

Open-Source Software for Power Industry Research, Teaching, and Training: A DC-OPF Illustration

Junjie Sun and Leigh Tesfatsion, *Member, IEEE*

Abstract—Software currently available for power industry studies is largely proprietary. Lack of open-source access prevents users from gaining a complete and accurate understanding of what has been implemented, restricts the ability of users to experiment with new software features, and hinders users from tailoring software to specific training scenarios. This study reports on the development of a stand-alone open-source Java solver for DC optimal power flow (DC-OPF) problems suitable for research, teaching, and training purposes. The DC-OPF solver is shown to match or exceed the accuracy of BPMPD, a proprietary third-party QP solver highly recommended by MatPower, when tested on a public repository of small to medium-sized QP problems. The capabilities of the DC-OPF solver are illustrated for a 5-node DC-OPF test case commonly used for training purposes.

I. INTRODUCTION

RESTRUCTURED wholesale power markets are extraordinarily complex. In the U.S. these markets typically involve spot and forward energy markets managed by independent system operators or regional transmission organizations (ISOs/RTOs) over AC transmission grids subject to congestion effects. As reported by Joskow [2, Table 1], over 50% of the generation capacity in the U.S. is now operating under this market design, and other regions of the U.S. are moving towards this form of organization.

The complexity of restructured wholesale power markets essentially forces electricity researchers to resort to computational methods of analysis. Unfortunately, much of the software currently available for computational electricity modeling is either proprietary or based on proprietary kernel routines and libraries.¹ This restricts the ability of electricity researchers to publish self-sufficient studies permitting full access to implementation.

A key stumbling block to developing open-source software for general research into restructured wholesale power markets is the need to model the AC/DC optimal power flow (OPF) problems that must repeatedly be solved by ISOs/RTOs in order to generate daily unit commitment and dispatch schedules, as well as locational marginal prices (LMPs), for both spot and forward energy markets ([4, Chpt. 13], [5]). Developing algorithms for the successful solution of

optimization problems involving mixed collections of equality and inequality constraints, even when specialized to quadratic objective functions (as in DC-OPF approximations to AC-OPF problems), is a daunting task full of pitfalls for the unwary. Consequently, having fully open access to software implementations is essential for verification purposes.

This study reports the development by the authors of QuadProgJ, a stand-alone open-source Java solver for *strictly convex quadratic programming (SCQP)* problems that can be used to solve small to medium-sized DC-OPF problems for research, teaching, and training purposes. QuadProgJ implements the well-known dual active-set SCQP algorithm developed by Goldfarb and Idnani [6]. As detailed in [1] and summarized below, QuadProgJ matches or exceeds the accuracy of the proprietary QP solver BPMPD when tested on a public repository of SCQP problems with up to 1500 decision variables plus constraints. BPMPD has been in use since 1998, and is considered to be a high-performance QP solver for power system applications.²

To our knowledge, QuadProgJ is the first open-source SCQP solver developed in Java. As evidence of its usefulness, it has been incorporated as a critical component of an open-source Java-implemented model of a restructured wholesale power market operating over a realistically rendered transmission grid in accordance with core FERC-recommended market design features [8].

In addition, this study proposes a physically meaningful augmentation of the standard DC-OPF problem that permits the direct generation of solution values for locational marginal prices (LMPs), voltage angles, and voltage angle differences together with real power injections and branch flows. A five-node test case is used to illustrate how QuadProgJ, coupled with a Java outer shell DCOPFJ developed by the authors, directly generates complete solution values for this augmented DC-OPF problem. In particular, DCOPFJ automates the SI/PU conversion and matrix/vector representation of all needed input data for this augmented DC-OPF problem.

QuadProgJ/DCOPFJ can be downloaded as freeware³ from <http://www.econ.iastate.edu/tesfatsi/ElectricOSS.htm>.

²BPMPD is a proprietary third-party QP solver highly recommended by MatPower. It is available for downloading in executable or DLL form from a website maintained by its original developers at <http://www.sztaki.hu/~meszaros/bmpd/>. The following statement appears on page 30, Appendix A, 4th paragraph, of the MatPower user's manual [7]: "Of all of the packages tested, the bmpd solver has been the only one which worked reliably for us. It has proven to be very robust and has exceptional performance."

³Interested readers can follow the instructions on the ElectricOSS web site to compile and run the Java source code for QuadProgJ/DCOPFJ.

This study is an abridged version of Sun and Tesfatsion [1], supported in part by the National Science Foundation under Grant NSF-0527460.

Junjie Sun (corresponding author: jjsun@iastate.edu) and Leigh Tesfatsion (tesfatsi@iastate.edu) are with the Department of Economics, Iowa State University, Ames, IA 50011 USA.

¹The power system toolbox PSAT developed by Federico Milano [3] is an important exception. Although developed and tested using Matlab, a commercial product with proprietary kernel routines and libraries, PSAT can also be run on GNU/Octave, a freely-available Matlab clone.

II. DC-OPF FORMULATION

As detailed in [8], a commonly used representation for an hourly DC-OPF problem in pu terms with price-insensitive loads is as follows:

Minimize generator total variable costs

$$\sum_{i=1}^I [A_i P_{Gi} + B_i P_{Gi}^2] \quad (1)$$

with respect to real power injections and voltage angles

$$P_{Gi}, i = 1, \dots, I; \quad \delta_k, k = 1, \dots, K$$

subject to:

Real power balance constraint for each node $k = 1, \dots, K$:

$$0 = \sum_{j \in J_k} P_{Lj} - \sum_{i \in I_k} P_{Gi} + \sum_{km \text{ or } mk \in BR} F_{km} \quad (2)$$

where

$$F_{km} = B_{km} [\delta_k - \delta_m] \quad (3)$$

Real power thermal constraint for each branch km :

$$|F_{km}| \leq F_{km}^U \quad (4)$$

Real power production constraint for each generator $i = 1, \dots, I$:

$$P_{Gi}^L \leq P_{Gi} \leq P_{Gi}^U \quad (5)$$

Voltage angle setting at reference node 1:

$$\delta_1 = 0 \quad (6)$$

Here P_{Gi} denotes real power produced by generator i , $TC_i(P_{Gi}) = A_i P_{Gi} + B_i P_{Gi}^2 + FCOST_i$ denotes the total cost function for generator i , P_{Lj} denotes the real power load for the j th load-serving entity (LSE), B_{km} denotes the inverse of reactance for branch km , F_{km} denotes the real power flowing on branch km , F_{km}^U denotes the thermal limit for branch km , P_{Gi}^L and P_{Gi}^U denote the lower and upper real power production limits for generator i , and δ_1 denotes the reference node 1 voltage angle.

As shown in Sun and Tesfatsion [1], this standard DC-OPF problem can equivalently be represented in the numerically desirable form of a strictly convex quadratic programming (SCQP) problem if the balance constraints (2) are used to eliminate the voltage angles δ_k by substitution. However, this elimination prevents direct generation of solution values for LMPs since, by definition, the LMP for node k is the solution value for the multiplier (shadow price) for the k th nodal balance constraint.

For this reason, we replace the standard DC-OPF objective function (1) with the following augmented form:

$$\sum_{i=1}^I [A_i P_{Gi} + B_i P_{Gi}^2] + \pi \left[\sum_{km \in BR} [\delta_k - \delta_m]^2 \right] \quad (7)$$

where π is a positive soft penalty weight on the sum of squared voltage angle differences. As carefully demonstrated in Sun and Tesfatsion [1], the augmented DC-OPF objective function (7) provides a number of benefits based on both physical and mathematical considerations.

First, the resulting *augmented DC-OPF problem* now has a numerically desirable SCQP form permitting the direct generation of pu solution values for LMPs and voltage angles as well as for real power production levels and branch flows. Second, the validity of the DC-OPF problem as an approximation for the underlying AC-OPF problem relies on an assumption of small voltage angle differences, and the augmented DC-OPF problem permits this assumption to be subjected to systematic sensitivity tests through variations in the penalty weight π . Third, solution differences between the non-augmented and augmented forms of the DC-OPF problem can be reduced to arbitrarily small levels by selecting an appropriately small value for the penalty weight π .

As detailed in Sun and Tesfatsion [1], this augmented DC-OPF problem can be formulated into the following strictly convex quadratic programming (SCQP) form:

Minimize

$$\mathbf{f}(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{G} \mathbf{x} + \mathbf{a}^T \mathbf{x} \quad (8)$$

with respect to

$$\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_M)^T \quad (9)$$

subject to

$$\mathbf{C}_{\text{eq}}^T \mathbf{x} = \mathbf{b}_{\text{eq}} \quad (10)$$

$$\mathbf{C}_{\text{iq}}^T \mathbf{x} \geq \mathbf{b}_{\text{iq}} \quad (11)$$

where \mathbf{G} is an $M \times M$ symmetric positive definite matrix.

III. SCQP TEST RESULTS FOR QUADPROGJ

QuadProgJ implements the well-known dual active-set SCQP method developed by Goldfarb and Idnani [6] in a numerically stable way by utilizing Cholesky decomposition and QR factorization. As currently implemented, QuadProgJ is suitable for small to medium-sized SCQP problems for which speed and efficiency of computation are not critical limiting factors. Since QuadProgJ is open source, additional improvements (e.g. sparse matrix techniques) can be introduced by users as needed to further enhance its performance.

The accuracy of QuadProgJ for small to medium-sized SCQP problems has been tested using a public repository of SCQP minimization problems prepared by Maros and Mészáros [9] for testing their interior-point QP solver, BPMPD.⁴ For each of these test problems, the solution value for the minimized objective function obtained by QuadProgJ was compared against the corresponding solution value reported for BPMPD.

⁴See the BPMPD website at <http://www.sztaki.hu/~meszaros/bmpdp/> for extensive information about BPMPD. Detailed input and output data for the SCQP test cases are available online at http://www.sztaki.hu/~meszaros/public_ftp/qpdata/.

Table I reports the general structure of these SCQP test problems, the reported BPMPD solution values, and the corresponding QuadProgJ solution values.⁵ Specifically, Table I reports the relative difference (RD) between the minimum objective function value $f^* = f(x^*)$ obtained by QuadProgJ and the minimum objective function value f_{BPMPD} attained by BPMPD, where

$$\text{RD} \equiv \frac{f^* - f_{\text{BPMPD}}}{|f_{\text{BPMPD}}|} \quad (12)$$

To help ensure a fair comparison, f^* has been rounded off to the same number of decimal places as f_{BPMPD} .

In addition, Table I reports tests conducted to check whether all equality and inequality constraints are satisfied at the minimizing solution x^* obtained by QuadProgJ. More precisely, for any given SCQP test case, the equality constraints take the form

$$\mathbf{C}_{\text{eq}}^T \mathbf{x} = \mathbf{b}_{\text{eq}} \quad (13)$$

and the inequality constraints take the form

$$\mathbf{C}_{\text{iq}}^T \mathbf{x} \geq \mathbf{b}_{\text{iq}} \quad (14)$$

Let TNEC denote the total number of equality constraints for this test case (i.e. the row dimension of \mathbf{C}_{eq}^T), and let TNIC denote the total number of inequality constraints for this test case (i.e. the row dimension of \mathbf{C}_{iq}^T). Also, let x^* denote the solution obtained by QuadProgJ for this test case.

The equality constraints for each SCQP test case are checked by computing the *Equality Constraint Error (ECE)* for this test case, defined to be the TNEC \times 1 residual vector

$$\text{ECE} \equiv \mathbf{C}_{\text{eq}} \mathbf{x}^* - \mathbf{b}_{\text{eq}} \quad (15)$$

Table I reports the mean and maximum of the absolute values of the components of this ECE vector for each SCQP test case, denoted by $\text{Mean}|\text{ECE}|$ and $\text{Max}|\text{ECE}|$ respectively.

Similarly, the inequality constraints for each SCQP test case are checked by computing the *Inequality Constraint Error (ICE)*, defined to be the TNIC \times 1 residual vector

$$\text{ICE} \equiv \mathbf{C}_{\text{iq}} \mathbf{x}^* - \mathbf{b}_{\text{iq}} \quad (16)$$

Table I reports the *Number of Violated Inequality Constraints (NVIC)* for each SCQP test case, meaning the number of negative components in this ICE vector.

As seen in Table I, QuadProgJ either matches or improves on the BPMPD solutions for all of the small and medium-sized SCQP test cases for which the total number of constraints plus decision variables is less than 1500. This conclusion is supported by the observation that, for each of these test cases, the minimized objective function value $f^* = f(x^*)$ obtained by QuadProgJ either equals or is strictly smaller than the corresponding minimized objective function value f_{BPMPD} obtained by BPMPD, with no indication that the QuadProgJ solution x^* violates any equality or inequality constraints.⁶

⁵All of the results reported in Table I for QuadProgJ were obtained from runs on a laptop PC: namely, a Compaq Presario 2100 running under Windows XP SP2 (mobile AMD Athlon XP 2800+ 2.12 GHz, 496 MB of RAM). The reported results for the BPMPD solver are taken from Maros and Mészáros [9], who do not identify the hardware platform on which the BPMPD solver runs were made.

⁶Maros and Mészáros [9] do not provide constraint checks for the BPMPD solutions reported in their repository.

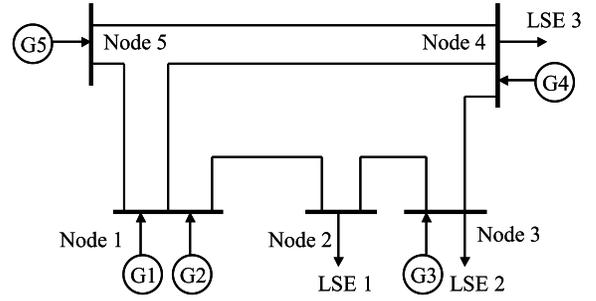


Fig. 1. A Five-Node Transmission Grid

This finding suggests that QuadProgJ is at least as accurate a solver as BPMPD for SCQP problems of this size.

IV. FIVE-NODE TEST CASE

In this section, QuadProgJ is used to solve an illustrative five-node DC-OPF test case taken from power systems texts and ISO-NE/PJM training manuals.

Table II presents SI input data for a day-ahead wholesale power market operating over a five-node transmission grid as depicted in Figure 1.⁷ This DC-OPF test case is solved by invoking QuadProgJ through the outer Java shell DCOPFJ. Specifically, given the SI input data presented in Table II, DCOPFJ invokes QuadProgJ to solve for optimal real power injections, LMPs, and various other output values. In particular, DCOPFJ automates the conversion of SI data to pu form for internal calculations and forms all needed matrix/vector representations.

Tables III and IV report the optimal solution values in SI units for real power production levels, voltage angles, LMP values, and branch flows for 24 successive hours in the day-ahead market. As indicated by these solution values, branch congestion occurs between node 1 and node 2 (and only these nodes) in each of the 24 hours. This can be verified directly by column P_{12} in Table IV, which shows that the real power flow P_{12} on branch $km = 12$ is at its upper thermal limit (250 MWs) for each hour. The direct consequence of this branch congestion is the occurrence of widespread LMP separation, i.e. the LMP values differ across all nodes for each hour. This can be verified by examining output columns LMP_1 - LMP_5 in Table IV.

Examining this LMP data more closely, it is seen that LMP_2 and LMP_3 (the LMPs for nodes 2 and 3) exhibit a sharp change in hour 17, increasing between hour 16 and hour 17 by about 100% and then dropping back to “normal” levels in hour 18 and beyond. Interesting, this type of sudden spiking in LMP values is also observed empirically in MISO’s Dynamic LMP Contour Map for real-time market prices, which is updated every five minutes; see <http://www.midwestmarket.org>.

⁷The transmission grid, reactances, and locations of Generators and LSEs, and initial hour-0 load levels for this 5-node example are adopted from an example developed by John Lally [10] for the ISO-NE that is now included in training manuals prepared by the ISO-NE and PJM. The general shape of the LSE load profiles is adopted from a 3-node example presented in Shahidepour et al. [5, pp. 405-407].

TABLE I
SCQP TEST CASES: STRUCTURAL ATTRIBUTES, BPMPD SOLUTION VALUES, AND TEST RESULTS

NAME ^a	TND ^b	TNEC ^c	TNIC ^d	fBPMPD ^e	Mean ECE ^f	Max ECE ^g	NVIC ^h	f* ⁱ	RD ^j
DUAL1	85	1	170	3.50129662E-02	0.0	0.0	0	3.50129657E-2	-1.42804239E-8
DUAL2	96	1	192	3.37336761E-02	0.0	0.0	0	3.37336761E-2	0.0
DUAL3	111	1	222	1.35755839E-01	6.66E-16	6.66E-16	0	1.35755837E-1	-1.47323313E-8
DUAL4	75	1	150	7.46090842E-01	2.11E-15	2.11E-15	0	7.46090842E-1	0.0
DUALC1	9	1	232	6.15525083E+03	2.40E-12	2.40E-12	0	6.15525083E+3	0.0
DUALC5	8	1	293	4.27232327E+02	5.33E-15	5.33E-15	0	4.27232327E+2	0.0
HS118	15	0	59	6.64820452E+02	NA ^k	NA	0	6.64820450E+2	-3.00833103E-9
HS21	2	0	5	-9.99599999E+01	NA	NA	0	-99.96	-1.00040010E-9
HS268	5	0	5	5.73107049E-07	NA	NA	0	-5.47370291E-8	-1.09550926
HS35	3	0	4	1.11111111E-01	NA	NA	0	1.11111111E-1	0.0
HS35MOD	3	0	5	2.50000001E-01	NA	NA	0	2.50000000E-1	-4.00000009E-9
HS76	4	0	7	-4.68181818E+00	NA	NA	0	-4.68181818	0.0
KSIP	20	0	1001	5.75797941E-01	NA	NA	0	5.75797941E-1	0.0
QPCBLEND	83	43	114	-7.84254092E-03	5.66E-16	8.94E-15	0	-7.84254307E-3	-2.74145844E-7
QPCBOEI1	384	9	971	1.15039140E+07	2.05E-6	9.58E-6	0	1.15039140E+7	0.0
QPCBOEI2	143	4	378	8.17196225E+06	3.42E-6	1.37E-5	0	8.17196224E+6	-1.22369628E-9
QPCSTAIR	467	209	696	6.20438748E+06	4.34E-7	6.01E-6	0	6.20438745E+6	-4.83528799E-9
S268	5	0	5	5.73107049E-07	NA	NA	0	-5.47370291E-8	-1.09550926
MOSARQP2	900	0	600	-0.159748211E+04	NA	NA	—	OOME ^l	—

^aCase name - see Maros and Mészáros [9] for a detailed description of their test case repository

^bTotal number of decision variables

^cTotal number of equality constraints

^dTotal number of inequality constraints

^eMinimizing solution value reported for the BPMPD solver

^fMean of the absolute values of ECE (Equality Constraint Error)

^gMaximum of the absolute values of ECE

^hTotal number of violated inequality constraints

ⁱMinimum objective function value as computed by QuadProgJ

^jRelative difference $[f^* - \text{fBPMPD}] / |\text{fBPMPD}|$; a negative value indicates QuadProgJ improves on BPMPD.

^kNA indicates “Not Applicable,” meaning there are no constraints of the indicated type.

^lOut-of-Memory Error indicated by a run-time Java Exception: java.lang.OutOfMemoryError

This rather dramatic LMP peaking in hour 17 can be traced to several factors. First the load profile for each LSE peaks at hour 17. Second, when solving the DC-OPF problem to meet the high load in hour 17, the ISO has to take into consideration the maximum production limit for Generator 3 as well as the thermal inequality constraint between node 1 and node 2. Both of these constraints turn out to be binding. Specifically, as seen in Table III, Generator 3 is dispatched in hour 17 at its maximum production limit (520 MWs); and, as seen in Table IV, the real power flow in branch $km = 12$ is at its upper limit (250 MWs) for all 24 hours. Given the configuration of the transmission grid, to meet the hour 17 peak load the ISO is forced to back down (relative to hour 16) the less expensive production of Generators 1 and 2 and to use instead the more expensive production of the “peaker” Generator 4.

After the peak hour 17, the load returns to lower levels. The ISO is then able to dispatch Generator 1 and 2 at their more “normal” levels, with Generator 1 at its upper production limit, and to avoid dispatching any production from Generator 4. Furthermore, the LMPs drop back to their more normal levels after hour 17.

ACKNOWLEDGMENT

The authors thank Donald Goldfarb, William Hogan, Daniel Kirschen, Chen-Ching Liu, Jim McCalley, Michael J. D. Powell, Jim Price, Harold Salazar, Johnny Wong, Tong Wu, and

especially Deddy Koesrindartoto for helpful conversations on topics related to this study.

REFERENCES

- [1] J. Sun and L. Tesfatsion, “DC optimal power flow formulation and solution using QuadProgJ,” Department of Economics, Iowa State University, Ames, Iowa, Working Paper No. 06014, 2006. [Online]. Available: <http://www.econ.iastate.edu/tesfatsi/DC-OPF.JSLT.pdf>
- [2] P. Joskow, “Markets for power in the united states: An interim assessment,” *The Energy Journal*, vol. 27, no. 1, pp. 1–36, 2006.
- [3] F. Milano, “An open source power system toolbox,” *IEEE Transactions on Power Systems*, vol. 20(3), pp. 1199–1206, 2005.
- [4] A. J. Wood and B. F. Wollenberg, *Power Generation, Operation, and Control*. New York, NY: John Wiley & Sons, Inc., 1996.
- [5] M. Shahidepour, H. Yamin, and Z. Li, *Market Operations in Electric Power Systems*. New York, NY: IEEE/Wiley-Interscience, John Wiley & Sons, Inc., 2002.
- [6] D. Goldfarb and A. U. Idnani, “A numerically stable dual method for solving strictly convex quadratic programs,” *Mathematical Programming*, vol. 27, pp. 1–33, 1983.
- [7] R. D. Zimmerman, C. E. Murillo-Sanchez, and D. D. Gan, *MATPOWER - A MATLAB Power System Simulation Package: User’s Manual*, Power Systems Engineering Research Center (PSERC), School of Electrical Engineering, Cornell University, 2005.
- [8] J. Sun and L. Tesfatsion, “Dynamic testing of wholesale power market designs: An open-source agent-based framework,” Department of Economics, Iowa State University, Ames, Iowa, Working Paper No. 06025, 2006. [Online]. Available: <http://www.econ.iastate.edu/tesfatsi/DynTestAMES.JSLT.pdf>
- [9] I. Maros and C. Mészáros, “A repository of convex quadratic programming problems,” Department of Computing, Imperial College, London, U.K., Department Technical Report DOC 97/6, 1997.
- [10] J. Lally, “Financial transmission rights: Auction example,” in *Financial Transmission Rights Draft 01-10-02*, m-06 ed. ISO New England, Inc., January 2002, section 6.

TABLE II
5-NODE DC-OPF INPUT DATA IN STANDARD UNITS (SI)

Base Values ^a											
S_o	V_o										
100	10										
K^b	π^c										
5	0.05										
Branch											
From	To	F_{km}^U (SI) ^d	x^e								
1	2	250	0.0281								
1	4	150	0.0304								
1	5	400	0.0064								
2	3	350	0.0108								
3	4	240	0.0297								
4	5	240	0.0297								
Gen ID	atNode	FCost (SI) ^f	A (SI) ^g	B (SI) ^h	P_G^L (SI) ⁱ	P_G^U (SI) ^j					
1	1	16	14	0.005	0	110					
2	1	19	15	0.006	0	100					
3	3	28	25	0.010	0	520					
4	4	10	30	0.012	0	200					
5	5	24	10	0.007	0	600					
LSE ID	atNode	L-00 ^k	L-01	L-02	L-03	L-04	L-05	L-06	L-07		
1	2	350.00	322.93	305.04	296.02	287.16	291.59	296.02	314.07		
2	3	300.00	276.80	261.47	253.73	246.13	249.93	253.73	269.20		
3	4	250.00	230.66	217.89	211.44	205.11	208.28	211.44	224.33		
ID	atNode	L-08	L-09	L-10	L-11	L-12	L-13	L-14	L-15		
1	2	358.86	394.80	403.82	408.25	403.82	394.80	390.37	390.37		
2	3	307.60	338.40	346.13	349.93	346.13	338.40	334.60	334.60		
3	4	256.33	282.00	288.44	291.61	288.44	282.00	278.83	278.83		
ID	atNode	L-16	L-17	L-18	L-19	L-20	L-21	L-22	L-23		
1	2	408.25	448.62	430.73	426.14	421.71	412.69	390.37	363.46		
2	3	349.93	384.53	369.20	365.26	361.47	353.73	334.60	311.53		
3	4	291.61	320.44	307.67	304.39	301.22	294.78	278.83	259.61		

^aFor simplicity, base voltage V_o and base apparent power S_o are set to give a base impedance $Z_o = V_o^2/S_o = 1$.

^bNumber of nodes

^cSoft penalty weight π for voltage angle differences

^dUpper limit (in MWs) on the magnitude of the flow of real power in branch km

^eReactance x_{km} (in ohms) for branch km

^fFixed costs (in \$/h) for generator i

^gCost coefficient (in \$/MWh) for generator i

^hCost coefficient (in \$/MW²h) for generator i

ⁱLower limit (in MWs) on real power production for generator i

^jUpper limit (in MWs) on real power production for generator i

^kL-H: Load (in MWs) for hour H, where H=00,01,...,23

TABLE III
5-NODE DC-OPF SOLUTION (SI): REAL POWER PRODUCTION LEVELS (IN MWs) AND VOLTAGE ANGLES (IN RADIANS)

Hour	P_{G1}^*	P_{G2}^*	P_{G3}^*	P_{G4}^*	P_{G5}^*	δ_2^*	δ_3^*	δ_4^*	δ_5^*
00	110.00	13.87	332.53	0.00	443.59	-0.0702	-0.0595	-0.0394	0.0164
01	110.00	13.44	269.41	0.00	437.54	-0.0702	-0.0624	-0.0385	0.0162
02	110.00	13.16	227.70	0.00	433.54	-0.0702	-0.0643	-0.0379	0.0161
03	110.00	13.01	206.66	0.00	431.52	-0.0703	-0.0653	-0.0376	0.0160
04	110.00	12.87	185.99	0.00	429.53	-0.0703	-0.0662	-0.0373	0.0160
05	110.00	12.95	196.33	0.00	430.53	-0.0702	-0.0658	-0.0375	0.0160
06	110.00	13.01	206.66	0.00	431.52	-0.0703	-0.0653	-0.0376	0.0160
07	110.00	13.30	248.75	0.00	435.55	-0.0703	-0.0633	-0.0382	0.0162
08	110.00	14.01	353.20	0.00	445.58	-0.0703	-0.0585	-0.0397	0.0164
09	110.00	14.58	437.00	0.00	453.61	-0.0702	-0.0546	-0.0409	0.0166
10	110.00	14.73	458.03	0.00	455.63	-0.0702	-0.0536	-0.0412	0.0167
11	110.00	14.80	468.37	0.00	456.62	-0.0702	-0.0532	-0.0413	0.0167
12	110.00	14.73	458.03	0.00	455.63	-0.0702	-0.0536	-0.0412	0.0167
13	110.00	14.58	437.00	0.00	453.61	-0.0702	-0.0546	-0.0409	0.0166
14	110.00	14.51	426.67	0.00	452.62	-0.0702	-0.0551	-0.0407	0.0166
15	110.00	14.51	426.67	0.00	452.62	-0.0702	-0.0551	-0.0407	0.0166
16	110.00	14.80	468.37	0.00	456.62	-0.0702	-0.0532	-0.0413	0.0167
17	2.07	0.00	520.00	108.88	522.63	-0.0702	-0.0488	-0.0300	0.0222
18	107.35	6.12	520.00	0.00	474.13	-0.0702	-0.0507	-0.0418	0.0175
19	110.00	15.08	510.08	0.00	460.63	-0.0702	-0.0512	-0.0419	0.0168
20	110.00	15.01	499.76	0.00	459.63	-0.0702	-0.0517	-0.0418	0.0168
21	110.00	14.87	478.71	0.00	457.62	-0.0702	-0.0527	-0.0415	0.0167
22	110.00	14.51	426.67	0.00	452.62	-0.0702	-0.0551	-0.0407	0.0166
23	110.00	14.09	363.91	0.00	446.60	-0.0702	-0.0580	-0.0399	0.0164

TABLE IV

5-NODE DC-OPF SOLUTION (SI): LMPs (NODAL BALANCE CONSTRAINT MULTIPLIERS IN \$/MWh), TOGETHER WITH THE REAL POWER BRANCH FLOW P_{km} (IN MWs) AND ASSOCIATED THERMAL LIMIT P_{km}^U (IN MWs) FOR EACH BRANCH km

Hour	LMP ₁	LMP ₂	LMP ₃	LMP ₄	LMP ₅	P_{12}^a	P_{14}	P_{15}	P_{23}	P_{34}	P_{45}
00	15.17	35.50	31.65	21.05	16.21	250.00	129.65	-255.77	-100.00	-67.47	-187.82
01	15.16	33.95	30.39	20.60	16.13	250.00	126.71	-253.27	-72.93	-80.32	-184.27
02	15.16	32.92	29.55	20.30	16.07	250.00	124.77	-251.61	-55.04	-88.81	-181.93
03	15.16	32.40	29.13	20.15	16.04	250.00	123.79	-250.77	-46.02	-93.09	-180.74
04	15.15	31.89	28.72	20.00	16.01	250.00	122.83	-249.95	-37.16	-97.30	-179.58
05	15.16	32.15	28.93	20.07	16.03	250.00	123.31	-250.36	-41.59	-95.19	-180.16
06	15.16	32.40	29.13	20.15	16.04	250.00	123.79	-250.77	-46.02	-93.09	-180.74
07	15.16	33.44	29.97	20.45	16.10	250.00	125.75	-252.45	-64.07	-84.52	-183.11
08	15.17	36.01	32.06	21.20	16.24	250.00	130.61	-256.60	-108.86	-63.26	-188.98
09	15.18	38.08	33.74	21.81	16.35	250.00	134.51	-259.92	-144.80	-46.20	-193.69
10	15.18	38.60	34.16	21.96	16.38	250.00	135.49	-260.76	-153.82	-41.92	-194.87
11	15.18	38.85	34.37	22.03	16.39	250.00	135.97	-261.17	-158.25	-39.81	-195.45
12	15.18	38.60	34.16	21.96	16.38	250.00	135.49	-260.76	-153.82	-41.92	-194.87
13	15.18	38.08	33.74	21.81	16.35	250.00	134.51	-259.92	-144.80	-46.20	-193.69
14	15.17	37.82	33.53	21.73	16.34	250.00	134.03	-259.51	-140.37	-48.30	-193.11
15	15.17	37.82	33.53	21.73	16.34	250.00	134.03	-259.51	-140.37	-48.30	-193.11
16	15.18	38.85	34.37	22.03	16.39	250.00	135.97	-261.17	-158.25	-39.81	-195.45
17	14.02	78.24	66.07	32.61	17.32	250.00	98.83	-346.76	-198.62	-63.15	-175.88
18	15.07	45.55	39.78	23.90	16.64	250.00	137.64	-274.17	-180.73	-29.93	-199.96
19	15.18	39.88	35.20	22.33	16.45	250.00	137.91	-262.83	-176.14	-31.32	-197.80
20	15.18	39.63	35.00	22.26	16.43	250.00	137.43	-262.42	-171.71	-33.42	-197.22
21	15.18	39.11	34.57	22.11	16.41	250.00	136.45	-261.58	-162.69	-37.71	-196.03
22	15.17	37.82	33.53	21.73	16.34	250.00	134.03	-259.51	-140.37	-48.30	-193.11
23	15.17	36.28	32.28	21.28	16.25	250.00	131.11	-257.02	-113.46	-61.08	-189.58
						P_{12}^U	P_{14}^U	P_{15}^U	P_{23}^U	P_{34}^U	P_{45}^U
						250.00	150.00	400.00	350.00	240.00	240.00

^aAs usual, the real power P_{km} flowing along a branch km is positive in value if and only if the flow is from node k to node m .