A FAST AND ROBUST VARIABLE METRIC METHOD FOR OPTIMUM POWER FLOW

by

Sarosh N. Talukdar and Theo C. Giras

DRC-18-37-81

September 1981

A FAST AND ROBUST VARIABLE METRIC METHOD FOR OPTIMUM POWER FLOWS

Sarosh N. Talukdar, Member, IEEE
Power Engineering Program
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Theo C. Giras, Member, IEEE
Research & Development
Westinghouse Corporation
Pittsburgh, Pennsylvania 15235

SUMMARY

The Han-Powell algorithm for nonlinear programming has several attractive features. It is very fast and on test problems has outperformed its competition by considerable margins. It is also very robust. It is neither necessary to begin with a feasible point nor to tightly converge the constraints at each iteration. Its only major disadvantage stems from its use of certain nonsparse approximations to Hessian matrices. These matrices are of dimension (m x m) where m is the number of decision variables in the problem. If the Han-Powell algorithm is directly applied to the Optimum Power Flow (OPF) problem, all the network variables are treated as decision variables. For networks with about 1000 nodes, the resulting Hessian-matrix-approximations become too big to conveniently handle.

One remedy is to use an elimination procedure to reduce the number of decision variables. Berna, Locke and Westerberg have suggested one such procedure. We have tested it and found it to be promising, though it does not completely circumvent the difficulty of dealing with large Hessians. This paper develops another and seemingly more attractive elimination procedure. The basic idea is an old one - to use the equality constraints to eliminate some of the variables. However, this idea is implemented in a new way with concepts borrowed from the fields of network dissection and parallel processing.

The computations are arranged in two nested loops. At the start of each circuit through the outer loop, the inner loop is invoked to eliminate some of the variables. This is done by satisfying the OPF problem's equality constraints to a tolerance, θ, specified by the outer loop. The result is a problem with fewer decision variables and fewer constraints. The outer loop, applies the original Han-Powell algorithm to this smaller problem. As it converges to a solution, it tightens the tolerance, θ, given to the inner loop.

We have organized the computations so that the inner loop is a conventional Newton-type-load-flow. Initially, only one iteration is required of this procedure. Later, as the outer loop converges, two or three may be required per iteration of the outer loop.

The method has been tested on small problems (50 buses or less) and has been found to perform admirably. Larger problems remain to be tested but the prognosis is good.

A FAST AND ROBUST VARIABLE METRIC METHOD FOR OPTIMUM POWER FLOWS

Sarosh N. Talukdar, Member, IEEE
Power EngineerIng Program
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

Theo C. Giras, Member, IEEE
Research & Development
Westinghouse Corporation
Pittsburgh, Pennsylvania 15235

ABSTRACT

The Han-Powell algorithm has proved to be extremely fast and robust for small Optimum Power Problems [1]. There is every reason to believe its performance could be extended to large problems, provided its one serious disadvantage is eliminated. This disadvantage stems from its use of aonsparse approximations to certain Hessian matrices. These matrices are of dimension (a x a) where 3 i3 the number of decision variables. Since all the network variables are retained by the algorithm as decision variables, the Hessians quickly get too big to be conveniently accommodated. One remedy is to add a variable-reduction-procedure. Serna, Locke and Westerberg [2] have developed one such procedure. It helps but does not completely eliminate the difficulties. This paper develops another reduction procedure with concepts borrowed from the fields of network dissection and parallel processing. The computations are arranged in two nested loops. The Inner loop eliminates n < m of the variables by satisfying the problem's n equality constraints to a tolerance that is tightened as the problem's solution is approached. The outer loop applies the Han-Powell algorithm to the reduced problem. Besides eliminating the need for dealing with unvieldly Hessians, this "reduced method"* appears to be as robust as the original Han-Powell algorithm and converges at least as fast for small problems. The method has not yet been tested on large problems, but it is reasonable to expect that it will perform as well on them. *

INTRODUCTION

The physical and electrical aspects of optimum power flow problems have been more than adequately covered in the literature, e.g. [3]-{9J. For the purposes of brevity, we will not reiterate them here. Rather, we will begin directly with a mathematical description of the problem, namely:

(OPF): $\min_{Z} f(Z)$

subject to $g(2)-0$

$h(Z) \geq 0$

where

- **f** is an objective function that usually reflects fuel costs but can be selected to represent other concerns Ilka delivery losses and deviations from some preselected schedule
- **Z** is an a-vector of network variables like bus voltages and generator powers
- **g** is a function vector of dimension a. An element of g represents the total power entering a bus. Thus, the equality constraints in (OPF) are equivalent to the equations of a traditional load flow.
- **h** is a function vector of dimension p. It embodies equipment ratings, security limits, acceptable-quality-of-service ranges and contingency constraints.

Because transients are aeglecjted in OPF formulations* the g and h vectors contain ao derivative or integral operations but only algebraic functions. Nevertheless, the OPF problem is difficult to solve. The reason is that the g and h vectors are large. Each can contain thousands of entries. As operating conditions become more *swrex*$^{\wedge}_{9}$ and they inevitably will [10], the g and h vectors will become even larger. Thus, we can expect the OPF problem to grow even more computationally formidable than it now is.

Existing methods for solving OPF problems, e.g. [3]-[9]$_9$ have several deficiencies stemming from their use of penalty functions to handle constraints (which slow convergence), variable swapping schemes (which increase logical complexity) and feasible point algorithms (which require the starting point to meet all the constraints). In addition, existing methods are often la^iri^g in robustness. Under stressful circumstances they often bog down or fail to converge {12].

In reference [1] Giras and Talukdar suggested the use of the Han-Powell method [13]-[15] with a variable-reduction scheme developed by Berna, Locke and Westerberg [2]. The resulting algorithm is fast and very robust but still uses certain large and aonsparse Hessian matrices. la this paper ue will describe a reduction procedure that appears to be more convenient while preserving the speed and robustness of the original algorithm.

THE HAN-'OWELL ALGORITHM

In each iteration of this algorithm, Z_{old}, an estimate to the solution of (OPF), is improved by taking a step of length 2 in a direction-of-govement, 3. la other words, Z_{new}, the improved estimate, is obtained from

$$Z_{new} = Z_{old} + \lambda S$$

The direction-of-aovement is determined by solving a Quadratic Programming Problen whose objective function is a. second order approximation of the original objective function and whose constraints are first order (linear) approximations of the original constraints. The Quadratic Programming Problem has the fora:

(QPP): $\min_{S} \{f + \frac{\partial f}{\partial Z^T} S + \frac{1}{2} S^T H S\}$

subject to $g + \frac{\partial g}{\partial Z^T} S_m Q$

$h + \frac{\partial h}{\partial Z^T} S \geq C$

where T denotes transpose

f,g,h are evaluated at $Z = Z_{old}$ as are their derivatives

and H is a positive definite approximation to $\partial^2 f / \partial Z \partial Z^T$, the Hessian of f which in turn, is the Lagrangian of (OPF).

For the first two iterations E is usually set to unity. Thereafter, it is updated by plugging the latest available information on the gradients of f into formulae given in [13]. This updating process is the identifying characteristic of a Variable Metric (alias: Quasi-Newton) Method.

Once (QPP) has been formulated it can be salved by a standard Quadratic Programming code like [16], to give S. Sext, the step# size a is chosen so that

$$\psi(Z_{new}) = \psi(Z_{old} + \alpha S) \leq \text{old}$$

where ? is a penalty function of the form:

$$\wedge(Z) = f(Z) + X^T g(Z) - Hi^T [\text{Min}(o, h(Z))] \quad (2)$$

and \,u are non-negative vectors. Han [15] has shown that if X and u are sufficiently large-the overall method will converge even when the starting point is infeasible.

Further details on the algorithm can* be found in [13]-[14]. Ve will conclude the discussion here with a brief critique of its features.

Its aain strengths are speed and robustness. On test problems it has outperformed its competition by considerable margins [21, [13]. Moreover, it is neither necessary to begin with a feasible point nor to tightly converge the constraints at each iteration. Even with starting points well outside the feasible region the method rapidly converges to an optimum solution.

Its only aajor disadvantage stems from the use of the Hessian approximation, H. This matrix is nonsparse and of dimension (m x a). In OPF problems "a is often 1000 or greater. The resulting H aatrices are too large to conveniently tackle with available Quadratic Programming codes.

3ema, Locke and %*esterberg [2] have suggestad an elimination procedure that reduces the size of the matrix to be handled in (QPP)* but requires a good deal of precomputation. In the next section ve will present what appears to be a more attractive reduction procedure.

A BESTED REDACTION PR0CZDUR2

The basic idea that will be developed here is an old one - to use the n equality constraints in (OPF) to eliminate n of its variables. • This idea is central to the 3. educed Gradient and *G2C* net hods, though their primary intent in using it is to satisfy constraints; ours is to sake the optimization problem smaller. Also, we will implement the idea in a new vay with techniques borrowed from the areas of network dissection and parallel processing.

First, we partition the a-vector of network variables, Z, into two subvectors X and U so that X is of dimension n and U is of dimension m-a. In the same vay, we partition the direction-of-movement-vector, S, into S_x and S_r. Ve will refer to tJ as the reduced-decision-vector and to S_u as the reduced direction-of-movement.

Mext* we eliminate the variables in X and find S_u.

Finding the Reduced-Direction-of—Movement

Rewriting (OPF) in terms of U and X gives:

(OPT[1]): $\min_{U,X} f(U,X)$

subject to $g(U,X) = 0$

$h(U,X) \geq 0$

Let S-y(U) (3)

be a solution to tha equality constraints in CQPF'l, i.e.

$$g(U, y(U)) = 0 \quad (4)$$

3y replacing X with y(U) in (0?F[r]) we get a smaller . problem in n fewer variables, namely:

(ROPF): $\min_{U} f(U, y(U))$

subject to $h(U, y(U)) \geq 0$

Applying the Han-?owell method to this smaller problem we get the Quadratic Programming Problem whose solution, S^, is the reduced-direction-of-movement. This Quadratic Programing Problem has the form:

(RQPP): $\min_{S_u} \{f + r(\nabla_u f)^T S_u + \frac{1}{2} S_u^T G S_u\}$

subject to $b + (\nabla_u h)^T S_u \geq 0$

wnere $\nabla_u f = \frac{\partial f}{\partial U} - \frac{\partial X^T}{\partial U} \frac{\partial f}{\partial X}$ (5)

$\nabla_u h = \frac{\partial h^T}{\partial U} - \frac{\partial X^T}{\partial U} \frac{\partial h^T}{\partial X}$ (6)

and G is a positive definite approximation to the Hessian $\partial^2\bar{f}/\partial U \partial U^X$; $\bar{f}$ is the Lagrangian of (ROPF).

Estisating the Coefficients of (RQPP)

To assemble (HQPP) we need f, h, T_t^f, 7^h and G, all evaluated at U_{old}, the incumbent estimate for the decision-variable-vector. As already mentioned, G is calculated from the gradients, $7_a f$ and $7_u h$. These in turn can be obtained from (5) and (6). Explicit expressions for all the terms in the right-hand-sides of these equations are readily available except for $\partial Y^L/\partial U$. In addition, we need to know *y* in order to evaluate f and h.

In summary then, to assemble (RQPP), we need to datersine the values of $y(U_{old})$ and $\partial Y^X(U_{old})/\partial U$. How is ihis to be done? A similar problem arises %hn certain tearing and dissection methods are used to divide large networks into smaller pieces that can be solved in parallel (see, for example, [17], [IS]). Experience with these methods indicates that Y and its derivatives do not have to be known exactly. Instead, they can be approximated by the results from a Newton-type-iteration as follows:

Guess X_0

Set

$$X_{k+1} = X_k - \left[\frac{\partial g(U_{old}, X_k)}{\partial X^T}\right]^{-1} g(U_{old}, X_k); k=0,1,\ldots r-1 \quad (7)$$

vhera r is such that $|| X_{r-L} - X_r || \leq 0$

and 3 is a tolerance to be discussed in the "Remarks" at the end of this section.

Set $$Y(U_{old}) = X_r \quad (8)$$

Set $$-\frac{\partial Y^T(U_{old})}{\partial U} = -\frac{\partial g^T(U_{old}, X_r)}{\partial U}\left[\frac{\partial g^T(U_{old}, X_r)}{\partial X}\right]^{-1} \quad (9)$$

(This expression is most easily derived by differentiating (4) to give:

$$0 = \frac{\partial g^T}{\partial C} + \frac{\partial Y^T}{\partial D}\frac{\partial g^T}{\partial y}$$

and noting that *y* and X are interchangeable)

Step Size Selection

Once (RQPP) has been assembled it can be solved for S_{nt} the reduced-direction-of-movement, by using any Quadratic Programming code, e.g. [16]. The next problem is to find the step size, a.

There are several ways to adapt the Han-Powell-step-size-selection-procedure (c.f. equations (1) and (2)) to our situation. Perhaps the simplest is co continue with the reduced formulation (ROPF) and choose 3 so that:

$$\Psi(U_{new}) = \min_{\alpha \in [0,1]} \{\Psi(U_{old} + \alpha S_u)\} \quad C1C)$$

where $$\Psi(U) = f(U, Y(U)) + u^T [\min\{0, h(U, Y(U))\}] \quad (U)$$

U is the vector of Xuinn Tucker aultipiiers determined in solving (RQPP)

y'(U) is a linear approximation to $^V(U)$, nameiy:

$$Y'(U) = Y(U_{old}) + \frac{\partial Y}{\partial U^T}[U - U_{old}] \quad (12)$$

We have found this approach to work reasonably veil.

The Overall Algorithm and the Tolerance,3

The overall algorithm has the form shown in Fig. 1. To implement it ve need a ^ay to select 3.

Habbat, Sangiovanni-Vincentaili and Hsieh [131 have considered a related problem in necvork dis-section. In their problem the outer loop was capable of quadratic convergence when y(U) was known exactly. They showed that the outer loop would continue to converge quadratically when y(U) was approximated as in (7) and (3) provided that

$$e \leq ||\Delta U||^2 \quad (i3)$$

where AU is the change in the value of C prescribed by the outer loop in the previous iteration.

The Han-Powell algorithm (our outer loop) can achieve quadratic convergence (when the number of active inequality constraints is a-a) but usually converges at a lower (super linear) rate. ' This would suggest that we use a formula of the fora:

$$||\Delta U||^\beta$$

with $1 \leq 3 \leq 2$. This issue is still under investigation. So far our experience has been that one iteration of the inner loop suffices till the overall solution is approached when cwo or three inner loop iterations may be necessary.

THE REDPCED-OECISIOS-VARIABLS-VSCTOR

In this section we consider the problem of partitioning the network variables, Z, into the reduced-decision-variable-vector, U, and the other vector, X. The partition must be such that we can find a *y($)* with which to replace X. That is, it must be possible to solve the equality constraints in (OPF) for X in terms of tl. We will describe a partition that meets this criterion and is attractive for other reasons as well.

The elements of *Z* are given by:

$$Z^T = \{\ldots, V_i, \ldots, P_i, Q_i, \ldots \mid k \neq C \text{ and } i \neq C\}$$

where $V_i \angle\theta_i$ is the complex voltage of the i-ch bus

$P_L + jQ_L$ is the complex power injected into the i-th bus by a generator, load, ac-tie-line or dc-line

W Is a set of the settings for the tap changing and phase shifting transformers. The elements of W are assumed to be continuous.

A,3,C are the sets of slack, generator and load buses.

We select the reduced decision vèctor, U, so It contains the variables over which the dispatcher has direct and independent control. Specifically:

$$U^T=\{V_k, V_i, P_i, W \mid k\epsilon A \text{ aad } i\epsilon S\}$$

$$X^T=\{P_k, Q_k, \delta_i, Q_i, V_h, \delta_n \mid k\epsilon A,\ i\epsilon B,\ h\epsilon C\}$$

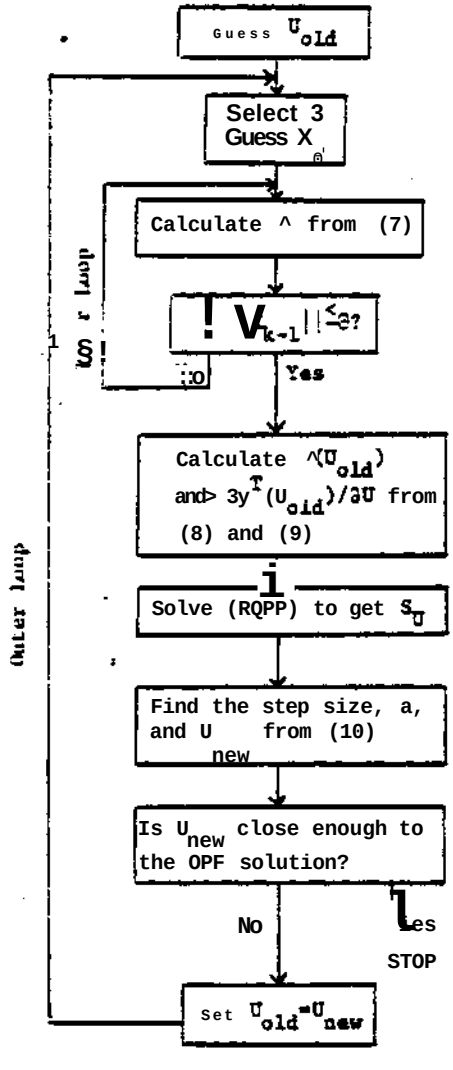

Fig. I A Flow Chart of che Overall Algorithm.

This choice stakes X the set of variables whose values are calculated by a conventional load flow. Thus, as long as a load flow is possible we can express X in terms of a — a situation that experience indicates occurs for a wide range of values of U, probably all the U's of interest. The partition has other advantages as well, among them [1]:

1. The partition is constant and does aot have to be changed repeatedly as happens with many reduced gradient procedures.

2. It Is aesthetically satisfying to use the variables over which the dispatcher has direct and independent control as the decision variables in optimization. Stott [12] has noted that these variables may also be better for dealing with contingency constraints, suppression of ineffective rescheduling and finding an operationally laplementable sequence of control. adjustments from che existing point to che computed optimal point.

3. The code from a Newton-type-load-flow can be used to generate the approximation, Y(U), and its derivatives.

AN SXAMPI2

We have run a number of small examples (less than 50 buses) and compared che Han-Powell algorithm with our "reduced algorithm." The results indicate that the "reduced algorithm" is at least as robust (in its ability to converge to an optimum solution from infeasible starting points) and converges in the same number or fewer overall iterations. As an illustration consider the AE? 30 Bus Test System shown in Fig. 2. Parameter values and operating data for this system can be found in [6]. We used this data together, with the generator cost functions and starting values given in Table I and II. The resulting OPF problem had 60 equality constraints, 30 inequality constraints and 69 network variables. The Han-Powell algorithm converged in five iterations, our "reduced algorithm" in four iterations of the outer loop with an average of cwo load-flow* iterations per outer-loop-iteration. We have not made operation counts but the advantages of the • "reduced algorithm'* are fairly apparent. While the Han-Powell algorithm must contend with 60 variables and Hessians of dimension (69 x 69) the "reduced algorithm" has nine decision variables and a Hessian of dimension (9 x 9). This reduction in size is achieved at the expense of about one additional load-flow-iteration per outer-loop-iteration; a computational bar gin by any standards.

CONCLUSION

The Han-Powell algorithm has several features that make it very attractive for OPF problems. Specifically, it is logically straightforward and can be applied without resorting to intricate maneuvers; it is very robust and will converge to an optimum solution even from infeasible starting points; and it converges rapidly. Its one major disadvantage in che context of OPF problems stems from its use of all che network variables as decision variables. As a result it oust contend with large Hessian matrices.

la this paper we have described a procedure for reducing the size of the decision vector by a variables, where a is the number of equality constraints in the OPF problem. Thereby, the Hessian satrices are made correspondingly smaller.

The reduction is achieved by using one to three iterations of a Newton-type—load-flow within each iteration of the overall optimization procedure.

The reduction makes is possible to accommodate large networks containing 10Q0 buses or more. However, cests en such networks remain to be performed. A concern is the number of iterations chat will be required. In general, optimization methods of the type used here provide only Quadratic Termination - a property that implies an increase ia che number of iterations with increase in network size. However, there is a mitigating factor. The Han-Powell algorithm and our modification of it, approach quadratic convergence rates as Che number of active inequality constraints approaches the number of decision variables. Quadratic Convergence *la* such quicker than Quadratic Termination. With it, the number of iterations is usually independent of problem size. OP? problems tend to be'constraint-bound. Therefore, it is reasonable to expect that the method described here will require about as many iterations for large networks as it does for small ones•

TABLE I Generator Operating Costs

$C_i = V_i + T_i P_i^2$

where C_i, P_i are Che cost and power output (ISO of the i-ch generator.

i (Bus #)	a_i	a_i
1	4.25	0.025
2	3.00	0.025
3	2.00	0.00375
4	1.00	0.0625
30 (slack)	10.00	0.00834

TABLE II Starting Point

All angles are zero.
All voltages, except slack bus, set to unlty.
Slack sec to 1.06.

i	*i*	2	3	4
P_{i} (MW)	50	50	50	40

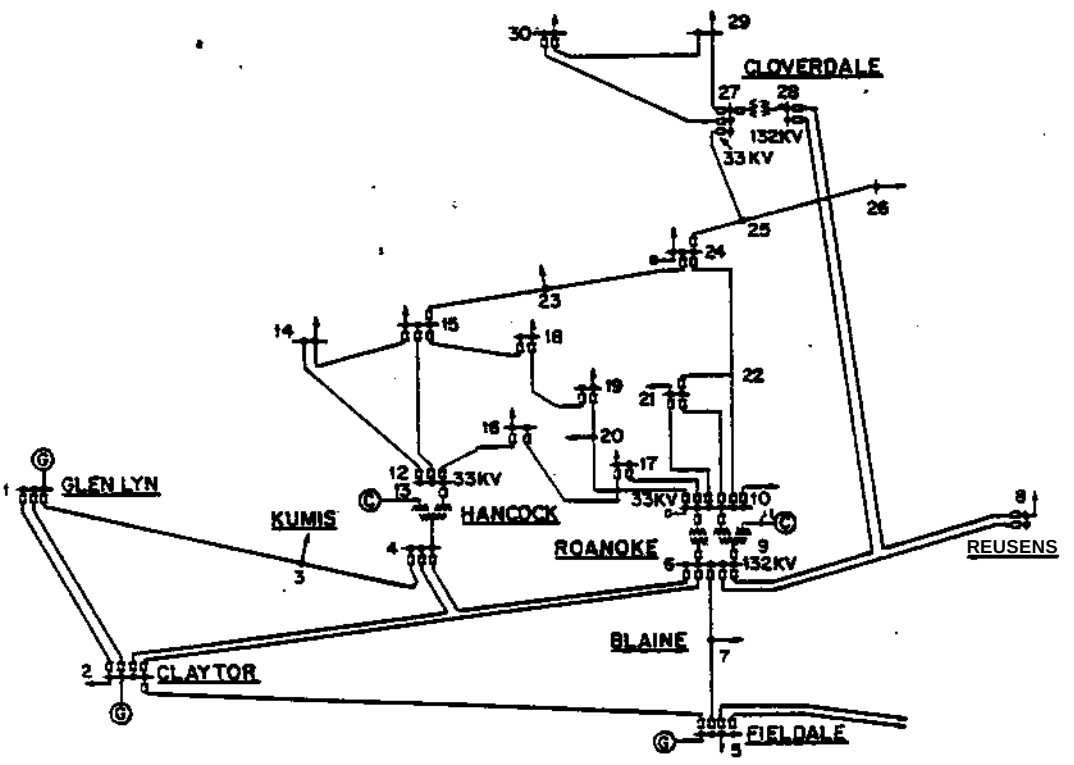

Fig. 2 AEP 30 Bus Test System

REFERENCES

[1] T. Giras and S.N. Talukdar, "A Quasi-Newton Method for Optimal Power Flows," submitted to the International Journal of Electrical Power & Energy Systems, July 1980.

[2] T.J. Berna, M.H. Locke and A.W. Westerberg, "A New Approach to Optimization of Chemical Processes," AICHE Journal, January 1980.

[3] B. Stott, O. Alsac and J.A. Marinho, "The Optimal Power Flow Problem," presented at the SIAM International Conference on Electric Power Problems: The Mathematical Challenge Seattle, March 1980.

[4] H.H. Happ, "Optimal Power Dispatch - A Comprehensive Survey," IEEE Transactions on PAS, Vol. PAS-96, No. 3, May/June 1977.

[5] B. Stott, J.L. Marinho and O. Alsac, "Review of Linear Programming Applied to Power System Rescheduling," PICA-79, Cleveland, 1979.

[6] B. Stott and B. Hobson, "Power System Security Control Calculations using Linear Programming, Parts I and II," IEEE Transactions on PAS, Vol. PAS-97, No. 5, September/October 1978.

[7] F. Wu, G. Gross, J.F. Luini and P.M. Look, "A Two-Stage Approach to Solving Large-Scale Optimal Power Flows," PICA-79, Cleveland, 1979.

[8] H.W. Dommel and W.F. Tinney, "Optimal Power Flow Solutions," IEEE Transactions on PAS, Vol. PAS-87, pp. 1866-1976, 1968.

[9] W.R. Barcelo, W.W. Lemon and H.R. Koen, "Optimization of the Real-Time Dispatch with Constraints for Secure Operation of Bulk Power Systems," IEEE Transactions on PAS, Vol. PAS-96, pp. 741-757, May/June 1977.

[10] M.G. Morgan and S.N. Talukdar, "Electric Load Management: Some Technical, Economic, Regulatory and Social Issues," invited paper, IEEE Proceedings, pp. 241-312, February 1979.

[11] M. Avriel, Nonlinear Programming Analysis and Methods, Prentice-Hall, 1976.

[12] Unpublished communications from B. Stott.

[13] M.J.D. Powell, "A Fast Algorithm for Nonlinearly Constrained Optimization Calculations," presented at the 1977 Dundee Conference on Numerical Analysis.

[14] M.J.D. Powell, "The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations," presented at the Nonlinear Programming 3 Symposium, Madison, Wisconsin, 1977.

[15] S.P. Han, "A Globally Convergent Method for Nonlinear Programming," Report No. 75-257, Department of Computer Science, Cornell University, 1975.

[16] R. Fletcher, "A General Quadratic Programming Algorithm," HL70/1249, Atomic Energy Research Establishment, Harwell, U.K., March 1970.

[17] S.N. Talukdar and D. Thomas, "Modular Algorithms and Multiprocessors for Simulating Power Systems," EPRI Special Report EL-566-SR on the Workshop: "Exploring Applications of Parallel Processing to Power System Analysis Problems," Palo Alto, California, October 1977.

[18] N.B.G. Rabbat, A.L. Sangiovanni-Vincentelli and H.Y. Hsieh, "A Multilevel Newton Algorithm with Macromodeling and Latency for the Analysis of Large-Scale Nonlinear Circuits in the Time Domain," IEEE Transactions on CAS, Vol. CAS-26, No. 9, September 1979.