Optimal power flow formulations and their impacts on the performance of solution methods

Sereeter, Baljinnyam; Vuik, Cornelis; Witteveen, Cees; Palensky, Peter

DOI
10.1109/PESGM40551.2019.8973585

Publication date
2019

Document Version
Final published version

Published in
2019 IEEE Power and Energy Society General Meeting, PESGM 2019

Citation (APA)

Important note
To cite this publication, please use the final published version (if applicable). Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights. We will remove access to the work immediately and investigate your claim.
Green Open Access added to TU Delft Institutional Repository

‘You share, we take care!’ – Taverne project

https://www.openaccess.nl/en/you-share-we-take-care

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.
Optimal power flow formulations and their impacts on the performance of solution methods

Baljinnyam Sereeter†, Cornelis Vuik†, Cornelis Witteveen† and Peter Palensky†
†Faculty of Electrical Engineering, Mathematics and Computer Science,
Delft University of Technology, The Netherlands
*Email: b.sereeter@tudelft.nl

Abstract—In this paper, we study four equivalent mathematical formulations of the Optimal Power Flow (OPF) problem and their impacts on the performance of solution methods. We show how four mathematical formulations of the OPF problem can be obtained by rewriting equality constraints given as the power flow problem into four equivalent mathematical equations using power balance or current balance equations in polar or Cartesian coordinates while keeping the same physical formulation. All four mathematical formulations are implemented in Matpower. In order to identify the formulation that results in the best convergence characteristics for the solution method, we apply MIPs, KNITRO, and FMINCON on various test cases using three different initial conditions. We compare all four formulations in terms of impact factors on the solution method such as a number of nonzero elements in the Jacobian and Hessian matrices, a number of iterations and computational time on each iteration. The numerical results show that the performance of the OPF solution method is not only dependent upon the choice of the solution method itself, but also upon the exact mathematical formulation used to specify the OPF problem.

I. INTRODUCTION

The OPF problem provides the optimal operational state of the electrical power system while satisfying system constraints and control limits. Many sub-classes of the OPF problem have been developed over the years using various objective functions, control variables and system constraints such as economic dispatch, security constrained OPF (SCOPF), unit commitment, loss minimization and probabilistic OPF (POPF) [1]–[3]. These OPF problems are physical formulations that are derived from the physical properties of actual power systems. In general, the Power Flow (PF) problem is used as the main equality constraints for the OPF problem. Moreover, the PF problem is given in complex numbers and can be rewritten into four equivalent mathematical equations given in real numbers and variables using power balance or current balance equations in polar or Cartesian coordinates [4], [5]. Therefore, we obtain four mathematical formulations of the OPF problem for a single physical formulation. These four formulations are equivalent since we just rewrite the mathematical equations for the equality constraints while keeping the same physical formulation. Due to the different mathematical equations, however, each formulation can result in different numerical and analytical properties for the OPF solution method.

In practice, researchers develop a new method or do the simulation based on only one (at most two) mathematical formulation of the OPF problem and compare the result with another method using the other formulation. The formulation having power balance equations in polar coordinates (known as Polar power-voltage) is mostly used in the literature. It is questionable how an OPF solution method performs if we change the chosen formulation to the other three mathematical formulations. When the OPF solver using one formulation does not converge, can the same method using another formulation converge? Which mathematical formulation results in the smallest computational time for each iteration of the solution method? Which formulation is more robust to the change of initial conditions? As far as we know no complete comparison exists between these four mathematical formulations of the OPF problem.

In [6], [7], three formulations (Polar Power-Voltage (PSV), Rectangular Power-Voltage (RSV) and Rectangular Current-Voltage (RIV)) are used to compare optimization software packages such as SNOPT, IPOPT, and KNITRO. Both papers suggest numerous strategies for choosing the initial condition. Both PSV and RIV formulations show the best performance in terms of CPU time in [7] whereas the formulation using rectangular coordinates is preferred in [6]. Furthermore, formulations PSV and RSV in [6], [7] have the same nonlinear power balance equations in different coordinates used as equality constraints for the OPF problem. However, the RIV formulation used in both papers has the linear current balance equations where the injected complex current at buses is specified and not computed from specified complex power as given in [8], [9]. Thus, the RIV formulation is not equivalent to PSV and RSV formulations. Additionally, the formulation Polar Current-Voltage (PIV) is not considered in both papers. Therefore, the comparison in [6], [7] is not complete due to missing and inequivalent formulations.

In this paper, we study all four equivalent mathematical formulations of the OPF problem and try to understand which formulation results in the best performance for OPF solution methods. We consider the OPF problem with minimization of active power generation costs as a cost function, power flow equations as equality constraints and squared apparent power limits as inequality constraints. All four mathematical formulations of the OPF problem are implemented in Matpower which is a Matlab package for solving power flow and optimal power flow problems. Originally, Matpower had only one formulation using the power balance equations in polar coordinates for the OPF computation. The other three formulations will be
included in the next version of Matpower and two technical notes [10], [11] are written for the theoretical explanation. We use optimization solvers such as Matpower’s Interior Point Method (MIPS) [12], KNITRO, and Matlab’s FMINCON for the comparison of all four formulations. We test all three solvers on various test cases taken from Matpower and IEEE PES Power Grid. Three different initial conditions are used in the numerical experiments.

This paper is structured as follows. The physical and mathematical formulations of the OPF problem are described in Section II. Numerical results of MIPS, KNITRO, and FMINCON using three different starting points on various test cases are compared for all four mathematical formulations in Section III. Finally, Section IV describes the conclusions obtained from the results of this paper.

II. OPF FORMULATIONS

The general OPF problem can be written as follows:

\[
\text{minimize } f(x, u) \\
\text{subject to } g(x, u) = 0, \quad h(x, u) \leq 0
\]

where \(x\) and \(u\) are vectors with the state and control variables respectively, and \(f(x, u)\) is the objective function to be minimized (maximized). The vector functions \(g(x, u)\) and \(h(x, u)\) represent equality and inequality constraints respectively.

A. Variables

In general, state variables \(x\) include bus voltage magnitude \(|V_i|\), bus voltage angle \(\delta_i\), branch power flow \(S_{ij}^\text{r}\), generator active \(P_i^a\) and reactive \(Q_i^a\) power outputs, the real \(V_i^r\) and imaginary \(V_i^m\) parts of the complex voltage respectively. Control variables \(u\) are generally chosen as active power generations, voltage magnitudes at generator buses, transformer tap settings, transformer phase shifters, generator voltage control settings, load shedding, shunt reactive devices, HVDC stations and Static Var Controllers [1].

B. Objective function

In this paper, we consider the objective function \(f(x, u)\) as:

\[
f(x, u) = \sum_{i=1}^{N_g} (C_i^0 + C_i^1 P_i^a + C_i^2 (P_i^a)^2)
\]

where \(N_g\) is a number of generators in the network and \(C_i^0, C_i^1, C_i^2\) are the positive coefficients of the polynomial cost functions. Moreover, the objective is to minimize the total cost for the active power generation in the system.

C. Equality constraints

Usually, the power flow equations are used as equality constraints \(g(x, u)\):

\[
S_i = V_i \sum_{k=1}^{N_b} Y^e_{ik} V_k^* \quad \forall i \in 1, ..., N
\]

where \(N_b\) is a number of buses in the network, \(S_i\) is the injected complex power, \(V_i\) is the complex voltage at bus and \(Y^e_{ij}\) is an element of the admittance matrix. Moreover, the power flow problem (3) can be rewritten into four equivalent mathematical equations given in real numbers and variables using the power balance or current balance equations in polar or Cartesian coordinates [4], [5] as given in sections (II-C1)-(II-C4).

1) Power balance equations in polar coordinates (PP):

\[
g_1(x, u) = \begin{bmatrix} \sum_{k=1}^{N_b} |V_i||V_k|(G_{ik} \cos \delta_{ik} + B_{ik} \sin \delta_{ik}) - P_i^a \\ \sum_{k=1}^{N_b} |V_i||V_k|(G_{ik} \sin \delta_{ik} - B_{ik} \cos \delta_{ik}) - Q_i^a \end{bmatrix} = 0
\]

where \(G_{ij}\) and \(B_{ij}\) are the conductance and the susceptance of the transmission line between bus \(i\) and \(j\) respectively.

2) Power balance equations in Cartesian coordinates (PC):

\[
g_2(x, u) = \begin{bmatrix} \sum_{k=1}^{N_b} (V_i^r(G_{ik} V_k^r - B_{ik} V_k^m) + V_i^m(B_{ik} V_k^r + G_{ik} V_k^m)) - P_i^a \\ \sum_{k=1}^{N_b} (V_i^r(G_{ik} V_k^r - B_{ik} V_k^m) - V_i^m(B_{ik} V_k^r + G_{ik} V_k^m)) - Q_i^a \end{bmatrix} = 0
\]

3) Current balance equations in polar coordinates (CP):

\[
g_3(x, u) = \begin{bmatrix} \sum_{k=1}^{N_b} |V_k|(G_{ik} \cos \delta_{ik} - B_{ik} \sin \delta_{ik}) \\ \sum_{k=1}^{N_b}|V_k|(G_{ik} \sin \delta_{ik} + B_{ik} \cos \delta_{ik}) \end{bmatrix}
\]

4) Current balance equations in Cartesian coordinates (CC):

\[
g_4(x, u) = \begin{bmatrix} \sum_{k=1}^{N_b} (G_{ik} V_k^r + B_{ik} V_k^m) - P_i^a V_i^r + Q_i^a V_i^m \\ \sum_{k=1}^{N_b} (G_{ik} V_k^r - B_{ik} V_k^m) - P_i^a V_i^r - Q_i^a V_i^m \end{bmatrix}
\]

D. Inequality constraints

The inequality constraints are specified using the maximum and minimum limits for transmission lines, control, and state variables.

1) Branch flow limits: We consider inequality constraints \(h(x, u)\) as squared branch flow limits for the apparent power:

\[
h_{ij}(x, u) = \left[\frac{|S_{ij}^f(x, u)|^2}{|S_{ij}^f(x, u)|^2}\right] \leq \left[\frac{|S_{ij}^{\text{max}}|^2}{|S_{ij}^{\text{max}}|^2}\right]
\]

where \(S_{ij}^f(x, u)\) and \(S_{ij}^{\text{max}}\) are the apparent power of branch flow from side and to side respectively, \(S_{ij}^{\text{max}}\) is the maximum branch flow limits between bus \(i\) and \(j\). We denote a number of transmission lines in the network by \(N_l\).
2) Variable limits: The following variable limits are considered in this paper:

\[ |V_i|^{\text{min}} \leq |V_i| \leq |V_i|^{\text{max}}, \]  
\[ (P_i^g)^{\text{min}} \leq P_i^g \leq (P_i^g)^{\text{max}}, \]  
\[ (Q_i^g)^{\text{min}} \leq Q_i^g \leq (Q_i^g)^{\text{max}}, \]  
\[ |V_i|^2 \leq (V_i^m)^2 \leq |V_i|^{\text{max}}. \]  

E. Four equivalent formulations of the OPF problem

Combining (2) and (8) with one of (4)-(7) depending on the choice of the formulation and coordinates, we can obtain four equivalent mathematical formulations for a single physical formulation of the OPF problem (1). Table I shows the summary of all four formulations for the number of variables, equality and inequality constraints.

### Table I: Summary of all four formulations

<table>
<thead>
<tr>
<th>Coordinates</th>
<th>PP</th>
<th>CP</th>
<th>PC</th>
<th>CC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables</td>
<td>(</td>
<td>V_i, P_i^g, Q_i^g</td>
<td>)</td>
<td>(</td>
</tr>
<tr>
<td>Nonlinear</td>
<td>Power balance in Polar</td>
<td>Current balance in Cartesian</td>
<td>Power balance in Cartesian</td>
<td>Current balance in Cartesian</td>
</tr>
<tr>
<td>equality</td>
<td>(4) ( 2N_b )</td>
<td>(6) ( 2N_b )</td>
<td>(5) ( 2N_b )</td>
<td>(7) ( 2N_b )</td>
</tr>
<tr>
<td>constraints</td>
<td>Branch apparent power flow ( (8) )</td>
<td>Variable limits ( (12) )</td>
<td>( N_b )</td>
<td>( N_b )</td>
</tr>
</tbody>
</table>

### III. Numerical Results

In this section, we present the result of numerical experiments of all four mathematical formulations in order to verify the formulation resulting in the best performance for the OPF solution method. We implement all four mathematical formulations in Matpower and apply three optimization software packages such as MIPS, KNITRO, and FMINCON. In the numerical experiments, we use test cases from Matpower and IEEE PES Power Grid Library (PGLib) that are given in Table II. The following impact factors on the solution method are considered for the comparison:

- number of nonzero elements (NNZ) in the Jacobian and Hessian matrices
- number of iterations for the solution method
- computational time for each iteration of the solution method.

Both feasibility and optimality tolerances are set to \( 10^{-6} \) and the number of iterations is limited by 450. The constant power load model is considered for all loads. The performance of the non-convex optimization problems such as OPF problems strongly depends on the choice of starting points. Therefore, we use three different initial conditions for all solution methods as given in Table III. All experiments are performed on an Intel computer i5-4690 3.5 GHz CPU with four cores and 64 Gb memory, running a Debian 64-bit Linux 8.7 distribution.

### Table II: Description of considered test cases

<table>
<thead>
<tr>
<th>Systems</th>
<th>Buses</th>
<th>Generators</th>
<th>Branches</th>
<th>Abbr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matpower-case89</td>
<td>89</td>
<td>12</td>
<td>21</td>
<td>c89</td>
</tr>
<tr>
<td>PGLib-case118</td>
<td>118</td>
<td>54</td>
<td>186</td>
<td>c118</td>
</tr>
<tr>
<td>Matpower-case300</td>
<td>300</td>
<td>69</td>
<td>411</td>
<td>c300</td>
</tr>
<tr>
<td>PGLib-case588</td>
<td>588</td>
<td>167</td>
<td>686</td>
<td>c588</td>
</tr>
<tr>
<td>PGLib-case2383</td>
<td>2383</td>
<td>327</td>
<td>2896</td>
<td>c2383</td>
</tr>
<tr>
<td>Matpower-case2736</td>
<td>2736</td>
<td>420</td>
<td>3504</td>
<td>c2736</td>
</tr>
<tr>
<td>Matpower-case3120</td>
<td>3120</td>
<td>505</td>
<td>3693</td>
<td>c3120</td>
</tr>
</tbody>
</table>

### Table III: Three options for the initial condition

<table>
<thead>
<tr>
<th>Options</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>IC-1</td>
<td>Interior point estimation (midpoint of their bounds)</td>
</tr>
<tr>
<td>IC-2</td>
<td>Use the current state in given test case</td>
</tr>
<tr>
<td>IC-3</td>
<td>Solve the power flow problem and use the resulting state</td>
</tr>
</tbody>
</table>

### A. Number of nonzero elements

Table IV shows the number of nonzero elements in the Jacobian and Hessian matrices that are recomputed at each iteration of MIPS. The best result is highlighted in bold. For the Jacobian matrix, there is no big difference between all four mathematical formulations. However, both formulations using the current balance equations (CP and CC) result in less nonzero entries for the Hessian matrix compared to PP and PC formulations. Especially, the CC formulation gives the smallest number of nonzero elements for the Hessian matrix on all test cases. Therefore, the CC formulation is the best choice for computing the Jacobian and Hessian matrices with respect to memory requirements. The IPM algorithm assembles the

### Table IV: Number of nonzero elements in the Jacobian and Hessian matrices after one iteration of MIPS

<table>
<thead>
<tr>
<th>NNZ</th>
<th>Test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>c118</td>
<td>c300</td>
</tr>
<tr>
<td>Jacobian</td>
<td>PP</td>
</tr>
<tr>
<td>PC</td>
<td>2046</td>
</tr>
<tr>
<td>CP</td>
<td>2152</td>
</tr>
<tr>
<td>CC</td>
<td>2118</td>
</tr>
<tr>
<td>Hessian</td>
<td>PP</td>
</tr>
<tr>
<td>PC</td>
<td>1670</td>
</tr>
<tr>
<td>CP</td>
<td>894</td>
</tr>
<tr>
<td>CC</td>
<td>864</td>
</tr>
</tbody>
</table>
TABLE V: Condition number of the reduced and linearized KKT conditions after one iteration of MIPS on test case c3120

<table>
<thead>
<tr>
<th>ICs</th>
<th>Formulations</th>
<th>Test cases</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PP</td>
<td>PC</td>
</tr>
<tr>
<td>IC-1</td>
<td>8.95 x 10^2</td>
<td>5.01 x 10^2</td>
</tr>
<tr>
<td>IC-2</td>
<td>1.57 x 10^3</td>
<td>1.21 x 10^3</td>
</tr>
<tr>
<td>IC-3</td>
<td>1.43 x 10^3</td>
<td>1.39 x 10^4</td>
</tr>
</tbody>
</table>

1) MIPS: In Table VI, we provide the number of iterations of MIPS using three different starting points on various test cases. From the table, we see that PP and CP formulations result in a faster convergence for MIPS compared to PC and CC formulations for most of the test cases. Between PP and CP formulations, MIPS using the CP formulation is slightly better. Regarding the initial conditions, IC-1 shows the robust performance for MIPS on all test cases. Both initial conditions IC-2 and IC-3 bring a Non-Convergence (NC) for two test cases (c89 and c2383). MIPS using the PP formulation diverge for both IC-2 and IC-3 on these two cases whereas CC and CP formulations deliver just one NC on those test cases. The PC formulation is the slowest variant but results in the robust convergence properties for MIPS on all scenarios. However, MIPS with the PC formulation is the slowest variant in terms of iterations. When a variant of MIPS using polar coordinates cannot converge to the optimal solution for some problems, another variant using Cartesian coordinates can be a good replacement.

2) KNITRO: Table VII show the number of iterations of KNITRO using three different starting points. According to the table, KNITRO with the PP formulation is the fastest variant overall in terms of iterations. However, as we have seen in the previous section, the PP formulation also provides the bad performance for KNITRO using IC-2 on test cases c89 and c2636. Moreover, the other three variants of KNITRO perform better than KNITRO using PP on those test cases. Regarding the initial conditions, all four variants of KNITRO converge to the optimal solution for all three initial conditions. Moreover, KNITRO using IC-1 converges faster than KNITRO using IC-2 and IC-3 in terms of iterations.

2) KNITRO: Table VII show the number of iterations of KNITRO using three different starting points. According to the table, KNITRO with the PP formulation is the fastest variant overall in terms of iterations. However, as we have seen in the previous section, the PP formulation also provides the bad performance for KNITRO using IC-2 on test cases c89 and c2636. Moreover, the other three variants of KNITRO perform better than KNITRO using PP on those test cases. Regarding the initial conditions, all four variants of KNITRO converge to the optimal solution for all three initial conditions. Moreover, KNITRO using IC-1 converges faster than KNITRO using IC-2 and IC-3 in terms of iterations.

3) FMINCON: Matlab’s optimization solver FMINCON has various choices for the solution algorithm. In this work, we use the algorithm-4 that applies Interior point with user-supplied Hessian. In Table VIII, we display the number of iterations of FMINCON using three different starting points on various test cases. All four variants of FMINCON performs differently depending on the choice of the initial condition and the test case. Overall there is no formulation that is better than others. The PP formulation shows a bad performance for FMINCON on many test cases. Furthermore, PC and CC formulations which are the worst choice for MIPS and KNITRO, show the best performance for FMINCON on many test cases.

C. CPU time on each iteration

In Figure 1, the computational time on each iteration of all three solvers (CPU time / Number of iterations) is plotted for the comparison of all four formulations. From the figure, we discover that CP formulation shows the smallest computational time on each iteration for all three solvers. Additionally, all three solvers (MIPS, KNITRO and FMINCON) converge to the same objective value for all three initial conditions and four mathematical formulations on each test case.

IV. Conclusion

In this paper, we studied four equivalent mathematical formulations (PP, PC, CP, and CC) of the OPF problem and their computational impacts on the performance of the
OPF solution methods. In order to identify the mathematical formulation resulting in the best computational properties for the OPF solution method, the numerical experiments were carried out using MIPS, KNITRO, and FMINCON on various test cases of Matpower and IEEE PES Power Grid Library. All four mathematical formulations were compared in terms of the impact factors on the solution method such as the number of iterations and computational time on each iteration.

For MIPS, the CP formulation showed the fastest convergence and the smallest number of nonzero elements in the Jacobian and Hessian matrices, number of iterations and computational time on each iteration.

The numerical results showed that the performance of the OPF solution method is not only dependent upon the choice of the solution method itself, but also upon the exact mathematical formulation used to specify the OPF problem. When the OPF solution method using a certain formulation does not converge, one can obtain the optimal solution by just applying the other equivalent formulation while keeping the same algorithm.

Another contribution of this paper is the implementation of all four mathematical formulations of the OPF problem in Matpower. Originally, Matpower had only one formulation using the power balance equations in polar coordinates (PP) for the OPF computation. Therefore, the other three formulations (PC, CP, and CC) of the OPF problem were implemented in Matpower and will be included in the next version. Additionally, two technical notes [10], [11] were written to specify the first and second order derivatives of the equality and inequality constraints.

For the subsequent research, all four formulations can be applied to other OPF problems with different load models, objective functions and inequality constraints using other deterministic optimization methods or heuristic optimization methods.

ACKNOWLEDGMENT

This research is supported by NWO (the Netherlands Organization for Scientific Research Science), domain Applied and Engineering Sciences, Grant No 14181.

REFERENCES