

MATPOWER: A Matlab Power System Simulation Package

R. Zimmerman
rz10@cornell.edu
Power Systems Engineering Research Center (PSERC)
School of Electrical Engineering, Cornell University

D. Gan
deqiang@ee.cornell.edu
Power Systems Engineering Research Center (PSERC)
School of Electrical Engineering, Cornell University

INTRODUCTION	1
THE POWER FLOW SOLVER	2
THE OPTIMAL POWER FLOW FORMULATION	2
THE OPTIMIZATION TOOLBOX BASED OPF SOLVER	2
THE LP-BASED OPF SOLVER	2
General Introduction	3
The Sparse Solver	4
The Dense Solver	7
The Formulation of LP Sub-problem.....	7
Solving the LP Sub-problem.....	8
Computing KT Multipliers (incomplete).....	8
Checking KT Conditions (incomplete)	8
Computing Generation-load Shift Factors (incomplete)	9
The Jacobian Matrix of Load Flow	9
Linearized Formulation of Line Flow	10
TEST RESULTS (INCOMPLETE)	10
THE COMBINED UC-OPF SOLVER (INCOMPLETE)	10
FINAL REMARKS (INCOMPLETE)	10
REFERENCE	10

Introduction

The restructuring of the electric power industry in many parts of the world is creating a need for new analytical as well as simulation tools. To test new ideas and methodologies for the operation of competitive power systems, researchers need to have ready access to simulation tools which are easy to use and modify. The MATPOWER package, a set of Matlab m-files developed by PSERC at Cornell University, is intended as such a tool.

At present, MATPOWER can be used to solve the following problems:

- power flow (based on Newton's method)
- optimal power flow (using the 'constr' function in Matlab's Optimization Toolbox)
- optimal power flow (using an LP-based approach)
- optimal power flow with a heuristic for turning off expensive generators

Future versions of MATPOWER may include the ability to do:

- economic dispatch
- unit commitment
- combined unit commitment and optimal power flow
- transient stability
- stability-constrained optimal power flow

This note is simply a technical note maintained by the authors describing the details of the algorithms implemented in MATPOWER. Its purpose is to provide technical details for those interested in modifying or extending the current functionality.

The Power Flow Solver

The power flow solver in MATPOWER is based on a standard full-Jacobian Newton's method, described in detail in many textbooks. It does not include any transformer tap changing or feasibility checking. The method for building the Jacobian came from Chris DeMarco.

We found that it solves the IEEE 300-bus system on our Sun Ultra in under 2 seconds.

The Optimal Power Flow Formulation

The OPF problem is formulated as follows:

$$\text{Minimize}_{P_g, Q_g} f_i$$

$$S.T. \quad P_{gi} - P_{Li} - P(V_i) = 0 \quad (\text{Active power equations})$$

$$Q_{gi} - Q_{Li} - Q(V_i) = 0 \quad (\text{Reactive power equations})$$

$$\tilde{S}_{ij}^f \leq S_{ij}^M \quad (\text{Constraints on apparent power of line flow, from side})$$

$$\tilde{S}_{ij}^t \leq S_{ij}^M \quad (\text{Constraints on apparent power of line flow, to side})$$

$$V_i \leq V^M \quad (\text{Voltage constraints})$$

$$-V_i \leq -V^m \quad (\text{Voltage constraints})$$

$$P_{gi} \leq P_{gi}^M \quad (\text{Active power generation limits})$$

$$-P_{gi} \leq -P_{gi}^m \quad (\text{Active power generation limits})$$

$$Q_{gi} \leq Q_{gi}^M \quad (\text{Reactive power generation limits})$$

$$-Q_{gi} \leq -Q_{gi}^m \quad (\text{Reactive power generation limits})$$

The OPF formulation described above is used for both the Optimization Toolbox based OPF and the LP-based OPF. The objective function of the OPF problem is assumed to be summation of costs of individual generators. The cost of each generator is expressed in the form of either quadratic function or piece-wise linear curve.

The Optimization Toolbox based OPF Solver

Implementation of the Optimization Toolbox based OPF is quite straightforward. For details, see the file OTopf.m.

The LP-based OPF Solver

In this chapter, we will describe the algorithms implemented in our LP-based OPF solver.

The objective function of the OPF problem is assumed to be summation of costs of individual generators. The cost of each generator is expressed in the form of either quadratic function or piece-wise linear curve. The quadratic cost function of i th generator is defined as follows:

$$f_i = C_{0i} + C_{1i} P_{gi} + C_{2i} P_{gi}^2$$

The piece-wise linear cost function of i th generator is defined as follows (see Fig. 2):

$$f_i = c_i + \sum_{k=1}^{n} P_{gil}^{(k)} + \dots + P_{ginseg(i)}^{(n)}$$

$$P_{gi} = P_{gil} + \dots + P_{ginseg(i)}$$

$$0 \leq P_{gij} \leq P_{gij}^M$$

Where P_{gil} is a constant, P_{gij} denotes the slope of each linear function, P_{gij} is called segmental power in this note. Note that the cost curve must be convex, otherwise the solver could get wrong solution. This approach of modeling generator cost function is called *one-segment-one-variable approach* [Stott, 1979].

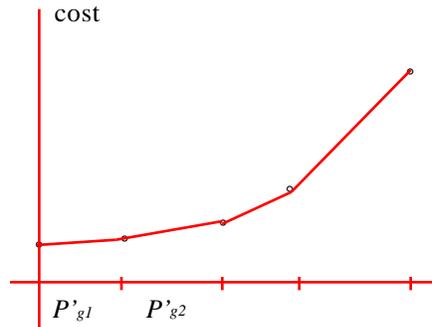


Fig. 2 Cost Curve of Generator

General Introduction

LP-based OPF method has been extensively examined, and used in quite a few modern power systems. The algorithm we coded is perhaps different from that in production-grade software. The major difference is that our algorithm is much simpler. The reason of doing so is, certainly, to make the OPF solver more re-usable. The flow chart of our solver is demonstrated in Fig. 3.

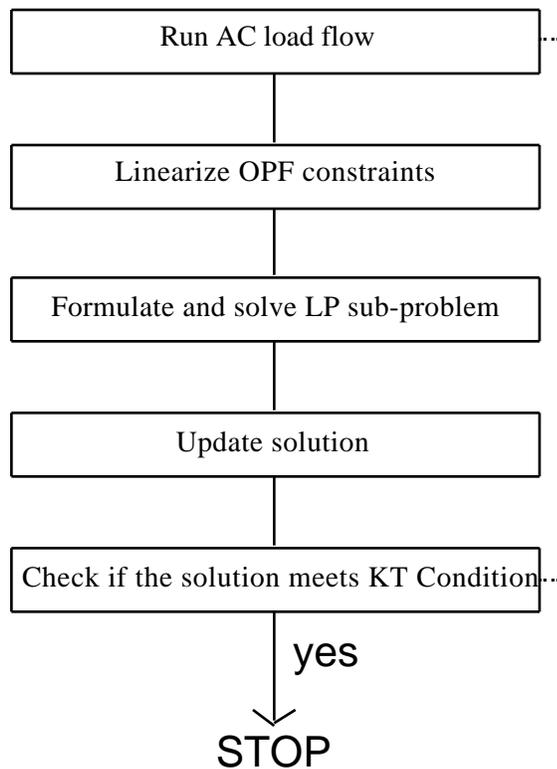


Fig. 3 Flow Chart of LP-based OPF Solver

The stopping criteria, the selection of starting point, handling infeasibilities, etc. will be explained in the coming version of this note.

The LP-based OPF solver can not handle OPF problems with quadratic cost function directly. If the cost information of generators is given by quadratic functions, the quadratic curves will be discretized into piece-wise linear functions in M-file opf.m. So in this chapter, we assume that the cost of generators is piece-wise linear.

We have developed two variants of LP-based OPF solver, one is called dense solver, and another is sparse solver. In the algorithm of dense solver, load flow equations are eliminated, so are network voltages. In sparse solver, load flow equations are preserved. The dense solver is more independent on the performance of LP solver but is less readable, while sparse solver is more readable but is more dependent on the performance of LP solver. In subsequent text, we will give the details underlying both solvers.

The Sparse Solver

The algorithm of sparse solver is pretty straightforward. In this section, we will just give the equations of linearized OPF only:

$$\text{Minimize } V, P_G, Q_G \quad \left(\sum_{i=1}^{n_1} P_{g1i} + \dots + \sum_{i=1}^{n_{l-1}} P_{g1, nseg(i)} \right) + \dots + \left(\sum_{i=1}^{n_l} P_{gn1} + \dots + \sum_{i=1}^{n_{ng}} P_{gn, nseg(i)} \right)$$

$$S.T. \quad P_{gi} - \sum_{j=1}^{nseg(i)} P_{Gij} = 0 \quad (i=1, \dots, ng)$$

$$\sum_{j=1}^n \frac{P_i}{V_j} V_j + \sum_{j=2}^n \frac{P_i}{V_j} V_j - P_{gi} = 0 \quad (i=1, \dots, nb)$$

$$\sum_{j=1}^n \frac{Q_i}{V_j} V_j + \sum_{j=2}^n \frac{Q_i}{V_j} V_j - Q_{Gi} = 0 \quad (i=1, \dots, nb)$$

$$\sum_{j=1}^n \frac{\tilde{S}_l^f}{V_j} V_j + \sum_{j=2}^n \frac{\tilde{S}_l^f}{V_j} V_j \quad S_l^M - S_l^0 \quad (l=1, \dots, nl)$$

$$\sum_{j=1}^n \frac{\tilde{S}_l^t}{V_j} V_j + \sum_{j=2}^n \frac{\tilde{S}_l^t}{V_j} V_j \quad S_l^M - S_l^0 \quad (l=1, \dots, nl)$$

$$V_i \quad V^M - V_i^0$$

$$- V_i \quad V_i^0 - V^m$$

$$P_{Gi} \quad P_{Gi}^M - P_{Gi}^0 \quad (i=1, \dots, ng)$$

$$- P_{Gi} \quad P_{Gi}^0 - P_{Gi}^m \quad (i=1, \dots, ng)$$

$$Q_{Gi} \quad Q_{Gi}^M - Q_{Gi}^0 \quad (i=1, \dots, ng)$$

$$- Q_{Gi} \quad Q_{Gi}^0 - Q_{Gi}^m \quad (i=1, \dots, ng)$$

$$P_{Gij} \quad P_{Gij}^M - P_{Gij}^0 \quad (i=1, \dots, ng, j=1, \dots, nseg(i))$$

$$- P_{Gij} \quad P_{Gij}^0 \quad (i=1, \dots, ng, j=1, \dots, nseg(i))$$

The above formulation can be expressed in matrix form as follows:

$$\begin{array}{l}
 \text{MIN} \quad \begin{bmatrix} 0 & 0 & 0 & 0 \end{bmatrix} \begin{matrix} P_g \\ Q_g \\ P_g \end{matrix} \\
 \\
 \text{S.T.} \quad \begin{matrix} \mathbf{P}_g & -M_{ppg} & \mathbf{P}_g & = 0 \\
 \mathbf{J}_{PV,ref} \mathbf{V} + \mathbf{J}_{P,ref} & -\mathbf{E}_{ref} & \mathbf{P}_{gref} & = 0 \\
 \mathbf{J}_{PV} \mathbf{V} + \mathbf{J}_P & - & \mathbf{P}_g & = 0 \\
 \mathbf{J}_{QV} \mathbf{V} + \mathbf{J}_Q & & - \mathbf{Q}_g & = 0 \\
 \\
 \frac{\tilde{\mathbf{S}}^f}{\mathbf{V}} \mathbf{V} + \frac{\tilde{\mathbf{S}}^f}{\mathbf{V}} & & & \mathbf{S}^M - \mathbf{S}^0 \\
 \frac{\tilde{\mathbf{S}}^t}{\mathbf{V}} \mathbf{V} + \frac{\tilde{\mathbf{S}}^t}{\mathbf{V}} & & & \mathbf{S}^M - \mathbf{S}^0 \\
 \mathbf{V} & & & \mathbf{V}^M - \mathbf{V}^0 \\
 - \mathbf{V} & & & \mathbf{V}^0 - \mathbf{V}^m \\
 \\
 & \mathbf{P}_g & & \mathbf{P}_g^M - \mathbf{P}_g^0 \\
 & - \mathbf{P}_g & & \mathbf{P}_g^0 - \mathbf{P}_g^m \\
 & & \mathbf{Q}_g & \mathbf{Q}_g^M - \mathbf{Q}_g^0 \\
 & & - \mathbf{Q}_g & \mathbf{Q}_g^0 - \mathbf{Q}_g^m \\
 \\
 - \mathbf{P}_g^0 & \mathbf{P}_g & \mathbf{P}_g^M - \mathbf{P}_g^0 & \\
 - & \mathbf{P}_g & & \text{(Step size constraints)} \\
 - & \mathbf{Q}_g & & \text{(Step size constraints)}
 \end{matrix}
 \end{array}$$

Note: $E_{ref}, J_{PV,ref}, J_{P,ref}$ are the rows corresponding to reference bus.

The Dense Solver

The Formulation of LP Sub-problem

The voltage variables and load flow equations in sparse formulation can be eliminated based on:

$$V = \begin{bmatrix} J_{PV} & J_P & -1 & E & 0 & P_g \\ J_{QV} & J_Q & 0 & E & 0 & Q_g \end{bmatrix} \cdot$$

$$V = \begin{bmatrix} G_{11} & G_{12} & P_g \\ G_{21} & G_{22} & Q_g \end{bmatrix}$$

$$\tilde{S} = \frac{\tilde{S}}{V} \quad \tilde{S} \quad V = \frac{\tilde{S}}{V} \quad \tilde{S} \quad \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \quad V = \begin{bmatrix} T_1 & T_2 \end{bmatrix} \quad \begin{bmatrix} P_g \\ Q_g \end{bmatrix}$$

$$M_{jg} = J_{ref} \quad G$$

The well-known dense formulation of LP sub-problem is as follows:

$$MIN \quad \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} P_g \\ Q_g \\ P_g \end{bmatrix}$$

$$S.T. \quad \mathbf{P}_g \quad - M_{ppg} \quad \mathbf{P}_g = 0$$

$$(M_{jgP} - I_{ref}) \quad \mathbf{P}_g \quad + \quad M_{jgQ} \quad \mathbf{Q}_g = 0 \quad (I_{ref}: \text{ the ref row of unit matrix})$$

$$\begin{array}{lll} T_1^f \quad P_g & + & T_2^f \quad Q_g & \mathbf{S}^M - \mathbf{S}^0 \\ T_1^t \quad P_g & + & T_2^t \quad Q_g & \mathbf{S}^M - \mathbf{S}^0 \\ G_{11} \quad \mathbf{P}_g & + & G_{12} \quad \mathbf{Q}_g & \mathbf{V}^M - \mathbf{V}^0 \\ -G_{11} \quad \mathbf{P}_g & - & G_{12} \quad \mathbf{Q}_g & \mathbf{V}^0 - \mathbf{V}^m \end{array}$$

$$\begin{array}{lll} \mathbf{P}_g & & \mathbf{P}_g^M - \mathbf{P}_g^0 \\ - \mathbf{P}_g & & \mathbf{P}_g^0 - \mathbf{P}_g^m \\ & \mathbf{Q}_g & \mathbf{Q}_g^M - \mathbf{Q}_g^0 \\ & - \mathbf{Q}_g & \mathbf{Q}_g^0 - \mathbf{Q}_g^m \end{array}$$

$$\begin{array}{lll} -\mathbf{P}_g^0 & \mathbf{P}_g & \mathbf{P}_g^M - \mathbf{P}_g^0 \\ - & \mathbf{P}_g & \\ - & \mathbf{Q}_g & \end{array}$$

Solving the LP Sub-problem

It is well recognized that only a few constraints in a typical OPF problem are binding. Based on this observation, the so-called Iterative Constraint Search [B. Stott, 1979] is employed in our code to solve LP sub-problem. We will explain the algorithm in detail in coming version of our note.

Computing KT Multipliers (incomplete)

Since load flow equations do not appear explicitly in the dense LP sub-problem, so we can not get KT multipliers corresponding to load flow equations directly from LP solver, we have to compute them. Before we explain how to compute KT multipliers, let us recall what conditions optimal solution and KT multipliers meet. Applying the Kuhn-Tucker condition directly to a LP problem, we have:

$$\begin{aligned} \mathbf{A}^T \lambda &= -c \\ \mathbf{A}x &= b \\ \begin{bmatrix} \mathbf{A}^T & 0 \\ 0 & \mathbf{A} \end{bmatrix} \begin{bmatrix} \lambda \\ x \end{bmatrix} &= \begin{bmatrix} -c \\ b \end{bmatrix} \end{aligned}$$

where \mathbf{A} is composed of the binding terms of coefficients matrix \mathbf{A} of a LP problem. The above formula is the basis we compute KT multipliers. Let us rewrite the constraints of sparse LP formulation as:

$$\begin{aligned} -E_1 u + J_{p1} x &= 0 \\ -Eu + Jx &= 0 \\ \mathbf{D}u + \mathbf{B}x &= \tilde{\mathbf{S}} \end{aligned}$$

Where: u contains controllable variables, x contains state variables V and θ . Matrix \mathbf{D} and \mathbf{B} jointly denote a sub-matrix of sparse linearized LP matrix. The sub-matrix is composed of the binding rows of the inequality constraints. Binding constraints is identified after LP is solved.

Since $\mathbf{A}^T \lambda = -c$, it follows:

$$\begin{bmatrix} J_1^T & J^T & \mathbf{B}^T \\ -E_1^T & -E^T & \mathbf{D}^T \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = -c$$

We are only interested in the first equation, that is,

$$J_1^T \lambda + J^T \mu + \mathbf{B}^T \mu = 0$$

Variable λ and μ are provided by LP solver (the dual vector). Solving the above equation, we get μ .

Checking KT Conditions (incomplete)

$$\begin{aligned} L &= f_i(p_g) + \lambda_i g_i + \mu_i h_i \\ \frac{\partial L}{\partial p_{gij}} &= \frac{\partial f_i}{\partial p_{gij}} - \lambda_{ij} - \mu_{ij}^u - \mu_{ij}^l = 0 \end{aligned}$$

$$\frac{L}{P_{gi}} = - \frac{trans}{i} - \frac{lf}{i} - \mu_i^u + \mu_i^l = 0$$

$$\frac{L}{Q_{gi}} = - \frac{lf}{i} - \mu_i^u + \mu_i^l = 0$$

$$\frac{L}{V_i} = - \sum_k \frac{h_k}{V_i} - \mu_i^u + \mu_i^l - \sum_k \mu_k^s \frac{\tilde{S}_k^f}{V_i} - \sum_k \mu_k^s \frac{\tilde{S}_k^t}{V_i} = 0$$

$$\frac{L}{V_i} = \sum_k \frac{P(V_k)}{V_i} - \sum_k \mu_k^s \frac{\tilde{S}_k^f}{V_i} - \sum_k \mu_k^s \frac{\tilde{S}_k^t}{V_i} = 0$$

Computing Generation-load Shift Factors (incomplete)

This M-file is not available to users.

$$\begin{matrix} J_P & J_{PV} \\ J_Q & J_{QV} \end{matrix} V = \begin{matrix} P \\ Q \end{matrix}$$

$$\begin{matrix} J_P & + J_{PV} \\ J_Q & + J_{QV} \end{matrix} V = \begin{matrix} P \\ 0 \end{matrix}$$

$$P = \left(J_P - J_{PV} J_{QV}^{-1} J_Q \right) = J$$

$$P_{ij} = \frac{\partial P}{\partial X_{ij}} = H J^{-1} P$$

Matrix $H J^{-1}$ is known as matrix of generation-load shift factor.

The Jacobian Matrix of Load Flow

$$P(V, \theta) = V_i \sum_{j=1}^{NB} V_j \left(G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij} \right)$$

$$Q(V, \theta) = V_i \sum_{j=1}^{NB} V_j \left(G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij} \right)$$

$$\frac{\partial P_i}{\partial V_j} = V_i V_j \left(G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij} \right) \quad j \neq i$$

$$\frac{\partial Q_i}{\partial V_j} = -V_i V_j \left(G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij} \right) \quad j \neq i$$

$$\begin{aligned} \frac{\partial P_i}{\partial V_i} &= -V_i \sum_{j=1, j \neq i}^{NB} V_j \left(G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij} \right) \\ &= -Q_i - B_{ii} V_i^2 \end{aligned}$$

$$\begin{aligned} \frac{\partial Q_i}{\partial V_i} &= V_i \sum_{j=1, j \neq i}^{NB} V_j \left(G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij} \right) \\ &= P_i - G_{ii} V_i^2 \end{aligned}$$

$$\frac{\partial P_i}{\partial \theta_j} = V_i \left(G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij} \right) \quad j \neq i$$

$$\frac{\partial Q_i}{\partial \theta_j} = V_i \left(G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij} \right) \quad j \neq i$$

$$\frac{P_i}{V_i} = \frac{V_j}{V_i} (G_{ij} \cos \theta_{ij} + B_{ij} \sin \theta_{ij}) + 2V_i G_{ii} \quad \frac{Q_i}{V_i} = \frac{V_j}{V_i} (G_{ij} \sin \theta_{ij} - B_{ij} \cos \theta_{ij}) - 2V_i B_{ii}$$

$$= \frac{1}{V_i} (P_i + G_{ii} V_i^2) \quad = \frac{1}{V_i} (Q_i - B_{ii} V_i^2)$$

Linearized Formulation of Line Flow

$$S_{ij} = \frac{V_i - V_j}{R + jX} = \frac{V_i^2 - V_i V_j (\cos \theta_{ij} - j \sin \theta_{ij}) (\cos \theta_{ij} + j \sin \theta_{ij})}{R + jX} = \frac{V_i^2 - j V_i V_j \sin \theta_{ij} - V_i V_j \cos \theta_{ij}}{R^2 + X^2} (R - jX)$$

$$P_{ij} = \frac{R(V_i^2 - V_i V_j \cos \theta_{ij}) - V_i V_j X \sin \theta_{ij}}{R^2 + X^2} \quad Q_{ij} = -\frac{R V_i V_j \sin \theta_{ij} + X(V_i^2 - V_i V_j \cos \theta_{ij})}{R^2 + X^2}$$

$$\frac{P_{ij}}{V_i} = \frac{R(2V_i - V_j \cos \theta_{ij}) - V_j X \sin \theta_{ij}}{R^2 + X^2} \quad \frac{Q_{ij}}{V_i} = -\frac{R V_j \sin \theta_{ij} + X(2V_i - V_j \cos \theta_{ij})}{R^2 + X^2}$$

$$\frac{P_{ij}}{V_j} = \frac{-R V_i \cos \theta_{ij} - V_i X \sin \theta_{ij}}{R^2 + X^2} \quad \frac{Q_{ij}}{V_j} = -\frac{R V_i \sin \theta_{ij} - X V_i \cos \theta_{ij}}{R^2 + X^2}$$

$$\frac{P_{ij}}{i} = \frac{R V_i V_j \sin \theta_{ij} - V_i V_j X \cos \theta_{ij}}{R^2 + X^2} \quad \frac{Q_{ij}}{i} = -\frac{R V_i V_j \cos \theta_{ij} + X V_i V_j \sin \theta_{ij}}{R^2 + X^2}$$

$$\frac{P_{ij}}{j} = -\frac{R V_i V_j \sin \theta_{ij} - V_i V_j X \cos \theta_{ij}}{R^2 + X^2} \quad \frac{Q_{ij}}{j} = \frac{R V_i V_j \cos \theta_{ij} + X V_i V_j \sin \theta_{ij}}{R^2 + X^2}$$

$$\tilde{S}_{ij} = \sqrt{P_{ij}^2 + Q_{ij}^2} \quad \frac{\tilde{S}_{ij}}{V_i} = 2P_{ij}^0 \frac{P_{ij}}{V_i} + 2Q_{ij}^0 \frac{Q_{ij}}{V_i}$$

Test Results (incomplete)

The Combined UC-OPF Solver (incomplete)

Final Remarks (incomplete)

Reference

O. Alsac, J. Bright, M. Prais, B. Stott, "Further Developments in LP-based Optimal Power Flow", *IEEE Trans. On Power Systems*, vol. 5, no. 3, 1990, pp. 697-711

B. Stott, J.L. Marino, O. Alsac, "Review of Linear Programming Applied to Power System Rescheduling", 1979 PICA, pp 142-154