0	1	0	0	0	0	0	0				
Linear scalar (real, binary, 300 variables)															
Current value: 85.56															
Coefficients range: 0.96 to 0.99333															
SORI =															
432															

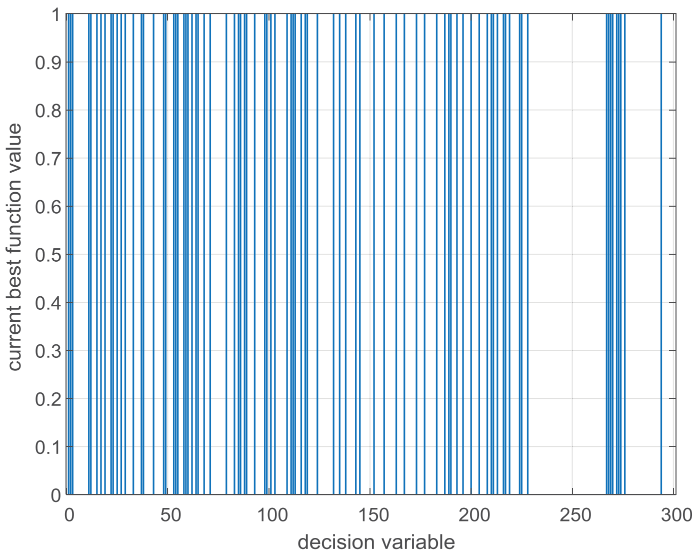

FIGURE 20. Optimal PMU Arrangement Achieved by BMIBNB/Intlinprog/FMINCON

TABLE 45. Optimal PMU Arrangement for the 300-bus Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 432
1, 2, 3, 11, 12, 15, 17, 19, 22, 23, 25, 27, 29, 33, 37, 38, 43, 48, 49. 53, 54, 55, 58, 59, 60, 62, 64 65, 68, 71, 79, 83, 85, 86, 88, 89, 93, 98, 99, 101,103, 109, 111, 112, 113, 116, 118, 119, 124, 132 135, 138, 143, 145, 152, 157, 163, 167, 173, 177, 183, 187, 189, 190, 193, 196, 200, 204, 208 210, 211, 213, 216, 217, 219, 224, 225, 228, 267, 268, 269, 270, 272, 273, 274, 276, 294

Tables 46 and 48 illustrate the BBA's performance calling the SCIP optimizer as an outer approximation solver [63]-[65]. This polyhedron formulation can be solved using an s-BBA embedded in SCIP optimizer [63]-[66].

During the iterative process, the optimizer calculates the variation between the upper and lower bounds and results in the best possible solution; a global one at the same time. The zero-gap is the desired tolerance to be met for the purpose of getting the optimum point at a one root-node.

Table 46. Optimization Process: Results of bmibnb routine

+++
* Starting YALMIP global branch & bound.
* Upper solver : fmincon
* Lower solver : SCIP
* LP solver : SCIP
* -Extracting bounds from model
* -Performing root-node bound propagation
* -Calling upper solver (no solution found)
* -Branch-variables: 300
* -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: 8.70000E+01 0.00 8.70000E+01 2 681s Solution found by heuristics
* Finished. Cost: 87 (lower bound: 87, relative gap 1.1364e-10%)
* Termination with relative gap satisfied
* Timing: 10% spent in upper solver (2 problems solved)
* 3% spent in lower solver (1 problems solved)
* 105% spent in LP-based domain reduction (600 problems solved)
* 1% spent in upper heuristics (1 candidates tried)
Elapsed time is 682.055768 seconds.
ans =
Columns 1 through 16
1 1 1 0 0 0 0 0 0 0 1 0 0 0 1 0
Columns 17 through 32
0 0 0 0 0 1 1 0 1 1 1 0 1 0 0 0
Columns 33 through 48

Table 46. Optimization Process: Results of bmibnb routine (continued)

1	0	0	0	1	1	0	0	0	0	1	0	0	0	0	1
Columns 49 through 64															
1	0	0	0	1	1	0	0	0	0	0	0	0	1	0	1
Columns 65 through 80															
0	0	0	1	1	0	1	0	0	0	0	1	0	0	0	0
Columns 81 through 96															
0	0	1	0	0	1	0	0	1	0	0	0	1	0	0	0
Columns 97 through 112															
0	1	1	0	1	0	1	0	0	0	0	0	1	0	1	1
Columns 113 through 128															
1	0	0	1	0	1	0	0	0	1	0	0	1	0	0	0
Columns 129 through 144															
0	0	0	1	0	0	1	0	0	1	0	0	1	0	0	0
Columns 145 through 160															
0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	1
Columns 161 through 176															
0	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0
Columns 177 through 192															
0	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0
Columns 193 through 208															
1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	0
Columns 209 through 224															
1	1	0	1	1	0	0	1	0	1	0	0	0	0	1	0
Columns 225 through 240															

Table 46. Optimization Process: Results of bmibnb routine (continued)

1	0	0	1	0	1	0	0	1	0	1	1	1	1	0	0
Columns 241 through 256															
0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
Columns 257 through 272															
0	0	0	0	0	1	0	0	1	0	0	1	1	1	0	1
Columns 273 through 288															
0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Columns 289 through 300															
0	0	0	0	0	0	0	0	0	0	1	1				
Linear scalar (real, binary, 300 variables)															
Current value: 87															
Coefficients range: 1 to 1															
SORI =															
384															

Figures 21 and 22 illustrate the best possible PMU sites for both case studies. Those optimal sites are shown in Tables 48 and 49. Table 49 illustrate the process optimization whereas the BBA tree is implemented to get the best optimal solution within zero-gap tolerance and a meaningless relative gap.

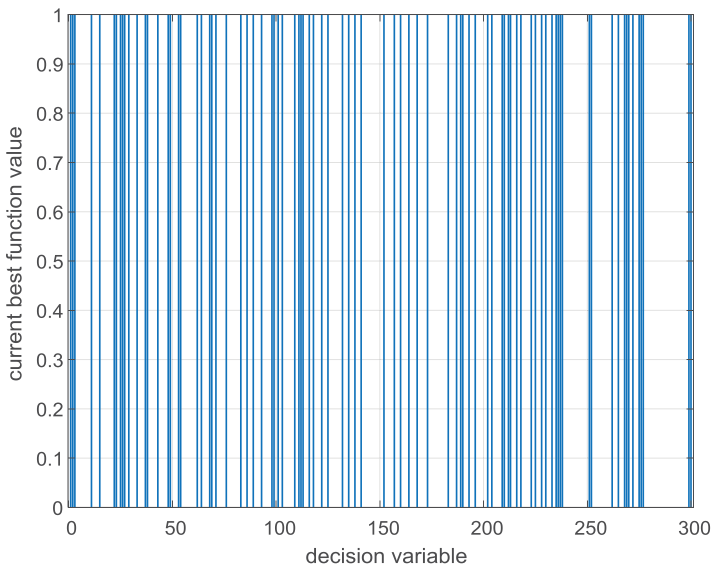

FIGURE 21. Optimal PMU Arrangement Achieved by BMIBNB Routine

TABLE 47. Optimal PMU Arrangement for the 300-bus Power Network

PMU Arrangement Locations with Observability Indicator Equal to 384
1, 2, 3, 11, 15, 22, 23, 25, 26, 27, 29, 33, 37, 38, 43, 48, 49, 53, 54, 62, 64, 68, 69, 71, 76 83, 86, 89, 93, 98, 99, 101, 103, 109, 111, 112, 113, 116, 118, 122, 125, 132, 135, 138, 141 152, 157, 160, 164, 168, 173, 183, 187, 189, 190, 193, 196, 202, 204, 209, 210, 212 213 216, 218, 223, 225, 228 230, 233, 235, 236, 237, 238, 251, 252, 262, 265, 268, 269, 270 272, 275, 276, 277, 299, 300

The optimal PMU localization sites are shown in Fig.22 whereas the exact sites are tabulated in Tables 48 & 49. Table 49 illustrates the performance of the multi-criterion objection function, and getting the global optimum point with a 0.00 optimality criterion [8]-[16].

The whole optimization model is successfully minimized at single run within an optimality gap. The optimality gap means how much better an optimum point can be achieved by the entire process optimization. The optimal solution is achieved with a zero-gap tolerance and meanglisess relative gap.

The iterative process ends up with the best optimal solution spending reasonable running time considering the optimization problem's dimension size. The B&B algorithm ends up with an upper bound within an acceptable relative gap, almost zero and within an absolute gap to be equal to zero and the lower bound is culprit to minimize it [68]-[70].

A global solution is achieved at the given root-node without signifact elapsed time. Thus, a global solution is attained at a given root-node within an absolute zero-gap and a meaningless percentage relative gap [70]. The $0-1$ polynomial model extends the programing models presented in [44]-[45] tested on the 300 bus system [61], [71]. The global BBA delivers the best optimal solution, that is, $J_{optimal} = 87$ with a system observability index equal to $SORI = 432$ [35], [44]-[45].

This measurement indicator redundancy is the upper amount for achieving measurements with maximum reliability in a Smart Grid [18], [23], [35], [44]-[45]. The optimal PMU's number and their positions in a power transmission grid help a lot to monitor in real-time the Smart Grid [2]-[3].

As a result of that, the gap tolerance is found to be equal to zero which ensures that a global optimality is discovered at a root-node. All log files illustrate that the optimizer routine is able to get a solution within 0.00 %, and this fact is a necessary and sufficient condition to certify optimality.

Table 48. Optimization Process: Results of bmibnb routine

++
* Starting YALMIP global branch & bound.
* Upper solver: fmincon
* Lower solver: SCIP
* LP solver: SCIP
* -Extracting bounds from model
* -Performing root-node bound propagation
* -Calling upper solver (no solution found)
* -Branch-variables: 300
* -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: 8.55600E+01 0.00 8.55600E+01 2 621s Solution found by heuristics
* Finished. Cost: 85.56 (lower bound: 85.56, relative gap 1.1553e-10%)
* Termination with relative gap satisfied

Table 48. Optimization Process: Results of bmibnb routine (continued)

* Timing: 10% spent in upper solver (2 problems solved)
* 1% spent in lower solver (1 problems solved)
* 104% spent in LP-based domain reduction (600 problems solved)
* 1% spent in upper heuristics (1 candidates tried)
Elapsed time is 622.548840 seconds.
ans =
Columns 1 through 16
1 1 1 0 0 0 0 0 0 0 1 1 0 0 1 0
Columns 17 through 32
1 0 1 0 0 1 1 0 1 0 1 0 1 0 0 0
Columns 33 through 48
1 0 0 0 1 1 0 0 0 0 1 0 0 0 0 1
Columns 49 through 64
1 0 0 0 1 1 1 0 0 1 1 1 0 1 0 1
Columns 65 through 80
1 0 0 1 0 0 1 0 0 0 0 0 0 0 1 0
Columns 81 through 96
0 0 1 0 1 1 0 1 1 0 0 0 1 0 0 0
Columns 97 through 112
0 1 1 0 1 0 1 0 0 0 0 0 1 0 1 1
Columns 113 through 128
1 0 0 1 0 1 1 0 0 0 0 1 0 0 0 0
Columns 129 through 144
0 0 0 1 0 0 1 0 0 1 0 0 0 0 1 0
Columns 145 through 160
1 0 0 0 0 0 0 1 0 0 0 0 1 0 0 0
Columns 161 through 176
0 0 1 0 0 0 1 0 0 0 0 0 1 0 0 0

Journal of Physics: Conference Series **2701** (2024) 012001 doi:10.1088/1742-6596/2701/1/012001

Table 48. Optimization Process: Results of bmibnb routine (continued)

Columns 177 through 192															
1	0	0	0	0	0	1	0	0	0	1	0	1	1	0	0
Columns 193 through 208															
1	0	0	1	0	0	0	0	0	1	0	1	0	0	0	1
Columns 209 through 224															
0	1	1	0	1	0	0	1	1	0	1	0	0	0	0	1
Columns 225 through 240															
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Columns 241 through 256															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Columns 257 through 272															
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1
Columns 273 through 288															
1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0
Columns 289 through 300															
0	0	0	0	0	1	0	0	0	0	0	0				
Linear scalar (real, binary, 300 variables)															
Current value: 85.56															
Coefficients range: 0.96 to 0.99333															
SORI =															
432															

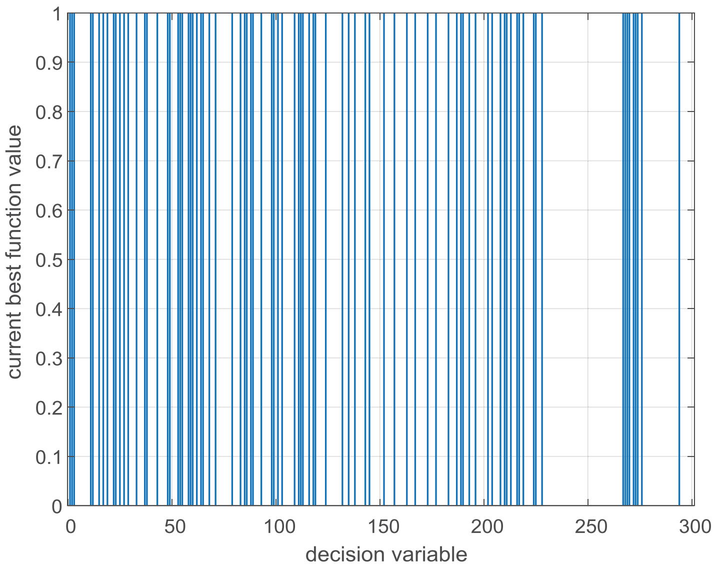

FIGURE 22. Optimal PMU Arrangement Achieved by BMIBNB/SCIP/FMINCON/Routines

TABLE 49. Optimal PMU Arrangement for the 300-bus Power Network

PMU Arrangement Locations with Maximum Observability Indicator Equal to 432
1, 2, 3, 11, 12, 15, 17, 19, 22, 23, 25, 27, 29, 33, 37, 38, 43, 48, 49, 53, 54, 55, 58, 59, 60, 62, 64 65, 68, 71, 79, 83, 85, 86, 88, 89, 93, 98, 99, 101, 103, 109, 111, 112, 113, 116, 118, 119, 124, 132 135, 138 , 143, 145, 152, 157, 163, 167, 173, 177, 183, 187, 189, 190, 193, 196, 202, 204, 208, 210 211, 213, 216, 217, 219, 224, 225, 228, 267, 268, 269, 270, 272, 273, 274, 276, 294

The simulation run illustrates the ability to accomplish optimality goals related to convergence speed and gap-tolerances and optimality metrics leading to Ɛ-optimality. The optimum point is a global solution within a zero-gap tolerance and a trivial amount of computational time spent by the optimizer function. Competitive trends getting involved in the optimization function are satisfied.

The solution has an objective 8.55600E+01 whereas the lower bound is at 8.55600E+01 which is responsible to minimize the absolute gap. The optimal solution is with a 0.00 % optimality whereas the relative gap is meaningless [70]. As presented by the log file, the lower bound closed the gap ensuring a globally optimality of certificate. The whole procedure is terminated at one root-node.

As observed, the two criteria declared in the augmented function are fully satisfied given that the optimal solution is a feasible one from the aspect of minimization of synchrophasor sensors optimally placed around the 300-bus system while the maximum reliability of measurements are achieved.
SCIP delivers the best optimal solution within a gap equal to zero [64]-[65].

Thus, a global optimal solution is achieved. The optimal result is a global solution given that it is achieved satisfying a zero-gap tolerance and a relative gap close to zero [70]. Thus, the B&B tree's upper bound is considered the best optimal solution for the mininimization problem [9]; the global one since the gap optimality-tolerance is zero [14]-[15], [70].

Hence, the lower bound is the culprit to close the gap [69]. A zero-gap tolerance and a relative gap close to zero or zero are calculated and they depend on the complexity of the minimization problem, the achievement of a global optimal solution is attained under warranty [53].

All algorithms are stable for solving the process optimization, and thus a globally convergent algorithmic scheme is developed to come across optimality in reasonable runtime considering the dimension size of the optimization problem [24]-[25].

The experimental results have illustrated that the new B&B algorithm is easy to give global optimality jointly with external solvers such as ILP and NLP solvers. Hence, the optimization problem is declared without considering uncertainties being relevant to the determination of a global optimal solution [24]-[25]. The log file is the certificate to ensure that the global solution has been achieved.

Three integer programming solvers were selected to count the lower bound, to solve the relaxed problems and estimate the variation between the upper and lower bounds [14]. Gurobi and MATLAB intlinprog are commercial solvers and SCIP is an open-source code [62], [66]-[67].

The only considerable experimental time is spent by the upper solver to come across the first incumbent or otherwise best integral solution [13]. As an Upper Solver, FMINCON was executed for the purpose of getting the best integral solution found so far [62]. Also, a considerable run time is spent in LP-based domain reduction. The elapsed time is reasonably spent considering the size of the minimization problem. We illustrate the log file for the minimization problem in Table 46.

A robust non-unique constrained global optimum point is derived spending a trivial run time considering the dimension size of the optimization problem. The gap was found zero where the relative gap is satisfied. Hence, the YALMIP B&B optimizer delivers a global optimum point [70].

17. Numerical Discussion about the simulation results

In this study, we have introduced and studied the transformation of a polynomial binary (Boolean) optimization model by polyhedral approximation methods, specifically the concept of using ILP solvers and NLP solvers to count the upper and lower levels in the B&B tree [8]-[14].

The foremost innovation is that a system of polynomial equalities can be linearized to a polyhedral approximation where the binary decision variables are getting relaxed to get an optimal solution [14]-[16]. The attracting attention characteristic of the proposed algorithmic scheme is that a global optimal solution is achieved as well as a trivial runtime is spent [70]. The algorithmic scheme was tested on power systems from $14\ to\ 300$ network buses and solved on a global solution. Some corresponding optimal results were derived and found to be global optimal solutions [8]-[9].

The global optimal solution ensures full condition observability and maximizes the number of measurements in unison for each power network bus. Hence, each power network node is monitored with the maximum number of times by a well-established PMUs number in the power grid either directly or indirectly [18], [23], [35], [38], [41]-[42], [44]- [45].

A computational difficult task is to come across the global optimum, and most times the nonlinear algorithm is getting stuck in a local optimum point [8]-[16], [24]-[31]. To avoid being trapped in a non-global minimum point, clustering methods are used to initialize the iterative process [44].

For that reason, we declare the transformation of the initial model into a polyhedral approximation, to ensure sufficient conditions for optimality where a global solution has been found for every benchmark IEEE test system [63]-[64].

It is significant to declare that the optimization model is able to be solved towards global optimality although the constraint function is non-convex with a binary restriction $\{\mathbf{0}, \mathbf{1}\}$ and usually a local minimum point is detected by the optimization calculations being used so far [8]-[16], [24]-[31].

Here we draw a process optimization to come across global optimal solutions with a BMIBNB solver. This optimizer routine interfaces an LP solver for branching implementation of the B&B tree.

The ILP solver as well as the NLP solver compute both the upper and lower bounds on the objective value at the given root-node. The absolute gap is the variation between those bounds to declare a condition for an adequate optimality.

The variation between those bounds results in a zero optimality gap value. Hence, a global optimal solution is attained for every experimental result [68]-[70].

The arousing interest of this research work is that the proposed programming model is solved globally. As found by simulating the polynomial model, the LP-based domain reduction is performed in less time than the case without selecting the above option [8]-[16].

As observed by the desired outcome, the cost function value is the best outcome given that this cost price is on the side of an absolute gap tolerance. The minimization problem is proved easy to be solved because the solution has been found at the given root-node within a 0.00 % optimality gap.

The optimality criterion is an absolute tolerance for optimality and it found to be zero giving with this way the best possible solution. This solution is the upper bound on the objective function for the minimization problem. The zero-gap tolerance is a desired outcome obtainable by a considerable termination criterion to take accountability to come across a global optimal solution point [14].

The optimal solution point is the best number of PMU which is a necessity to cover the complete power network [17]-[23]. This PMUs number is adequate in a second stage for maximizing the amount of the measuring times for each power network bus to be observed within the power transmission grid. The optimal solutions are equal in quality as well as in quantity to those found by nonlinear and evolutionary algorithms [44]-[45]. Hence, a certificate of optimality is achieved.

18. Noteworthy Remarks and Final Outlook

This study comes to cover this lack of global knowledge in getting optimal solutions related to the specific optimization problem. Thus, the aim is to deliver the true optimality of the mathematical model being optimized using the YALMIP in combination with MATLAB.

The binary (Boolean) optimization model is solved through the BMIBNB optimizer function to figure out the global optimal solution and tested on realistic power transmission systems.

BMIBNB is a customary branch & bound algorithm for solving non-convex optimization problems that relies on linear programming (LP) relaxations and convex approximations enclosed in the B&B tree. The BMIBNB optimizer routine interfaces suitable external solvers to solve the proposed programming model. The external solvers are ranked into NLP and ILP solvers.

The algorithmic procedure interacts with technology advanaced ILP functions and local solvers for the purpose of getting a global optimal solution.

This can be produced by using widespread backbone optimization tools acting in the YALMIP environment, an optimization library which is fully compatible with optimizer functions embedded in MATLAB optimization library, Gurobi, and open-source NLP and ILP solvers. These optimization functions enact an interior in YALMIP function to effectively solve the proposed $0-1$ NLP model for all supported optimization solvers, commercial or not in the direction of getting global optimality.

The aim is to define all decision variables in a binary symbolic format and the constraints needed for the optimization problem statement. BMIBNB interfaces external solvers such as FMINCON or IPOPT as an upper solver and a suitable ILP solver as a lower routine.

The upper solver such as the FMINCON or IPOPT routine is utilized to attain the upper bound on the optimum point, that is, the best integral solution achieved so far by the process optimization.

The ILP solver is adopted as an LP solver to implement the branching and exploring the nodes in the B&B tree as well as to prune infeasible regions [70]. The ILP optimizer builds the B&B tree to estimate the lower bounds whereas the local solver comes across the upper bounds.

Aslo, a lower bound is calculated being responsible to close the absolute gap. LP relaxations are solved at root-nodes and the upper and lower bounds are constantly compared so as a gap tolerance as well as a meaningless relative gap to be computed for getting the global optimum.

The routine is suitable for the implementation of non-convex optimization problems that rely on linear programming (LP) relaxations that encloses a convex act of estimating. The lower solver takes care of solving the LP relaxations during the B&B implementation tree whereas the nonlinear solver measures the appropriate computations of the upper bounds at one root-node. The upper bound is the objective solution while the lower bound is culprit to close the absolute gap giving a global optimality.

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. It encounters non-unique constrained global minima points with sufficient warranty within a zero-gap tolerance criterion.

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] Abur A. and Expósito A.G 2004 Power System State Estimation: Theory and Implementation (1st ed.). CRC Press

[4] Baldwin T. L., Mili L., Boisen M. B. and Adapa R. 1993 Power system observability with minimal phasor measurement placement", IEEE Trans. Power Syst., vol. 8, no. 2, pp. 707-715
[5] 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
[6] 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
[7] 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
[8] Arora J.S 2016 Introduction to Optimum Design MA", Elsevier Academic Press
[9] Bradley S. P., Hanx A. C., and Magnanti T. L. 1977 Applied Mathematical Programming", Boston MA: Addison-Wesley
[10] Sarmas E., Xidonas P, & Doukas, H 2020 Literature review. "In Multicriteria portfolio construction with python", Springer optimization and its applications (Vol. 163). Springer
[11] Kokkos I 2022 Practical Chemical Process Optimization With MATLAB® and GAMS®", Springer Optimization and Its Applications, Springer Cham
[12] Gerard D, Köppe M, and Louveaux Q 2017 Guided dive for the spatial branch-and-bound, J Glob Optim 68, 685–711
[13] Tawarmalani M and Sahinidis N.V 2002, Convexification and Global Optimization in Continuous and Mixed-Integer_Nonlinear Programming: Theory, Algorithms, Software, and Applications, Nonconvex Optimization and Its Applications, Vol. 65, Kluwer Academic Publishers
[14] Karlof J K 2006 Integer Programming: Theory and Practice, Boca Raton: Taylor & Francis Group LLC
[15] Williams H P 2013 Model Building in Mathematical Programming, New York: Wiley
[16] Chinneck J W 2008 Feasibility and Infeasibility in Optimization, Int. Ser. in Oper. Res. & Man. Sci
[17] Xu B and Abur A 2004 Observability analysis and measurement arrangement for systems with PMUs, IEEE PES Pow. Sys. Conf. and Exp. New York
[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] Ghosh S., Isbeih Y. J., Azman S. K., Moursi M. S. E. and El-Saadany E. 2022 Optimal PMU Allocation Strategy for Completely Observable Networks With Enhanced Transient Stability Characteristics, in IEEE Transactions on Power Delivery, vol. 37, 5, pp. 4086-4102
[20] Aghaei J., Baharvandi A., Akbari M.-A., Muttaqi K. M., M.-R. Asban R, and Heidari A. 2015, Multi-objective phasor measurement unit arrangement in electric power networks: integer linear programming formulation, Electric Power Compon. Syst., vol. 43, no. 17, pp. 1902–1911
[21] Theodorakatos N P 2018 Optimal phasor measurement unit arrangement for numerical observability using a two-phase branch-and-bound algorithm", Int. J. of Emer. Elec. Pow. Sys. 19, 3
[22] Babu R and Bhattacharyya B 2019 Strategic arrangements of PMUs for power network observability considering redundancy measurement Meas. J. of the Int. Meas. Confed. 134
[23] Xia N, Gooi H.B, Chen SX, Wang MQ, 2015 Redundancy based PMU placement in state estimation", Sustain Energy Grids Netw 2:23–31
[24] Luenberger D. G and Ye Y. 2008 Linear and Nonlinear Programming", 3rd ed. Springer
[25] Antoniou, A., W Lu. -S 2007 The optimization problem. In: Antoniou, A., Lu, W.-S. (eds.) Practical Optimization, pp. 1–26. Springer, Heidelberg
[26] 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, pp. 2427-2436
[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] Theodorakatos N.P, Lytras M, Babu R 2021 Generalized Pattern Search Algorithm Methodology for solving an Under-Determined System of Equality Constraints to achieve Power System Observability using Synchrophasors", Journal of Physics: Conference Series 2090, 012125
[32] Bečejac V and Stefanov P, G 2020 Groebner bases algorithm for optimal PMU placement. International Journal of Electrical Power & Energy Systems *Int. J. of Elec. Pow. & Ene. Sys.* 115
[33] 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
[34] 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
[35] 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
[36] 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, pp. 2550-2559
[37] Mishra C, Jones K D, Pal A and Centeno V 2016, Binary particle swarm optimization-based optimal substation coverage algorithm for phasor measurement unit installations in practical systems, IET Gen. Trans. Dist., 10, 2, pp. 555-562
[38] Dalali M, Karegar H.K, Optimal PMU arrangement for full observability of the power network with maximum redundancy using modified binary cuckoo optimization algorithm, IET Gener. Transm. Distrib, 10, 2817–2824
[39] Peng J, Sun Y., Wang H.2006 Optimal PMU arrangement for full network observability using Tabu search algorithm, Int. J. Electr. Power Energy Syst., 28 (4), pp. 223-231
[40] Koutsoukis N. C., Manousakis N. M., Georgilakis P. S., and Korres G. N 2013 Numerical observability method for optimal phasor measurement units placement using recursive Tabu search method," IET Gen. Transm. Distrib*.* 7, 4, pp. 347–356
[41] Singh S. P, and Singh S. P, 2017 A multi-objective PMU arrangement method in power system via binary gravitational search algorithm, Electric Power Compon. Syst*.*, vol. 45, no. 16, pp. 1832–1845
[42] Ramasamy S, Koodalsamy B, Koodalsamy C and Veerayan M 2021 Realistic Method for Arrangement of Phasor Measurement Units through Optimization Problem Formulation with Conflicting Objectives, Electric Power Components and Systems, 4-5, 49
[43] Shafiullah M, Hossain M. I., Abido M., T. Abdel-Fattah and Mantawy A. 2019 A modified optimal PMU arrangement problem formulation considering channel limits under various contingencies", Measurement*,* vol. 135, pp. 875-885
[44] Theodorakatos N. P, Lytras M.D., Moschoudis A, Kantoutsis K. Implementation of Optimization-based Algorithms for Maximum Power System Observability using Synchronized Measurements, *A*IP Conf. Proc. 28 September 2023; 2872 (1): 120006. https://doi.org/10.1063/5.0162930
[45] 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

[46] Samarth G., 2014 Improving strong branching by domain propagation, EURO J Comput Optim 2, 99–122

[47] Murray W, Ng KM 2010 An algorithm for nonlinear optimization problems with binary variables, Comput Optim Appl, 47, 257–288

[48] Miettinen K., Mäkelä M, 2002 On scalarizing functions in multiobjective optimization, OR Spectrum, 24, 193–213

[49] Nikulin Y., Miettinen K., and Mäkelä M.M 2012 A new achievement scalarizing function based on parameterization in multiobjective optimization", OR Spectrum, 34, 69–87

[50] Gunantara N., 2018 A review of multi-objective optimization: methods and its applications, Cogent Eng. 5(1):1502242

[51] Chugh T. 2020 Scalarizing Functions in Bayesian Multiobjective Optimization, 2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1-8

[52] Tuy, H 2016 «Polynomial Optimization. In: Convex Analysis and Global Optimization, » Springer Optimization and Its Applications, vol 110. Springer

[53] Kronqvist, J., Bernal D.E Lundell A 2019 A review and comparison of solvers for convex MINLP, Optim Eng., 20, 397–455

[54] Berthold, T 2008 Heuristics of the branch-cut-and-price-framework SCIP. In: Kalcsics, J., Nickel, S. (eds.) Operations Research Proceedings 2007. ORP, vol. 2007, pp. 31–36. Springer, Heidelberg

[55] Gleixner A.M, Berthold T., Müller, B, Weltge S 2017 Three enhancements for optimization-based bound tightening, J Glob Optim 67, 731–757

[56] Androulakis I.P., Maranas C.D., Floudas C.A 1995 aBB: a global optimization method for general constrained non-convex problems. J. Glob. Optim. **7**(4), 337–363

[57] Belotti P.., Kirches C., Leyffer S., Linderoth J., Luedtke J.., and Mahajan A. 2013 Mixed-integer nonlinear optimization", Acta Numerica, 22, 1-131, doi:10.1017/S0962492913000032

[58] Belotti P, Lee J, Liberti L, Margot F, and Wächter A., 2009 Branching and Bounds Tightening Techniques for Non-Convex MINLP, Optimization Methods and Software 24 (4–5): 597–634

[59] 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/

[60] Currie J, Wilson DI et al, 2012 OPTI: lowering the barrier between open-source optimizers and the industrial MATLAB user. Found Comput Aided Process Oper, 24:32

[61] 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

[62] The MathWorks Inc. Optimization Toolbox for use with MATLAB R®. User's Guide for Mathwork. 2016. Available online: www.mathworks.com

[63] Achterberg T. 2009 SCIP: Solving constraint integer programs", Math. Program. Comput. 1(1), 1–41

[64] Vigerske S, Gleixner, A, 2017 SCIP: global optimization of mixed-integer nonlinear programs in a branch-and-cut framework", Optim. Methods Softw

[65] 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.

[66] OPTI Toolbox v2.21 A free MATLAB toolbox for optimization. [Online]. Available: https://github.com/jonathancurrie/OPTI

[67] "Gurobi optimizer reference manual", 2022, [online] Available: http://www.gurobi.com.

[68] https://YALMIP.github.io/tutorial/globaloptimization/

[69] https://YALMIP.github.io/multi-objective-problems

[70] https://YALMIP.github.io/solver/bmibnb/

[71] https://icseg.iti.illinois.edu/power-cases/

[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