Impact Factor (SJIF): 3.632

International Journal of Advance Research in Engineering, Science & Technology

e-ISSN: 2393-9877, p-ISSN: 2394-2444

Volume 3, Issue 4, April-2016

OPTIMUM PMU PLACEMENT TECHNIQUES - A COMPARATIVE STUDY

Hardik Vakharia[1], Neelam Bhadauria[2]

[1]*PG Student – Electrical department, SSSRAGI*
[2] *Asst. Professor – Electrical department, SSSRAGI*

Abstract — PMU can improve state estimation accuracy significantly. Hence, system operability can be improved. Putting PMU at every bus is very costly solution. Here in this paper different Mathematical, Heuristic & Meta heuristic methods of optimum PMU placement techniques are studied & the result of each method for IEEE-6 Bus & IEEE-14 bus system are compared to find out optimum numbers PMUs & their location.

Keywords-PMU; observability rules; placement techniques; zero injection; SORI, OPP methods

I. INTRODUCTION

Due to the increasing development of power networks, their control systems and protection requirements are becoming complex. Many technologies have come along since the inception of state estimation that have improved its performance and drove it to become an integral part of control center operations. Today, Phasor Measurement Unit (PMU) technology will serve as the next step in improving the quality of the estimate of the system state, providing operators with better information to maintain a high level of system reliability. As the cost of the PMUs are the very high so that the putting PMUs at every bus is not a good solution. Optimum PMU placement techniques are applied to find out minimum number of PMUs so that complete observability can be achieved with minimum cost.

II. OPTIMUM PMU PLACEMENT ALGORITHMS

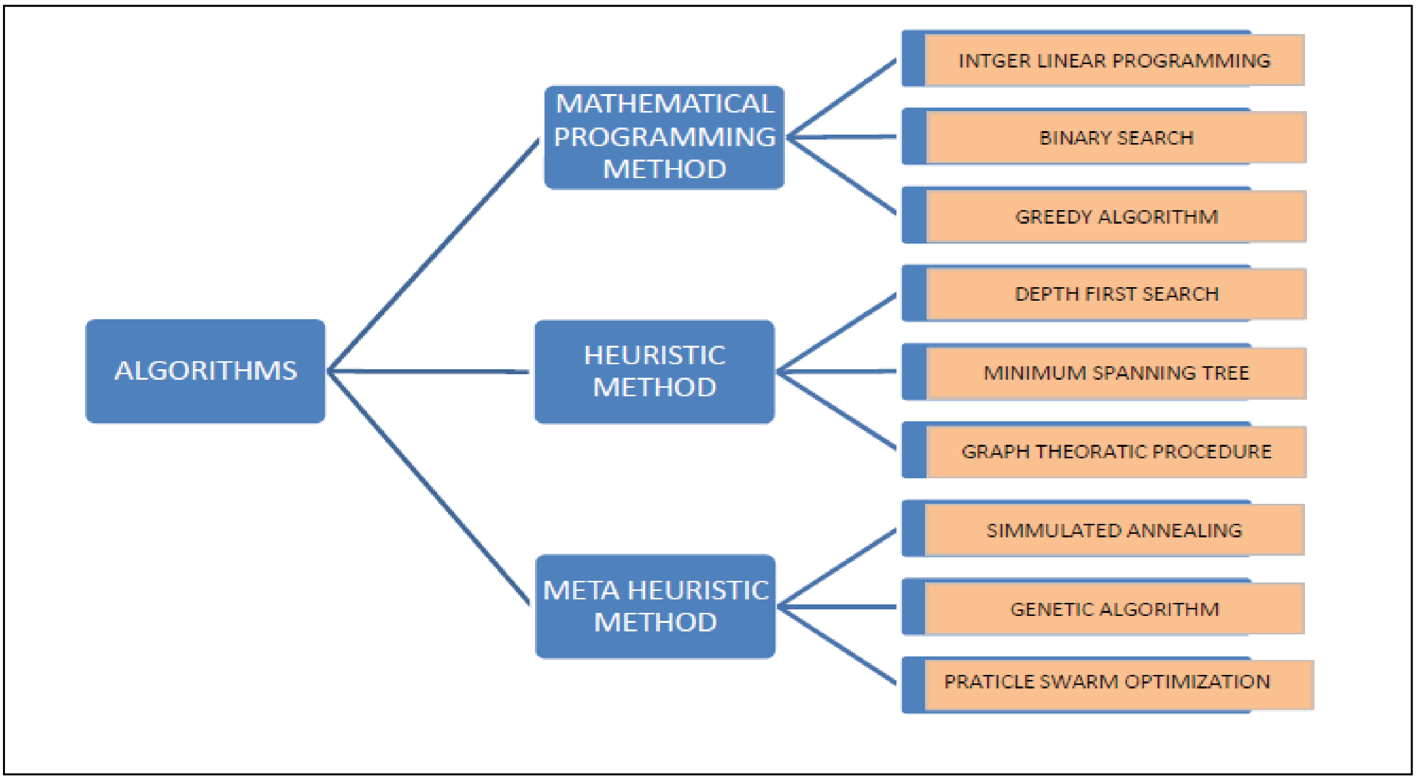

Fig 1. Optimum PMU Placement Algorithms

All Rights Reserved, @IJAREST-2016

1. INTEGER LINEAR PROGRAMMING

A linear programming (LP) problem in which all the design variable must take integer values is called linear integer programming problem.

The Steps to find the optimal solution is as follows:

- Step1. Form a bus connectivity matrix (A) from the network data.
- Step2. Find out the Isolated or Radial buses for the system. Also find buses connected to radial buses (BCRB).
- Step3. Remove the zero injection buses from the set of BCRB. Select remaining buses in the set as PMU placement bus.
- Step4. By using PMU bus selected in step3 find indirectly observable buses using KCL at zero injection bus.
- Step5. Find out Observability index of the each buses (OIB). OIB is the number of observable buses when placing PMU on respective bus. So for each bus value of OIB is different.
- Step6. Find out maximum value of OIB.i.e.OIBmax, and select the bus having maximum OIB.
- Step7. Form a set of buses having maximum OIB. Now remove zero injection buses from the set. If the number of remaining buses are greater than zero go to step8, else check whether there is any bus in set connected to zero injection bus connected to radial bus (ZIBCRB).If yes select that bus as PMU placement bus and go to step9.If no reduce value of maximum OIB by one and go to Step6.else find unobservable buses(UOB) and select any bus from ZIBCRB connected to UOB as PMU bus and go to Step9.
- Step8. If the number of buses found in step7 is one select the bus as PMU bus and go to step9.If the number is the more than one check for the non-zero injection bus connectivity among the buses obtained in Step7.
- Step9. Find directly observable buses and indirectly observable buses and find out total observable buses.
- Step10. If TOB is equal to the total number of the system bus then system becomes completely observable. Go to step11.Else if TOB<number of buses go to step5.
- Step11.a. Do pruning: Find the non-PMU buses which are connected to more than one PMU bus. For each combination place a PMU in the non-PMU bus and remove PMUs from the connected buses and check observability. If observability is satisfied, continue the process for other non-PMU buses else if observability is not satisfied do not prune for that non-PMU bus and continue the process for other non-PMU bus.
- Step11.b. Remove one PMU at a time from the set of PMUs obtained from the step 11(a) and check observability if found not observable do not remove that PMU, continue the process.

2. BINARY SEARCH

A Binary Search Algorithm, which was proposed, by Chakrabarti and Kyriakides was used to determine the minimum number of PMUs required. An Exhaustive Binary Search was employed to calculate the minimum number of PMUs needed to make system fully observable.

In general, a total number of combinations were able to be measured as

$$N_{solution} = \frac{P!}{N_{PMU}!(P - N_{PMU})!}$$

Where P represents the total number of buses in the power system and where NPMU represents the number of PMUs needed in the power system.

It is clear that the total number of buses in the power system will affect the total computational time in a non-linear relationship. For the purpose of reducing the execution period, the author introduced a theoretical upper bound of the minimum number of PMUs needed to make the system observable; this was derived [4].

$$N_{PMU}^{ub} = [(N + s/2)/3]$$

Where N is the total number of candidate buses in the system and s is the number of unknown power injections.

if the system is found to be unobservable for all of the combinations of PMU locations, then the minimum number of PMUs is increased by one, i.e., NPMU= (NPMU+1). If the system is found observable for any of the combinations of locations, the minimum number of PMUs is reduced by one, i.e., NPMU= (NPMU-1). The search is repeated until the minimum number of PMUs is obtained. The search process ensures that, if NPMU is the minimum number of PMUs, none of the solutions of length (NPMU-1) can make the system observable. An exhaustive set of combinations of size (NPMU-1) was examined for observability before concluding that NPMU is the minimum number of PMUs.

All Rights Reserved, @IJAREST-2016

3. GREEDY ALGORITHM

A combinatorial optimization algorithm that takes the best immediate, or local, solution while finding an answer is called greedy algorithm. Greedy Algorithm is a type of approximation algorithm used to solve the optimal question in a simple and effective way. Generally, Greedy Algorithm makes a locally optimal choice to lead to a globally optimal solution. Basically, this algorithm makes decisions according to a single rule which states that, at each stage, a PMU should be installed at the bus that covers the largest number of uncovered buses and this process should be repeated until the system is fully observed.

In Greedy Algorithm initial step, is to put PMU on the highest connected buses with a highest connectivity degree of in the network, and in the second step, the algorithm searches for the bus that covers the largest number of uncovered buses. Hence the network is fully observed and the Greedy Algorithm will be terminated.

4. DEPTH FIRST SEARCH

Depth First Search method (DFS) is applied extensively in earlier time, which is one of the tree search methods of PMU placement. It marks all vertices in a directed graph in the order they are discovered and finished, while partitioning the graph into a forest, this method uses only three rules. The first PMU is placed at the bus with the largest number of connected branches if there is more than one bus with this characteristic, one is randomly chosen. The following PMUs are placed with the same criterion, until the complete network visibility is obtained. DFS merely considered the "depth" through the process of expanding, which makes the observational topologies, increases the unwanted redundancy.

5. MINIMUM SPANNING TREE

Minimum Spanning Tree method (MST) is a modified depth first approach. The procedure is given in the flow chart.

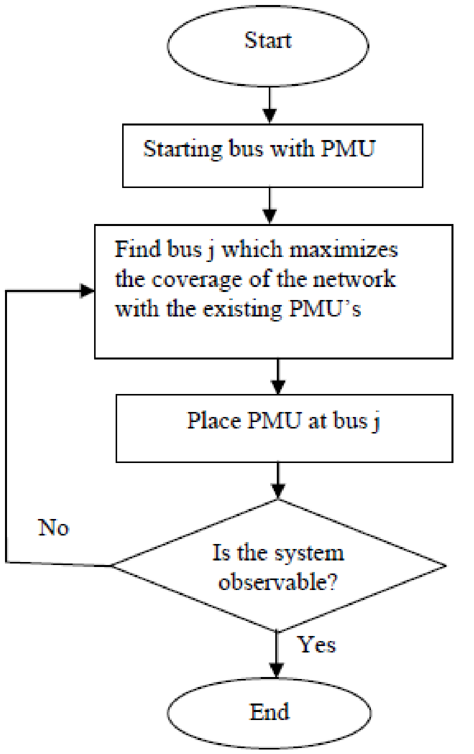

Fig 2. Flow Chart of Minimum Spanning Tree Method

6. GRAPH THEORATIC PROCEDURE

The dual search method can be greatly accelerated by beginning the search near the minimal solution. A reasonable starting point can be provided by a graph theoretic search which identifies a placement sets that makes the system observable. The steps for the method are as follows:

Step 1: Place a PMU at the bus with the highest number of incident lines in the unobservable region;
Step 2: Determine the region observed by the current PMU placement set.
Step 3: If the observed region does not cover the whole system, then go to Step 2; otherwise stop.

All Rights Reserved, @IJAREST-2016

7. SIMMULATED ANNEALING

Simulated annealing method is put forward by Metropolis in 1953. Simulated Annealing (SA) is a technique that finds a good solution to an optimization problem, by trying random variations of the current solution. A worse variation is accepted as the new solution with a probability that decreases as the computation proceeds. The slower the cooling schedule, or rate of decrease, the more likely the algorithm is to find an optimal or near-optimal solution. It solves the problems of combination optimization by simulating the physical anneal process of the solid matter (such as metal). By analogy with this physical process, each step of the SA algorithm replaces the current solution by a random" nearby" solution, chosen with a probability that depends both on the difference between the corresponding function values and also on a global parameter T (called the temperature), that is gradually decreased during the process. The dependency is such that the current solution changes almost randomly when T is large, but increasingly "downhill" as T goes to zero [6].

The procedures can be subdivided into several main steps, as follows:

(1) Select an initial condition $X0$ from the approve solution space randomly, calculate its target function value $f(X0)$, and select the initial control temperature $T0$ and the length of Markov Chain;

(2) Engender a random disturbance in the approve solution space, then gain a new state $X1$, calculate its target function value $f(X1)$;

(3) Judge, whether it satisfies $f(X1) < f(X0)$, if yes, then accept the new state $X1$ as the current state; if not, then judge whether it satisfy $X1$ according to Metropolis rule, if yes, then accept the new state $X1$ as the current state, else accept the state $X0$ as the current state;

(4) Judge, whether the sample process ends according to some convergence rule, if yes, then turn to (5); if not, then turns to (2);

(5) lower the control temperature T according to some annealing project;

(6) Judge, whether the annealing process ends according to some convergence rule, if yes, then turn to (7); if not, then turns to (2);

(7) Output the current solution as the best solution.

8. GENETIC ALGORITHMS

GA was inspired by the natural evolution of species. In natural evolution, each species searches for beneficial adaptations in an ever-changing environment. As species evolve, new genetic information is encoded in the chromosomes. This information changes by the exchange of chromosomal material during breeding (crossover) and also mutation. From the engineering standpoint, if we have two solutions with good approximation for a given problem, their combination might lead to a better solution. So, GA pertains to the search algorithms with an iteration of generation-and-test. With the characteristics of easier application, greater robustness, and better parallel processing than most classical methods of optimization, GA has been widely used for combinatorial optimization, structural designing, machine learning rule-based classifier systems etc.

9. BINARY PRATICLE SWARM OPTIMIZATION

BPSO is a swarm intelligence based optimization which was first introduced by Kennedy & Eberhart (1995). It's taken from the social behaviour and collective movements of birds and shoals of fish. Swarm intelligence comes from the study of collective behaviour in decentralized, self-originated systems like a swarm of social organisms, collective intelligence arises from interactions.

A number of agents or particles are employed in finding the optimal solution is guided by both individual and social knowledge of the particles.

The steps followed in BPSO are given below

1. Initialize the swarm, i.e. initial value of particle position and velocities.
2. Calculate the value of objective function i.e. and calculate personal (*pbest*) and global best (*gbest*).
3. Now calculate new values of velocity and position.
4. Calculate the value of objective function.
5. Sort the elements of the swarm according to their value of cost function.
6. For the highest values of the cost function, i.e. worst possible solutions, mutation is Introduced. For each position of element string, a random number is generated and if it is less than the predefined mutation constant, *dth* value alters from 0 to 1 or 1 to 0.
7. Update *fbest* (best designated value of cost function) and *gbest* (global best solution) and repeat process from step 3 until the convergence criteria is fulfilled.

All Rights Reserved, @IJAREST-2016

III. SYSTEM OBSERVABILITY REDUNDANCY INDEX (SORI)

In optimal PMU placement, the redundancy index is an important factor for representing the stability of the power network. Due to a multiple number of optimum solutions being available after applying the optimization algorithm, Bus Observability Index (BOI) will be implemented to indicate the performance on quality of optimization. In BOI, bus-i (βi) will be defined as the number of PMUs which are able to observe a given bus. Consequently, the maximum bus observability index is limited to maximum connectivity (ηi) of a bus plus one:

$$\beta i \leq \eta i + 1$$

In order to select the most favourable outcomes among a number of optimal solutions obtained using different optimization methods, the System Observability Redundancy Index (SORI) is, in principle, a measurement of the sum of bus coverage for all the implemented buses (i=1 to n) in an active system. Higher SORI value indicates that the PMU-based monitoring system is more The SORI can be calculated using Equation 2.11, where γ represents System Observability Redundancy Index

$$\gamma = \sum_{i=1}^{n} \beta_i$$

IV. MODELING & SIMULATION

1. IEEE-6 BUS SYSTEM PSAT MODEL

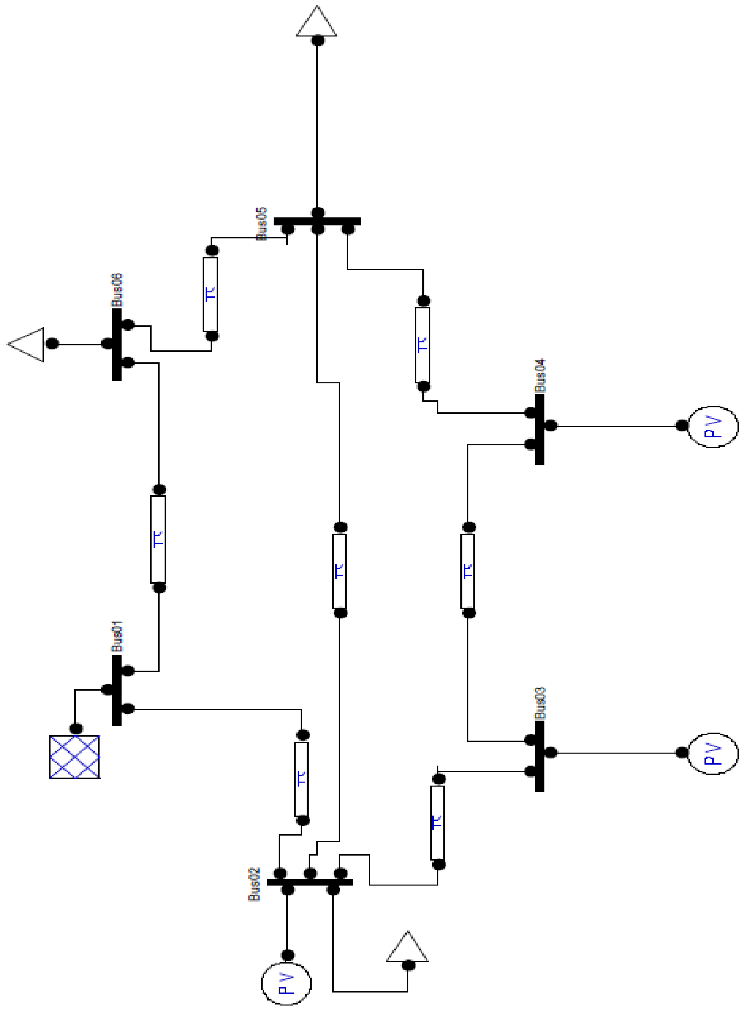

Fig 3. IEEE-6 BUS PSAT Model

All Rights Reserved, @IJAREST-2016

2. IEEE-6 BUS RESULT COMPARISION TABLES:

SR. NO.	ALGORITHMS	NO. OF PMU's	BUS LOCATION
1	Depth First Search	3	1,3,5
2	Graph Theoretic Procedure	3	1,3,5
3	Minimum Spanning Tree	3	1,3,5
4	Simulated Annealing	2	1,4
5	Linear Integar Programming	2	2,5
6	Binary Search	2	2,5
7	Greedy Approach	2	2,5
8	Genetic Algorithm	2	2,5
9	Modified Binary Particle Swarm Optimization	2	2,5

Table 1: Different Algorithms Result Comparison Table for IEEE-6 Bus System

Sr. No.	Algorithms	SORI
1	Depth First Search	9
2	Graph Theoretic Procedure	9
3	Minimum Spanning Tree	9
4	Simulated Annealing	6
5	Linear Integar Programming	8
6	Binary Search	8
7	Greedy Approach	8
8	Genetic Algorithm	8
9	Modified Binary Particle Swarm Optimization	8

Table 2: System Observability Redundancy Index Comparison Table for IEEE-6 Bus System

All Rights Reserved, @IJAREST-2016

3. IEEE-14 BUS SYSTEM PSAT MODEL

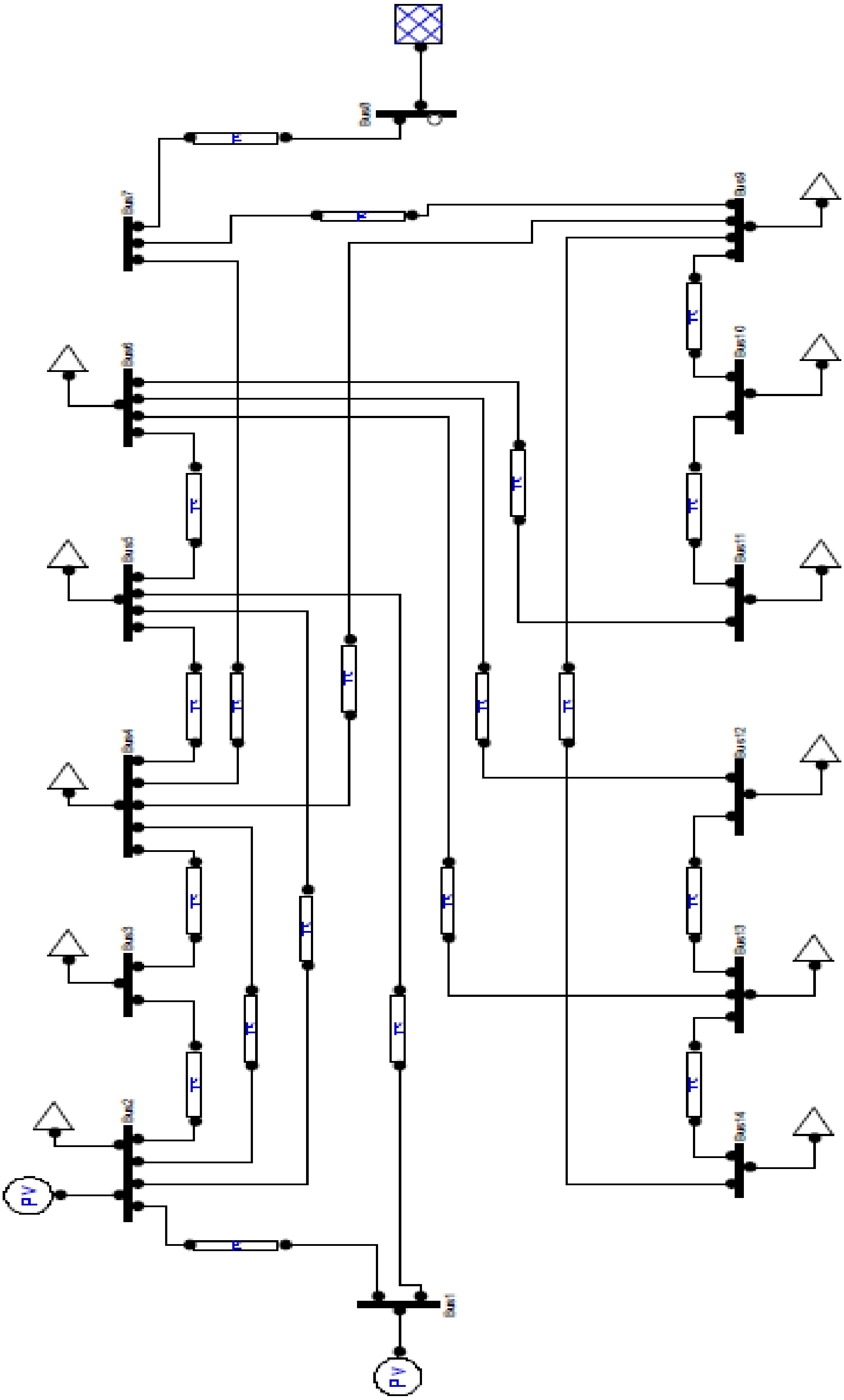

Fig 4. IEEE-14 BUS PSAT Model

All Rights Reserved, @IJAREST-2016

4. IEEE-14 BUS RESULT COMPARISION TABLES:

Sr. No.	Algorithms	No. of PMUS's	PMU Bus Location
1	Depth First Search	6	1,4,6,8,10,14
2	Graph Theoretic Procedure	6	1,4,6,8,10,14
3	Minimum Spanning Tree	6	1,4,6,8,10,14
4	Simulated Annealing	4	2,6,7,9
5	Linear integer Programming	4	2,6,7,9
6	Binary Search	4	2,6,7,9 / 2,7,10,13 / 2,7,11,13
7	Greedy Approach	4	2,6,7,9
8	Genetic Algorithm	4	2,6,7,9
9	Modified Binary Particle Swarm Optimization	4	2,6,7,9

Table 3: Different Algorithms Result Comparison Table for IEEE-14 Bus System

Sr. No.	Algorithms	SORI
1	Depth First Search	18
2	Graph Theoretic Procedure	18
3	Minimum Spanning Tree	18
4	Simulated Annealing	19
5	Linear integer Programming	19
6	Binary Search - 2,6,7,9	19
	Binary Search - 2,7,10,13	16
	Binary Search - 2,7,11,13	16
7	Greedy Approach	19
8	Genetic Algorithm	19
9	Modified Binary Particle Swarm Optimization	19

Table 4: System Observability Redundancy Index Comparison Table for IEEE-14 Bus System

All Rights Reserved, @IJAREST-2016

V. CONCLUSION

In this Paper, A comparative study of different methods of optimum PMU placement technique are shown. Different Numerical, Heuristic & Meta heuristic methods are applied to IEEE-6 & IEEE-14 bus system. Different techniques are implemented in the PSAT. SORI & minimum number of PMUs with their location are studied. Effect of zero injection are also consider for all the studies. A comparative study of all the methods are tabulated in this paper.

REFERENCES

1. B. K. Saha Roy, A. K. Sinha, A. K. Pradhan "Optimal Phasor Measurement Unit Placement for Power System Observability -A Heuristic Approach" Institute of Electrical and Electronics Engineers (IEEE), 2011.
2. J. S. Bhonsle1, A. S. Junghare "A Novel Approach for the Optimal PMU Placement using Binary Integer Programming Technique" International Journal of Electrical and Electronics Engineering (IJEEE) ISSN (PRINT): 2231 – 5284, Vol-1, Iss-3, 2012
3. Satyendra Pratap Singh and S.P. Singh "Optimal PMU Placement in Power System Considering the Measurement Redundancy" Advance in Electronic and Electric Engineering. ISSN 2231-1297, Volume 4, Number 6 (2014), pp. 593-598
4. Saikat Chakrabarti, Elias Kyriakides "Optimal Placement of Phasor Measurement Units for Power System Observability" IEEE transactions on power systems, vol. 23, no. 3, august 2008.
5. Qiao Li, Rohit Negi, Marija D. Ili´c "Phasor Measurement Units Placement for Power System State Estimation: A Greedy Approach" Institute of Electrical and Electronics Engineers (IEEE), 2011.
6. A. Y. Abdelaziz, Amr M. Ibrahim, Reham H. Salem "Power system observability with minimum phasor measurement units placement" International Journal of Engineering, Science and Technology Vol. 5, No. 3, 2013.
7. T. L. Baldwin L. Mili M. B. Boisen, Jr. R. Adapa "Power System Observability
8. With Minimal Phasor Measurement Placement" Institute of Electrical and Electronics Engineers (IEEE), 2003.
9. M. Hajian, A. M. Ranjbar, T. Amraee, and A. R. Shirani "Optimal Placement of Phasor Measurement Units: Particle Swarm Optimization Approach" Institute of Electrical and Electronics Engineers (IEEE), 2005.

All Rights Reserved, @IJAREST-2016