Optimal Placement of Phasor Measurement Units for Maximum Network Observability Using Python-Gurobi

Anil Khanal
North Carolina A & T State University
akhanal1@aggies.ncat.edu

Emmanuel Oleka
North Carolina A & T State University
eoleka@aggies,ncat.edu

Ali Osareh
North Carolina A & T State University
osareh@ncat.edu

Gary Lebby
North Carolina A & T State University
lebby@ncat.edu

Abstract

Advancements in synchrophasor technology have continued to open doors for the ever-increasing applications in electric power grid visualization and monitoring, and at the same time calls for more research into optimal deployment of the technology for greater efficiency [1]. Mathematical models and heuristic models have already been developed for finding the optimal positions for the phasor measurement units (PMUs) in an electric power network. Integer programing is a popular mathematical model that has been employed. In this paper, Python-Gurobi was used with integer linear programing to accomplish this goal. The optimal PMU placement problem is formulated to minimize the number of PMU installation, subject to full network observability. This was implemented in standard IEEE 14, 30, and 57 system busses. These approaches were further implemented in the 30-bus system on the Nigerian electric power grid. The results were the same as those achieved in previous works by different authors, but we had a remarkable improvement in CPU computation time. The integer linear programming method for obtaining the optimum placement of PMUs for complete observability of the system was performed in two parts: obtaining a minimum number of PMUs required for complete system observability considering an intact system and obtaining a minimum number of PMUs required for complete system observability considering one line/one PMU outage.

Introduction

PMUs are power system devices that provide synchronized real-time measurements of phasors of voltages and currents [2]. Synchronization is achieved by time sampling of voltage and current waveforms using timing signals from the global positioning system satellite (GPS) [1].

PMU placement at all substations allows direct measurement of the state of the network. However, PMU placement on each bus of a system is difficult to achieve, due to either the cost factor or limited communication channels. A PMU placed at one bus can measure the voltage phasor of that bus and the current phasors of all the incident busses. The voltage phasor at other adjacent buses can then be derived using Ohm's law [3, 4]. This implies that a system can be made observable with a lesser number of PMUs than the number of buses. Optimal placement involves deploying the least number of PMUs that can effectively monitor the system into the appropriate buses for complete network observability. The placement problem then can be formulated as an optimization problem [5-9, 13]. Many authors have contributed with various papers, starting with the constrained optimization formulation [8]. Tomlab and SCIP respectively were used in [9, 10] in finding the solution. This paper uses the same method for defining the problem at hand and uses Python-Gurobi for finding solutions.

The problem is divided into two parts. In the first part, the objective is to find the minimum number of PMUs satisfying the complete system observability constraint considering the intact system. The second part examines finding the optimal solution, considering one line loss or one PMU outage scenario.

Problem Formulation

The minimum number of PMU placement in the network NP-complete [9]. This implies that no polynomial time algorithm can be designed to solve the problem exactly. Work on optimal PMU placement using an ILP approach has been pioneered by [8]. The optimal PMU placement is formulated by minimizing the cost function, which is also the objective function.

$$\sum_{i=1}^{n} x_i \cdots\cdots (1)$$

Subject to the constraints:

$$Ax \geq e \quad where \;\; e \in [111....1]^T \wedge x \in [x_1 x_2 .....x_n]^T \cdots\cdots (2)$$

$$x_i = \begin{cases} 1 & if\ PMU\ is\ installed\ at\ Bus \quad i \\ 0 & Otherwise \end{cases} \wedge A = a_{i,j} = \begin{cases} 1 & if\ either\ i=j \quad or\ if\ i\ and\ j\ are\ adjacent\ nodes \\ 0 & Otherwise \end{cases} \cdots\cdots (3)$$

and A is the binary connectivity matrix of the system and n is the number of busses.

Python-Gurobi

The Gurobi optimizer [10] is a modern solver for (mixed integer) linear as well as other related (non-linear) mathematical optimization problems. The Gurobi optimizer is written in C, and it is available on all computing platforms and accessible from several programming languages. Standard independent modeling systems can be used to define and to model problems. Gurobi has been found to be one of the most CPU-efficient commercially available integer programing solvers as reported in [14], shown in Figure 1.

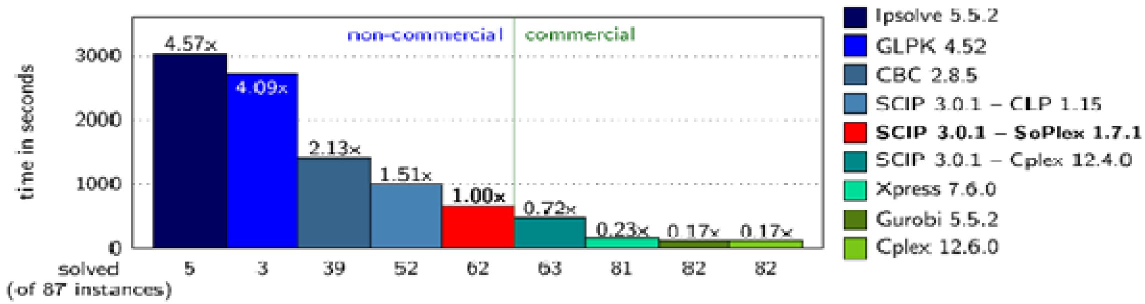

Figure 1. Comparison of different solvers applied to different benchmark problem instances

Case Studies

The ILP was run in an IEEE standard bus, and results were compared to see the performance. They are implemented in the IEEE 14, 30 and 57 bus system and the results are compared to the prototype solutions. This algorithm is further extended to test it in the 30-bus Nigerian grid.

The ILP model used Python-Gurobi [10] as the optimization tool to come up with the optimal PMUs number and their locations. The problem formulation details are elaborated only for the 30-bus system Nigerian grid, shown in Figures 2 and 3, while the results for all IEEE standard bus systems are presented in the tabulated format. These values are compared among each method and against the benchmark data by other authors [8, 11]

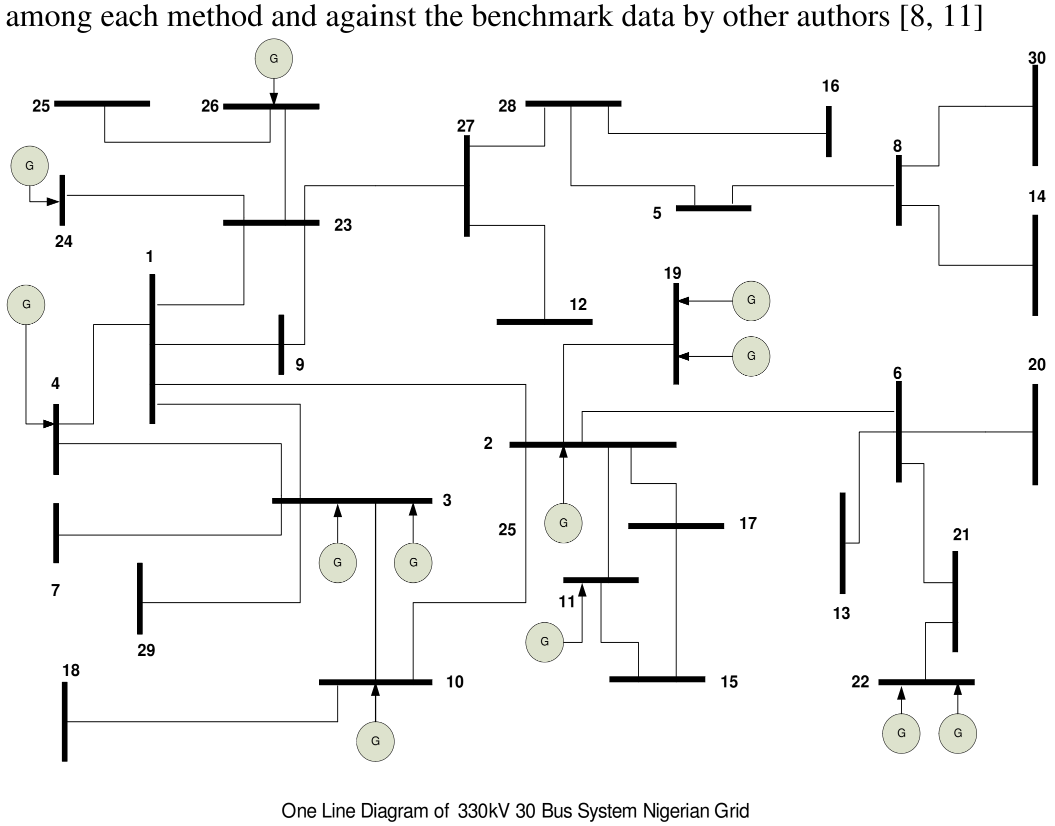

Figure 2. Nigerian grid

ISBN 978-1-60643-379-9

The constraints for the 30-bus Nigerian grid are defined with equation 2 and Figure 3 as

$$A = \begin{bmatrix}
1&1&1&1&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\
1&1&0&0&0&1&0&0&0&0&1&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0 \\
1&0&1&1&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\
1&0&1&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\
0&1&0&0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0&0 \\
0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&1&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\
0&0&1&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&1&0&0&0&0&0&0&0&0&1&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&1&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\
0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0&0&0 \\
1&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&1&1&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&1&0&0&0&0 \\
0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0&1&1&0&0&0&0 \\
0&0&0&0&0&0&0&0&1&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&1&1&0&0 \\
0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&1&1&0&0 \\
0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\
0&0&0&0&0&0&0&1&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&0&1
\end{bmatrix}$$

Figure 3. Constraint matrix for 30-bus system Nigerian grid

Experiments and Results

The results from the standard IEEE buses as well as 30 bus systems Nigerian Grid for ILP model using both Python-Gurobi and Tomlab are shown in the tables below. The tables are for the intact system and the system with one line outage or a single loss of a PMU.

Integer Linear Programming Approach

1. Intact system

Table 1. Number and locations of the PMUs for an intact system

Bus System	Location of PMUs	Number of PMUs
IEEE 14	2,7,10,13	4
IEEE 30	1,5,6,9,10,12,15,19,25,27	10
IEEE 57	1,6,9,15,19,22,25,27,28,32,36,39,41,44,47,50,53	17
30 Bus Nigerian Grid	2,3,6,8,10,16,17,21,23,26,27	11

2. Single line outage (single PMU loss)

Table 2. Number and locations of the PMU considering one PMU outage

Bus System	Location of PMUs	Number of PMUs
IEEE 14	2,4,5,6,7,8,9,10,13	9
IEEE 30	1,2,3,6,7,9,10,11,12,13,15,17,19,20,22,23,25,26, 27,28,30	21
IEEE 57	1,2,4,6,9,11,12,15,19,20,22,24,25,27,28,29,30,32, 33,35,36,37,38,39,41,44,46,47,50,51,53,54,56	33
30 Bus Nigerian Grid	1,2,3,6,7,8,10,12,13,14,15,16,17,18,19,20,21,22,23,2 25,26,27,28,29,30	26

The experiment with Tomlab [15] and Python-Gurobi in Table 4 has been performed with the computer with Intel Core i5-2450M-CPU @2.50GHz(4CPUs), ~2.5GHz and compared against each other and further compared with [12] for SCIP.

Table 3. Comparison of different solver in terms of CPU clock

Bus System	**CPU Time for SCIP**	**CPU Time for Tomlab**	**(Proposed) CPU time for Python-Gurobi**
IEEE 14	0.05	0.014	0.00001
IEEE 30	-	0.125	0.0001
IEEE 57	30.41	1.73	0.36
Nigerian 30 bus	-	1.57	0.05

Observations/Discussion

It was observed that the number of PMUs and their locations obtained using Tomlab and Python-Gurobi are the same as shown in Tables 1 and 2, but there is a significant improvement in CPU time in Python-Gurobi compared to Tomlab and even more compared to SCIP [13]. The number of PMUs required for network coverage increased when there was one PMU outage because the system had to reconfigure to assume a new structure posed by the outage.

Conclusion

Python-Gurobi has been used in solving the problem of optimal placement of PMUs in an electric power grid for complete observability and monitoring in LIP. It was discovered that it achieved the same solution as other tools, but it exceeds the other tools in CPU efficiency. Hence, we recommend the use of Python-Gurobi in solving problems of similar models.

References

[1] Phadke, A. G. (2002, October). Synchronized Phasor Measurement—A Historical Overview. *IEEE/PES Transmission and Distribution Conference and Exhibition*, *1*, 476-479.

[2] A. G. Phadke, A. G., Pickett, B., Adamiak, M., & Begovic, M. (1994, January). Synchronized Sampling and Phasor Measurements for Relaying and Control. *IEEE Transactions on Power Delivery*, *9*(1), 442-452.

[3] Dongjie, X., Renmu, H., & Tao, X. (2004, April). Comparision of Several PMU Placement Algorithms for State Estimation. *IEEE International Conference on Developments in Power System Protection*, 32-35.

[4] Denegri, G. B., Invernizzi, M., & Milano, F.. (2002, October). A Security Oriented Approach to PMU Positioning for Advanced Monitoring of a Transmission Grid. *IEEE International Conference on Power System Technology*, *2*, 798-803.

[5] Korkali, M., & Abur, A. (2009, July). Placement of PMUs with Channel Limits. *IEEE Power and Energy Society General Meeting*.

[6] Bladwin, T. L., Mili, L., Boisen, M. B., & Adapa, R. (1993, May). Power System Observability with Minimal Phasor Measurement Placement. *IEEE Transactions on Power Systems*, *8*(2), 705-715.

[7] Nuqui, R. F., & Phadke, A. G. (2005, October). Phasor Measurement Unit Placement Techniques for Complete and Incomplete Observability. *IEEE Transactions on Power Delivery*, *20*(4), 2381-2388.

[8] Xu, B., & Abur, A. (2004, October). Observability Analysis and Measurement Placement for Systems with PMUs. *Proceedings of IEEE Power System Conference and Exposition*, *2*, 943-946.

[9] D. J. Brueni, D. J. & Heath, L. S. (2005, December). The PMU Placement Problem. *Journal of Discrete Math*, *19*(3), 744–761.

[10] Gurobi Optimization. (2014). Retrieved from http://www.gurobi.com/

[11] D. Dua, D., Dambhare, S., Gajbhiye, R. K., & Soman, S. A. (2008, October). Optimal Multistage Scheduling of PMU Placement: An ILP Approach. *IEEE Transactions on Power Delivery*, *23*(4), 1812-1820.

[12] Meindl, B**., &** Templ Forschungsbericht, M., (n.d.) Analysis of Commercial and Free and Open Source Solvers for Linear Optimization Problems. CS-2012-1. Retrieved from http://www.statistik.tuwien.ac.at

[13] Kavasseri, R. & Srinivasan, S. K. (2011). Joint Placement of Phasor and Conventional Power Flow Measurements for Fault Observability of Power Systems. *IET Generation, Transmission & Distribution*, *5*(10), 1019-1024.

[14] Solving Constraint Integer Programming. (2014). Retrieved from http://scip.zib.de/

[15] Tomlab Optimization Inc. (n.d.). Retrieved from http://www.tomopt.com/

Biography

ANIL KHANAL completed his bachelor's degree in Electrical Engineering from National Institute of Technology, Karnataka, India, in 1999. He worked in an electrical transmission line project from 1999 to 2004 as an overall project coordinator and worked in GSM mobile telecommunications as an optimization and planning engineer. He completed his master's degree from University of Bridgeport in 2008. Since 2011, he has been pursuing a Ph.D. at North Carolina A & T State University. Research interests include renewable integration and reliability issues, use of synchrophasors for enhanced grid monitoring.

EMMANUEL OLEKA obtained a B.Eng and M.Eng in Electrical & Electronics Engineering from Enugu State University of Science and Technology, Enugu, Nigeria. He has worked in various capacities and is currently pursuing a Ph.D. degree in Electrical Engineering at North Carolina Agricultural and Technical State University, Greensboro, NC.

ALI R. OSAREH is currently an adjunct associate professor of Electrical and Computer Engineering at North Carolina Agricultural and Technical State University, specializing in power systems. Dr. Osareh leads industrial automation efforts in BIEES. Dr. Osareh can be reached at osareh@ncat.edu.

GARY L. LEBBY is currently a professor of Electrical and Computer Engineering at North Carolina Agricultural and Technical State University and a senior member of IEEE, specializing in power systems and artificial neural networks. Dr. Lebby is the director of the Laboratory for Biologically Inspired Engineering and Energy Systems (BIEES). Dr. Lebby can be reached at lebby@ncat.edu.