PARTICLE SWARM OPTIMIZATION TECHNIQUE FOR OPTIMIZING LOAD FLOW ANALYSIS

Muhammad Syafiq B. Mohd Zaman
Faculty of Electrical Engineering
Universiti Teknologi MARA, 40450 Shah Alam Selangor, Malaysia.

Abstract – The state of power system and methods of calculating this state are extremely important in evaluating the operation of the power system, the control of this system and the determination of future expansion for the power system. The state of power system is determined through load flow analysis that calculates the power flowing in the lines of the power system especially in determining the total losses in the power system. Particle swarm optimization (PSO) is a stochastic global optimization algorithm inspired by social behavior of bird flocking in search of food, which is simple but powerful, and widely used as a problem solving technique to a variety of complex problems in science and engineering. The PSO technique was proposed to solve on Institute of Electrical & Electronic Engineers (IEEE) 30-bus system. The proposed technique was able to minimize the losses in the power system with results of

Key Words : Particle Swarm Optimization, Load Flow Analysis, Line Losses.

I. INTRODUCTION

The computational solution of load flow has attracted much attention. Continuing research on numerical methods for load flow calculation is nevertheless justified by the considerable expenditure of computer resources and engineering effort on load flow analysis. Reduction of execution times allows the more effective use of computers, and increased robustness and accuracy contribute to the value of the analysis tool [2]. The recent introduction of scientific work-station computers, which provide network planning or operating engineers with integrated analysis and display systems, further emphasizes the need for fast and reliable load flow software. Improvements in solution accuracy and the ability to solve numerically difficult network problems permit the user to concentrate on the physical network rather than on its suitability for numerical analysis. Researchers, however, have been aware of the shortcoming of the classical solution algorithms i.e. N-R(N-R) and Fast Decoupled Load Flow(FDLF) when they are generically implemented and applied to ill-conditioned and/or poorly initialized power system.

Hence commercial power flow packages always modify these algorithms for enhanced robustness. The most popular method (FDLF) cannot handle Q-limit violation easily. The Gauss-Seidel (G-S) power flow technique, another classical power flow method, has been shown to be extremely inefficient in solving large power systems as well as ill-conditioned ones, but it can handle bus violations with ease [1]. Omine reduction of the static power system model has been widely used to decrease the computational burden of the network solution. Many methods have been developed to compute the actual reduction, and network reduction programs are used in industry today [2]. One method of network reduction is to eliminate all the P-Q buses and retain only P-V buses; then P-V bus data is used in the iteration cycle to restore new values for P-Q bus voltages. This method for load flow analysis saves computer time [3] but it does not, however, take into account Q-limit violations, and its accuracy is not good because of the many approximations included. This method is simple, reliable, fast, and, compared with other techniques, can handle the adjusted solution with ease. The sparsity is exploited in the reduction step, and is very useful for offline and online applications.

Traditionally, load flow analysis were calculated using the Gauss-Seidel Method or Optimal Load Flow (or N-R) method. The first two load flow methods (Gauss Seidel and N-R) require the determination of an admittance matrix. This can pose a serious problem when the matrix is sparse and cannot be inverted, which is the case in many applications with the Optimal Load Flow. In Optimal Load Flow, we determine the line losses of the power system by N-R Method.

The Particle Swarm Optimization (PSO) developed by Eberhart and Kennedy in 1995 that simulates the swarming behavior of animals in nature [1]. Swarming animals (such as ants and bees) are capable of performing only basic tasks individually. However, as a swarm, they exhibit some sort of intelligently organizational behavior, making them capable of performing complex tasks in a teamwork-like fashion. PSO has a successful track record in solving many optimization problems [4-8].

Since PSO agents (particles) are parallel in nature, PSO allows efficient and fast optimization of the problem [9]. Since PSO agents (particles) are parallel in nature, PSO allows efficient and fast optimization of the problem [9]. Furthermore, PSO requires only basic mathematical operators to perform optimization [2, 10]. Another benefit of PSO is that it requires low, constant computational and memory costs for each iteration [9-11]. PSO is motivated from this scenario and developed to solve complex optimization problems [7]. In this paper, the particle swarm algorithm was used to perform the optimization to minimize the losses in power system.

The organizational of this paper is as follows:

- A review on the particle swarm optimization algorithm in section II
- A review of load flow analysis in section III
- A review of project objectives in section IV
- A review of methodology in section V
- A review of results and discussion in section VI
- Finally a conclusion is represented in section VII

II. PARTICLE SWARM OPTIMIZATION

In the original form of PSO, each in a swarm population adjusts its position to search space based on the best position it has found so far, and the position of the known best fit particle in the entire population. The essence of PSO is to use these particles with the best known positions to guide swarm population to converge to a single optimum in the search space. Unlike other population based Evolutionary algorithms i.e., genetic algorithms, PSO does not need genetic operators such as crossover or mutation [14]. Thus it has advantages of easy implementation, fewer parameters to be adjusted, strong capability to escape from local optima as well as rapid convergence. In addition, because the PSO comprises a very simple concept and paradigms can be implemented more easily. With it, it has been demonstrated in certain instances that PSO outperforms other population based evolutionary computing algorithms in many practical engineering domains.

In recent years, PSO has been used increasingly as an effective technique for solving complex and difficult optimization problems. PSO has been successfully applied to function optimization, artificial neural network training, fuzzy system control, power system problems and many more. Therefore, PSO has also been found to be robust and fast in solving the non-linear, non-differentiable and multi modal problems [5]. In this paper, the load flow analysis in power system is introduced in PSO to optimize the load bus in power system.

III. LOAD FLOW ANALYSIS

In normal load flow problems, the N-R method for performing the load flow calculation was used. Taylor series expansion for a function of two or more variables is the basis of the N-R method. Partial derivatives of order greater than 1 are neglected in the series terms of the Taylor series expansion. The N-R method was use because it calculates corrections while taking into account all other interactions. The number of iterations required by the N-R method using bus admittances is practically independent of the number of buses. For these reasons shorter computer time for a solution of the load flow problem could occur when analyzing large electrical power systems.

The solution of the load flow problem is initiated by assuming voltage values for all buses except the slack bus. The slack bus is the point at which the voltage is specified and remains fixed. The voltage at the slack bus is fixed because the net power flow of the system cannot be fixed in advance until the load flow study is complete. The power calculation at the slack bus supplies the difference between specified real power into the system at the other buses and the total system output plus losses. The N-R method for load flow analysis will be used to solve the load flow problem at any value of bus.

IV. PROJECT OBJECTIVES

The project objectives are as follows:

- To determine the variables that effect the line flow losses
- To minimize the losses
- To optimize the system by particle swarm optimization technique

V. THEORY OF PSO

A. Particle Swarm Optimization with Constriction Factor (PSO$_{CF}$)

PSO is a population-based stochastic optimization technique inspired by animal swarming behavior in nature [5]. PSO iteratively searches for solutions in the problem space by taking advantage of the cooperative and competitive behavior of simple agents called particles.

For the Particle Swarm Optimization (PSO) method presented here, considered the PSO with Constriction Factor (PSO$_{CF}$) method. PSO$_{CF}$ modifies the original PSO algorithm to improve its convergence properties, by gradually decreasing particle velocities as the iteration progresses, so that particle movements near the optimum are localized.

The PSO algorithm search is directed by its velocity equation:

$$V_{id} = V_{id} + C_1(pBest - X_{id}) \times rand_1 + C_2(gBest - X_{id}) \times rand_2 \quad (1)$$

which modifies the particle's position, X_{id}:

$$X_{id} = X_{id} + V_{id} \quad (2)$$

where:
V_{id} = particle velocity.
X_{id} = particle position.
pBest = particle's best fitness so far.
gBest = best solution achieved by the swarm so far.
C_1 = cognition learning rate
C_2 = social learning rate.
$rand_1$, $rand_2$ = random numbers between 0 and 1.

where χ is calculated using:

$$\chi = \frac{2}{2 - \varphi - \sqrt{\varphi^2 - 4\varphi}} \quad (3)$$

(6)

and φ must conform to:

$$\varphi = C_1 + C_2, \varphi > 4 \quad (4)$$

Parameter *vMax* is the allowable maximum velocity during optimization. It acts as a constraint for prevention of velocity explosion [6, 7]. Generally, for each variable, *vMax* is set to a dynamic range of values [18].

Conversely, the value of X_{id} may be bounded using parameters *xMin* and *xMax* to disregard solutions outside an acceptable range [19]. Whenever X_{id} violates *xMin* or *xMax*, they are artificially brought back to their nearest side constraint (either *xMin* or *xMax*). Additionally, the velocity V_{id} is set to 0 each time this occurs to discourage further searches in that direction [19].

The swarm size is problem-specific, and there has been no literature recommendations regarding swarm size [19]. However, most researchers tend to utilize swarm size of around 10 to 50 particles to solve the given problem [19], while others may use more [10].

The algorithm for the PSO is detailed.

```
While (objective not met OR maximum iterations not reached)
  For each particle:
    Perform Vid update as directed by Eq.(1).
    Modify Xid according to Eq.(2).
    If (Xid > xMax)
      Change Xid = xMax.
      Set Vid = 0.
    ElseIf (Xid < xMin)
      Change Xid = xMin.
      Set Vid = 0.
    End
    Evaluate fitness of particle.
    If (fitness better than gBest)
      Update gBest with fitness.
    Else
      Don't update gBest.
    End
  Repeat for next particle.
End
```

VI. THEORY OF LOAD FLOW ANALYSIS

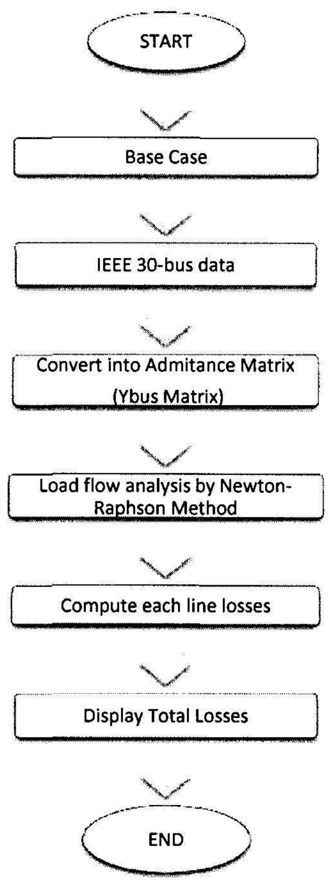

Figure 1: Flow Chart of Load Flow Analysis

VII. METHODOLOGY

In this paper, the Particle Swarm Optimization Technique considers the value of load bus in 30-bus IEEE test system. The PSO program was developed using MATLAB programming language. IEEE 30-bus system consists of 30 generator and 42 lines that connect between generator and load bus with transmission line.

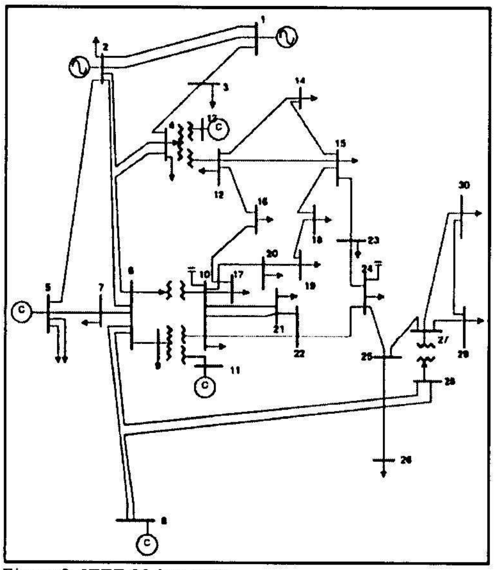

Figure 2. IEEE 30-bus system

All tests are run on an Acer Aspire 1692 computer with Intel Centrino M Central Processing Unit (CPU) running at 1.73 GHz with 2.00 GB of Random Access Memory (RAM). Microsoft Windows XP Professional Service Pack 3 was installed as the operating system. All programs are implemented in the MATLAB version 7.6.0.324 (R2008a) environment.

The optimization results of the PSO algorithm depends on the initial value of the particles prior to optimization. To investigate the effectiveness of the proposed method, the experiment was repeated 100 times for each particle with different initialization values. The random number generated should be similar to [10]. However, as the random number generation method in [10] was not described, the pseudo-random number generator called the MTA [21, 22], was used for the generation of random numbers.

In MTA, the sequence of random numbers generated is determined by the internal state of the generator. Each state will have unique computations and outcomes. The unique computations result in the generation of unique series of random numbers based on the state. To ensure repeatability of the experiments, the generator state is set to some fixed value each time the optimization executes to ensure that the same set of random numbers are generated.

For the PSO algorithm, the constriction factor method was used. The values of C_1 and C_2 were both set to 2.05 [23], since the values cannot violate the rule set in Eq.(4). Based on the values of C_1 and C_2, the value of χ is 0.7290 throughout the optimization course.

Next, the values of *xMin* and *xMax* were set to 0 and 1, respectively. This was done so that the particle values are always between 0 and 1. Further, since we have set *xMin* and *xMax* to 0 and 1, respectively, the dynamic range of *vMin* and *vMax* were respectively set to -1 (when V_{id} moves from 1 to 0) and +1 (when V_{id} moves from 0 to 1).

The objectives for the DPSO algorithm were set according to the optimal value of the fitness functions.

VIII. RESULTS AND DISCUSSION

Consider the initial bus data in IEEE 30-bus data, it consists of Real Power (MW) and Reactive Power (MVar). A 30 bus system as shown in figure 2 including three phase line section. The value obtained by PSO program was totally different from the initial value. The value given by PSO was smaller than the initial value. It's mean that the analysis using PSO gave the best value to minimize the losses in the power system. The analysis by load flow without PSO and after PSO was shown in Table I. The table including of losses obtained by load flow analysis without PSO and after optimized by PSO.

TABLE I
COMPARISON OF LOAD BUS DATA BEFORE AND AFTER OPTIMIZATION

No of Bus Data	Initial	**PSO**	Initial	**PSO**
	MW	**MW**	**Mvar**	**Mvar**
1	0	0.0862	0	0.9622
2	21.7	0.3261	12.7	0.9050
3	2.4	0.7806	1.2	0.6060
4	7.6	0.2043	1.6	0.2754
5	94.2	0.2603	19	0.1689

6	0	0.2158	0	0.1615
7	22.8	0.1709	10.9	0.8670
8	30.0	0.2313	20.0	0.4215
9	0	0.8698	0	0.3551
10	5.8	0.3242	2.0	0.7193
11	0	0.5845	0	0.3326
12	11.2	0.1030	7.5	0.9077
13	0	0.8843	0	0.5683
14	6.2	0.7711	1.6	0.4093
15	8	0.7229	2.5	0.5494
16	3.5	0.3330	1.8	0.7356
17	9.0	0.4840	5.8	0.5643
18	3.2	0.4840	0.9000	0.2267
19	9.5	0.0018	3.4	0.5490
20	2.2	0.4787	0.7000	0.8728
21	17.5	0.9312	11.2	0.3986
22	9.0	0.1175	0	0.0228
23	3.2	0.0587	1.6	0.3663
24	8.7	0.4946	6.7	0.5059
25	0	0.6831	0	0.7289
26	3.5	0.6721	2.3	0.7251
27	0	0.1888	0	0.6278
28	0	0.9194	0	0.7745
29	2.4	0.5092	0.9000	0.4087
30	10.6	0.0133	1.9	0.6434

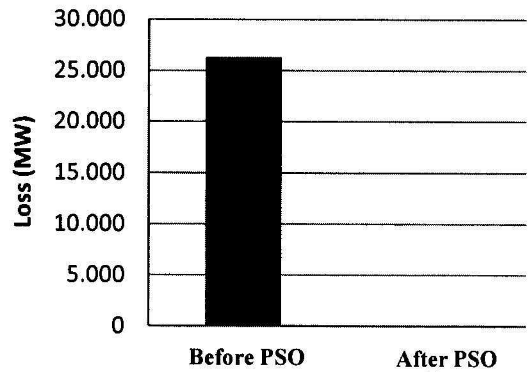

Figure 3. Comparison of total loss before PSO and after PSO

From figure 3, the total loss before PSO is about 26.331MW. After optimization by PSO, it can be seen that the losses have been minimized to minimum which is closely to zero. In our real life, the total losses after any optimization cannot be achieve to zero because of many causes i.e. transmission line losses.

For each dimension, the experiments were repeated 100 times with different random initialization values. For the first repetition, the initial Mersenne-Twister algorithm (MTA) state begins from 0 and increased in step of 50,000 for the next repetitions. For example, the first repetition of the initial MTA state is set to 0 and for the next repetition, the initial MTA state setting is at 50,000 and this continues until the 100th repetition. This is done to evaluate the convergence with different initial particle values.

IX. CONCLUSION

In this paper, load flow analysis has been performed using N-R method. PSO technique has been developed for optimizing the load flow analysis by giving the best value of load bus data to minimize the losses. As compared with the value of initial load bus data before PSO and after PSO in Table I conclude that the PSO values give the optimized solution in the power system analysis as shown in figure 1. The project described in this paper can be extended to analyze any of bus data.

X. ACKNOWLEDGEMENT

The author would like to express his deepest gratitude and appreciation to Project Supervisor, En Ahmad Ihsan B. Mohd Yassin for his invaluable suggestions, guidance, comments and advice for the completion of this project.

He also likes to express his gratitude to all his friends who help him upon completion this project.

XI. REFERENCES

[1] G.M Gillbert, J. J. Grainger, and S. H. Lee, "A Comparison Of Load Flow Analysis Using Distflow, Gauss-Seidel, and Optimal Load Flow Algorithm.", IEEE Transactions on Power Delivery, vol. 3, No. 3, July 1988 , pp. 1217-1223.

[2] A. Merlin and H. Back. "Search for a Minimal-Loss Operating Spanning Tree Configuration in Urban Power Distribution Systems", Proc. of 5th Power Systems Comp. Con., Cambridge, U.K., Sept. 1-5, 1975.

[3] D. W. ROSS, M. Carson, A. Cohen et al., "Development of Advanced Methods for Planning Electric Energy Distribution Systems", DE0 final report no SCI-5263, Feb 1980.

[4] H. H. Balci and J. F. Valenzuela, "Scheduling electric power generators using particle swarm optimization combined with the lagrangian relaxation method," *Int. J. Appl. Math Comput. Sci.*, vol. 14, pp. 411-421, 2004.

[5] J.S. Heo, K.Y. Lee, and R. Garduno-Ramirez, "Multiobjective control of power plants using particle swarm optimization techniques," IEEE Trans. on Energy Conversion, vol. 21, pp. 552-561, 2006.

[6] L. Khriji and K. El-Metwally, "Rational-based particle swarm optimization for digital image interpolation," Int. J. Intelligent Technology vol. 1, pp. 223-227, 2006.

[7] M.G. H. Omran, A. P. Engelbrecht, and A. Salman, "Dynamic clustering using particle swarm optimization with application in unsupervised image classification," Trans Engineering, Computing and Technology, vol. 9, pp.199-204, 2005. A. O.

[8] J. Kennedy, R.C. Eberhart, "Particle swarm optimization,"IEEE Trans. Neural Networks, vol. 4, pp. 1942–1948, 1995.

[9] K. E. Parsopoulos and M. N. Vrahatis, "Recent approaches to global optimization problems through Particle Swarm Optimization," Natural Computing, vol. 1, pp. 235-306, 2002.

[10] R.C. Eberhart, J. Kennedy, "A new optimizer using particle swarm theory," in proc. 4th international symposium on micromachine and human science, Nagoya, Japan, 1995, pp. 39–43.

[11] M. Senthil Arumugam, M.V.C. Rao, "On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems," Applied Soft Computing, vol.8, pp. 324–336, 2007.

[12] Y.H. Shi, R.C. Eberhart, "A modified particle swarm optimizer," in Proc. IEEE Trans. Evolutionary Computation, pp. 69-73, 1998.

[13] Y. Shi, R.C. Eberhart, "Empirical study of particle swarm optimization," in Proc. IEEE Trans. Evolutionary Computation, pp. 6–9, Jul. 1999.

[14] Y.H. Shi, R.C. Eberhart, "Fuzzy adaptive particle swarm optimization," in Proc. IEEE Trans. Evolutionary Computation, pp. 101–106, 2001.

[15] E. C. Laskari, K. E. Parsopoulos, and M. N. Vrahatis, "Particle swarm optimization for integer programming," in Proc. of the IEEE 2002 Congress on Evolutionary Computation 2002.

[16] M. G. H. Omran, A. Engelbrecht, and A. Salman, "Barebones particle swarm for integer programming problems," in Proc. IEEE Swarm Intelligence Symposium, 2007.

[17] X. Hu and R. C. Eberhart, "Solving constrained nonlinear optimization problems with particle swarm optimization," in 6th World Multiconf. on Systemics, Cybernetics and Informatics, Orlando, FL, 2002.

[18] R. Hassan, B. Cohanim, O. d. Weck, and G. Venter, "A Comparison of Particle Swarm Optimization and the Genetic Algorithm," in 1st AIAA Multidisciplinary Design Optimization Specialist Conf. vol. AIAA-2005-1897 Austin , TX, 2005

[19] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudorandom Number Generator," ACM Transactions on Modeling and Computer Simulation, vol. 8(1):3, pp. 3-30, 1998.

[20] MATLAB Function Reference Documentation. Mathworks Inc., Natick, MA, 2008.

[21] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in Proc. Congress of Evolutionary Computation San Diego, CA, 2000, pp. 84-88.

[22] Ihsan M. Yassin, Mohd N. Taib and Nooritawati Md. Tahir, "Modification Of Particel Swarm Optimization Algorithm for Optimization Of Discrete Values," in IEEE Trans. on