A Quasi Newton Approach for Optimal Generation Scheduling

Leena Heistrene[1], Poonam Mishra[2], Makarand Lokhande[3]

[1]*Electrical Engg. Dept., SOT, PDPU*
[2]*Mathematics Dept., SOT, PDPU*
[3]*Electrical Engg. Dept., SVNIT*

***Abstract* — Optimal generation scheduling problem has been tackled by researchers in a very broad perspective, ranging from the problem dimension issue with respect to the large number of design variables in a practical scenario, its spatial temporal constraints, robustness of the optimization algorithm etc. Accordingly, efforts have been directed towards coming up with a holistic solution that caters to the needs of an online fast accurate optimized economic dispatch solution. This paper has applied the Quasi Newton approach to the optimal generation scheduling problem owing to its robustness and low computer time to termination. Optimal solutions are presented for the IEEE 30 bus, 6 generator system and its solutions are compared with four other optimization methods namely sequential quadratic programming, interior point method, pattern search and genetic algorithm. Comparative results demonstrate the efficiency of the method when applied to the optimal generation scheduling problem.**

Index Terms* — *Optimal generation scheduling, Quasi Newton method, hessian matrix updation.

NOMENCLATURE

P_i	Real power generated by generator *i*
N	Total number of generators
a_i, b_i, c_i	Cost coefficients of generator *i*
$P_{i, \min}$	Minimum generation output of generator *i*
$P_{i, \max}$	Maximum generation output of generator *i*
$L_{i, \min}$	Lower transmission flow limit
$L_{i, \max}$	Upper transmission flow limit
$G_{k,i}$	Generation distribution shift factor of unit *i* to transmission interface k
$R_{i, \min}$	Minimum ramping rate limit of generator *i*
$R_{i, \max}$	Maximum ramping rate limit of generator *i*
λ	Lagrangian multiplier of power balance constraint
A_k	Approximate hessian matrix formed in the Quasi Newton method at the k^{th} iteration.
Δg_k	Difference in the gradient matrix from $(k-1)^{th}$ to the k^{th} iteration in the Quasi Newton method.

I. INTRODUCTION

The optimal generation scheduling problem, also known as the economic load dispatch problem, has been an area of interest for both researchers and power engineers from the advent of large power systems consisting of several alternators of different fuel sources. Different countries have adopted different optimization methods for generation scheduling of its respective power systems - right from the brute force techniques such as priority list methods, to the more advanced methods such as artificial intelligence (AI) techniques. Researchers have worked their way through from conventional methods too, based on calculus and search gradients techniques towards hybrid techniques involving a blend of conventional methods and AI based methods. Before this issue of optimization is addressed, the work done so far for the optimal generation scheduling/ economic load dispatch (ELD) problem is discussed below.

A typical ELD problem consists of minimizing the fuel cost function subject to equality constraint, also known as the power balance constraint, and inequality constraints such as generation limit constraint, transmission line flow constraint, ramp rate limits, etc. [1] This optimization problem has been solved by researchers by conventional methods in the past such as Newton Raphson method [2], Quadratic programming [3] and other gradient search based methods. But each of these has their own drawbacks such as the limitation of the Newton Raphson method being in acquiring a positive definite hessian matrix in every iteration. This is not practical in every case for two reasons. First of all, getting a positive definite hessian matrix may not be possible due to the system parameters involved in the case and secondly, even if a positive definite hessian matrix is achieved, it is computationally cumbersome to calculate the inverse of the hessian matrix in every iteration, especially when the system considered is large. Linear programming [4] and Dynamic programming [5] have also been adopted for solving this optimization problem. The limitation of linear programming is its requirement for a linear objective function which results in compromising with the accuracy of the fuel cost function whereas dynamic programming suffers from the curse of dimensionality. In recent times, artificial intelligence techniques have also been used such as genetic algorithm[6], simulated annealing[7] and particle swarm optimization. Such techniques have an advantage of being able to work with discontinuous objective functions and give global solutions but they suffer from the disadvantage of being prone to the kind of parameters used in the procedure. Since the choice of parameters is empirical, a major part of the technique is involved in determining good parameters. Hence the convergence of the problem becomes slower and requiring large number of iterations. This paper has used Quasi Newton algorithm for optimization and it is discussed in further detail in the forthcoming sections.

The paper is organized as follows. In Section II, the basic formulation of the ELD problem is presented, followed by a brief explanation of the Quasi Newton algorithm in Section III and the proposed algorithm in section IV. Thereafter, the proposed method is compared with other conventional and modern optimization techniques in section V and finally, conclusions drawn from the analysis are summarized in Section VI.

II. PROBLEM FORMULATION

Classical optimal generation scheduling/economic load dispatch problem takes the following form [8] - Minimization of the objective function given by

$$F = \sum_{i=1}^{N} a_i P_i^2 + b_i P_i + c_i \quad (1)$$

subject to the constraints given by

(1) Power balance constraint:

$$P_d = \sum_{i=1}^{N} P_i \quad (2)$$

(2) Generation output limit constraint:

$$P_{i,min} \leq P_i \leq P_{i,max} \quad (3)$$

(3) Transmission flow constraint:

$$L_{i,min} \leq G_{k,i} P_i \leq L_{i,max} \quad (4)$$

(4) Dynamic ramp rate constraint:

$$R_{i,min} \leq P_{i,t} - P_{i,t-1} \leq R_{i,max} \quad (5)$$

This paper has considered two constraints for their work, namely power balance constraint and the generation output limits given by equations (2) and (3). But the Lagrangian function, shown in equation (6), is formed using equality constraint alone whereas the inequality constraint is applied separately as explained in Section III.

$$L(P_i, \lambda) = F + \lambda\left(P_d - \sum_{i=1}^{N} P_i\right) \quad (6)$$

III. QUASI NEWTON METHOD

Quasi Newton method is based primarily upon the properties of quadratic functions [9]. It tries to imitate the inverse hessian matrix of the Newton method merely using the first order information. This is an advantage of Quasi Newton method over Newton method.

Quasi Newton method majorly involves computation of the objective function and its gradient. It stands apart from other algorithms in the way the search direction is chosen. At every iteration of the algorithm (except the first iteration), a certain matrix A_k is to be calculated. This matrix is supposed to be the approximation of the inverse of the actual hessian matrix of the objective function at that particular iterate. Hence many a times, the initialization of the hessian matrix A_k is also done by the actual inverse of the first hessian of the objective function. But this method works fine even if the initialized matrix of A_k is an identity matrix. In fact, this paper has started with an identity matrix for its initialization. But, if the number of iterations is more, then the algorithm can be fine tuned by using the final hessian matrix obtained after running the algorithm with the identity matrix as initialized value.

But the key step is the updation of the approximate inverse Hessian matrix A_k to A_{k+1} and this updation needs to be done at every iteration. This updation needs to be done such that it contains the essential features of the inverse hessian matrix without having to actually compute it. The essential features required include symmetry and positive definiteness and it must satisfy the condition of quasi Newton (whereby the algorithm gets its name). This algorithm works well for a quadratic function and hence it can be safely assumed that with smooth convex non quadratic functions as well, this algorithm would perform well. Although, the fuel cost function is generally approximated by a quadratic equation, this algorithm would work well even if the objective function were a more accurate cubic interpolation of the fuel cost data. This is yet another advantage of this method.

As explained above, the key to the algorithm is the approximate hessian matrix updation. This can be done in two ways. The first method is namely Davidon-Fletcher-Powell (DFP) method, also known as rank one correction formula whereas the second one is Broyden-Fletcher-Glodfarb-Shanno (BFGS) method, better known as rank two correction formula. The major advantage of the BFGS method is that the method seems to be significantly less dependent upon exact line searching. But DFP method is more popular and well suited for the constant hessian matrix i.e. the quadratic case. Since the objective function considered here, is a smooth quadratic, the DFP method is implemented, whose updating formula is given by,

$$A_k = \frac{A^{(k-1)} \Delta g^{(k-1)} \Delta g^{(k-1)T} A^{k-1}}{\Delta g^{(k-1)T} A^{(k-1)} \Delta g^{(k-1)}} \quad (7)$$

This method is robust and works well in the problem described in this paper, and this will be clear from the simulation results explained in section V. The major drawback of this method is the need to store the (N X N) matrix A_k.

IV. PROPOSED ALGORITHM

As discussed in section II, the Lagrangian function formed in equation (6) consists of the equality constraint alone. But provisions have been made for the inequality constraints as well in the proposed algorithm. The steps of this algorithm are as described below.

Step 1: Initialize the generator output values for all design variables P_i where i ranges from 1 to N. Initializing vector, Φ, is inclusive of both, the entire set of output generator values and the lagrangian multiplier. Thus,

$$\Phi = [P_1, P_2, \dots \dots P_i, \dots \dots P_N, \lambda]^T.$$

Step 2: Initialize the hessian matrix A_k, where k represents iteration count, by an equivalent identity matrix wherein the number of rows and columns of the identity matrix is equal to the number of elements in the initializing vector, Φ. Since we are initializing, the value of k is obviously, 1.

Step 3: Find the gradient vector g_k of the Lagrangian function. Calculate the value of the gradient vector by substituting the initializing vector, Φ, into g_k which will give us the descent direction. In fact, for the first iteration, i.e. k = 1, we are following the pattern of steepest direction method, and one can visualize this in the forthcoming steps.

Step 4: Use the Newton method, for calculating the unit step length α, by first substituting initializing vector Φ into x_k

$$x_{k+1} = x_k - \alpha_k A_k g_k \quad (8)$$

and then find improvised function of the design variable in terms of α_k. Thereafter, replace the design variables in the lagrangian function with x_{k+1}. Differentiate this function with respect to α and equate it with zero. Solving this equation would give the value of α_k, the unit step length for the kth iteration.

Step 6: Thereafter, the same value of α is substituted into equation (8) to find the value of the new design variable vector, x_{k+1}, for the iteration $(k+1)$. Similarly new values of gradient vector g_{k+1} are found using the new design variable vector, x_{k+1}.

Step 7: Update the Hessian matrix A_k using the formula as shown in (7). In case of non quadratic smooth functions, BFGS updation technique is better equipped to deal with more accurate updation of the hessian matrix as opposed to DFP.

Step 8: Check the fuel cost function value with the new updated generator output value and compare with previous values. If the decrease in cost is not within tolerance value then go back to step 3.

Step 9: Check the final generator output values with the maximum and minimum generator output limit constraints. If these constraints are violated by any of the design variables, then equate them to the limiting value and repeat the above mentioned problem from step 1 but this time, excluding the design variable which violated the rule.

V. Simulation Results

In this section, the computational results of the proposed algorithm on IEEE 30 bus system with 6 generators have been presented. The minimum and maximum output limits of the generating units and their individual cost coefficients are listed in Table 1. The proposed algorithm was implemented in MATLAB environment for a load demand of 270 MW. Results generated by this MATLAB code were compared with the results of the in-built optimization algorithms in MATLAB's 'optimtool' toolbox.

TABLE I. Generator data for 6 unit test system

Gen. no.	$P_{i,\,min}$	$P_{i,\,max}$	a_i (\$/MWh^2)	b_i (\$/MWh)	c_i (\$)
1.	50	200	0.00375	2.00	0
2.	20	80	0.01750	1.75	0
3.	15	50	0.06250	1.00	0
4.	10	35	0.00834	3.25	0
5.	10	30	0.02500	3.00	0
6.	12	40	0.02500	3.00	0

TABLE II. Comparison of termination count of each algorithm

Optimization Technique	P_1	P_2	P_3	P_4	P_5	P_6	Iterations
Genetic algorithm	150.4	45.7	27.7	15.1	11.5	19.6	51
Pattern search	182.3	37.7	18.0	10.0	10.0	12.0	112
Interior point method	174.9	44.6	18.5	10.0	10.0	12.0	21
Sequential Quadratic Programming	174.9	44.6	18.5	10.0	10.0	12.0	18
Quasi Newton method	174.9	44.6	18.5	10.0	10.0	12.0	14

Table II summarizes the comparative results of the proposed algorithm with other conventional and AI based methods. It can be seen that the number of iterations required is high in case of genetic algorithm and pattern search method. Sequential quadratic programming (SQP) and interior point method takes much less number of iterations but the proposed algorithm gives the desired results in the smallest number of iterations.

Table III compares the final optimized cost achieved by all the four different algorithms. One can observe that the conventional techniques give the most accurate result whereas genetic algorithm gives a result which is close to the accurate result but not the true value itself.

These observations may be referred in the following context. AI based techniques have its advantages over conventional gradient based techniques such as its ability to deal with discontinuous and non convex functions whereas conventional methods fail in this area. But conventional methods are faster and give much more accurate results as compared to AI based techniques as seen in Table II and III. Thus a hybrid approach of conventional and AI based technique can solve a much broader range of problems. For this, among conventional techniques, Quasi Newton algorithm can be hailed as the most robust technique. Results make it clear that it is the most suitable method for the optimal generation scheduling problem

TABLE III. COMPARISON OF OPTIMIZED COST ACHIEVED THROUGH EACH ALGORITHM

Optimization Technique	***Minimum cost achieved***
Genetic algorithm	735
Pattern search	723
Sequential quadratic programming	722
Interior point method	722
Quasi Newton method	722

VI. CONCLUSION

Optimal generation scheduling problem relies heavily on both speed of the optimization algorithm as well as the proximity of the solution to the true optimized point in order to carry out an economic operation of the power system. The comparative results in this paper indicate that although artificial intelligence (AI) based technique give global optimized solutions, they lag in speed and accuracy. On the other hand, conventional techniques give an accurate solution but it can tend to be slow at convergence in our problem. Quasi Newton stands out for its robustness, accuracy and most of all for its low level of iterations required for the outcome/solution. This also further supports the case that in future, work can be done in the direction of hybrid techniques involving Quasi Newton method and AI based technique. This would bring out the best of both worlds for the economic dispatch problem.

REFERENCES

[1] B. H. Chowdhary and Saifur Rahman, "A Review of recent advances in economic dispatch," IEEE Transactions on Power systems, vol.5, no.4, November 1990

[2] C. E. Lin, S.T. Chen and C. L. Huang "A direct Newton Raphson economic dispatch," IEEE Transactions on Power systems, vol. 7, August 2002.

[3] G. P. Grannelli and M. Montgna, "Security constrained economic dispatch using dual quadratic programming," Electric Power Systems Research, vol. 56, October 2000.

[4] A. Farag, S. Bayat and T. C. Cheng, "Economic load dispatch multiobjective optimization procedures using linear programming techniques," IEEE Transactions on Power systems, vol. 10, May 1995.

[5] C. Wang and S. M. Shahidehpour, "Optimal generation scheduling with ramping costs," IEEE Transactions on Power systems, vol. 10, Feb 1995.

[6] C. Chao-Lung, "Improved genetic algorithm for power economic dispatch of units with valve point effects and multiple fuels," IEEE Transactions on Power systems, vol. 20, Nov, 2005.

[7] D. N. Simopoulos, S. D. Kavatza and C. D. Vournas, "Unit commitment by an enhanced simulated annealing algorithm", IEEE Transactions on Power systems, vol. 21, February 2006.

[8] A. J. Wood and B. F. Wollenberg, "Power - Generation, Operation and Control", John Wiley & sons, 2003.

[9] A. Ravindran, K. m. Ragsdell and G. V. Reklaitis, "Engineering Optimization", John Wiley & sons, 2006.