Efficient Implementation of the OPF Problem by Interior-Point Methods

Walter Vargas Contreras, Catalina Gómez-Quiles and Antonio Gómez-Expósito

Departamento de Ingeniería Eléctrica
E.T.S.I., Universidad de Sevilla
Camino de los Descubrimientos, s/n. Isla de la Cartuja, 41092 Sevilla (España)
Phone:+34 954 487283, fax:+34 954 487284, e-mail: walvarcon@alum.us.es, catalinagq@us.es, age@us.es.

Abstract. In this paper we present an efficient implementation of the optimal power flow problem by interior point methods, including the basic primal dual algorithm as well as higher order schemes: Predictor Corrector, Multiple Centrality Corrections and Weighted Multiple Centrality Corrections. Using the total generation cost as objective function, those algorithms are programmed in Matlab with the help of an efficient complex matrix notation. Convergence rates and solution times for benchmark systems ranging from 14 to 3120 buses are presented.

Keyworks

Optimal power flow, generation dispatch, interior point methods, MATLAB, complex matrices.

1. Introduction

The formulation of the Optimal Power Flow (OPF) problem leads in a natural manner to a Nonlinear Programming (NLP) problem in which a set of nonlinear equations (optimality conditions) has to be solved. Several effective methods have been developed to solve this NLP problem, among which the Interior Point Method (IPM) has become the most popular due to its convergence robustness, computational efficiency, easy handling of inequality constraints and possibility of starting from a non-feasible point.

In this paper, the following IPM variants of the OPF have been efficiently implemented in MATLAB programming language: Primal Dual (PD), by Megiddo [1]; Predictor Corrector (PC), by Mehrotra [2]; Multiple Centrality Corrections (MCC), by Gondzio [3] and Weighted Multiple Centrality Corrections (WMCC), by Colombo-Gondzio [4]. Schemes PC, MCC and WMCC, known in the literature as higher-order IPMs, are aimed at improving the accuracy with which the Newton direction approximates the KKT equations.

Several contributions can be found in the literature regarding the application of IPMs to the OPF problem, such as the proposals by Granville [5] and Martinez *et. al.* [6, 7], where the PD IPM is applied to the optimal reactive power dispatch and active power loss reduction respectively. Wu *et al.* [8] implemented the PC IPM to minimize generation costs and active power losses. Torres and Quintana [9, 10] used the PC and MCC IPMs to optimize the active power losses, whereas the more recent contribution by Zhiguang and Quanyuan [11] employs the WMCC method to minimize generation costs.

In this paper, the above-mentioned IPM schemes are applied to the problem of minimizing generation costs. The main contribution lies in the development of an OPF formulation entirely based on the use of complex matrix notation, following the ideas proposed in [12], leading to a very compact and efficient Matlab code. Results on a number of benchmark cases, ranging from 14 to 3120 buses, are presented and compared with those provided by MATPOWER program [13].

2. OPF Formulation

The OPF problem can be written as a NLP problem, as follows:

$$\begin{aligned} &\min f(x) \\ &\text{s.t. } g(x) = 0 \\ &\quad h(x) \le 0 \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^{n_x}$ represents the variables of the problem, $f(x) \in \mathbb{R}$ is the objective function, $g(x) \in \mathbb{R}^{n_g}$ is the set of equality constraints and $h(x) \in \mathbb{R}^{n_h}$ represents the inequality constraints.

In this paper, the variables considered are: voltage magnitude and voltage phase for each bus (state variables), as well as the active and reactive power of generators (control variables). The equality constraints are the nonlinear active and reactive power balance equations. Inequality constraints include limits on active and reactive powers of generators, bus voltage magnitude limits, and maximum apparent power for lines and transformers.

3. Interior Point Algorithm

The first step of the IPM is to transform the inequality constraints of the original problem (1) into equality constraints by introducing slack variables, which must be positive. This leads to:

$$\begin{aligned} \min \quad & f(x) \\ \text{s.t.} \quad & g(x) = 0 \\ & h(x) + s = 0 \\ & s \geq 0 \end{aligned} \tag{2}$$

where x and s vectors constitute the so-called primal variables.

The main idea behind IPMs is to introduce a logarithmic barrier allowing simple constraints such as $s \geq 0$ to be easily incorporated into the objective function:

$$\begin{aligned} \min \quad & f(x) - \mu \sum_{i=1}^{nh} \ln(s_i) \\ \text{s.t.} \quad & g(x) = 0 \\ & h(x) + s = 0 \end{aligned} \tag{3}$$

where μ is a scalar, called the barrier parameter, that decreases monotonically to zero.

The associated Lagrangian function, transforming the problem with equality constraints (3) into an unconstrained problem is:

$$L(v) = f(x) - \mu \sum_{i=1}^{nh} ln\, s_i + \lambda^T g(x) + \pi^T (h(x) + s) \tag{4}$$

where $v = [x, s, \lambda, \pi]^T$, and λ and π are Lagrange multipliers vectors, called dual variables.

To obtain the local optimum of the original problem (1), the Karush-Kuhn-Tucker (KTT) first-order optimality conditions must be satisfied. That is, the derivatives of the Lagrangian function with respect to the primal and dual variables must all be equal to zero:

$$F(v) = \begin{bmatrix} L_x^\mu \\ L_s^\mu \\ L_\lambda^\mu \\ L_\pi^\mu \end{bmatrix}^T = \begin{bmatrix} {f_x}^T + {g_x}^T \lambda + {h_x}^T \pi \\ [s]\pi - \mu e \\ g(x) \\ h(x) + s \end{bmatrix} = 0 \tag{5}$$

where f_x is the gradient of $f(x)$, g_x is the Jacobian of $g(x)$ and h_x is the Jacobian of $h(x)$. The matrix $[s]$ is a diagonal matrix formed by vector s, and e is an appropriately sized vector whose elements are equal to one.

In (5), third and fourth equation represents primal feasibility conditions, first equation represents dual feasibility conditions and second equation represents complementarity condition.

A. Primal dual algorithm (PD)

The PD algorithm can be summarized in the following steps:

Step 0. Initialize iterations counter $k = 0$, selecting $\mu_0 > 0$ initial point v_0 that meets the strict positivity conditions $(s_0, \pi_0) > 0$.

Step 1. Apply Newton's method to solve the KKT system ¡Error! No se encuentra el origen de la referencia., which yields the following linear system to be solved in each iteration k.

$$F_v(v_k)\Delta v_k = -F(v_k) \tag{6}$$

where Δv_k is the vector of Newton directions and $F_v(v_k)$ is the matrix of partial derivatives of $F(v_k)$, with the following structure:

$$F_v(v_k) = \begin{bmatrix} L_{xx}^\mu & 0 & {g_x}^T & {h_x}^T \\ 0 & [\pi] & 0 & [s] \\ g_x & 0 & 0 & 0 \\ h_x & I & 0 & 0 \end{bmatrix} \tag{7}$$

where

$$L_{xx}^\mu = f_{xx} + g_{xx}\lambda + h_{xx}\pi \tag{8}$$

The symmetric matrices f_{xx}, g_{xx}, and h_{xx} are the Hessian matrix of $f(x)$, $g(x)$, and $h(x)$, respectively.

Step 2. Update variables, as follows:

$$v_{k+1} = v_k + \alpha_{p,d} \Delta v_k \tag{9}$$

where α_p and $\alpha_d \in (0,1]$ are the step lengths of the primal and dual variables, respectively, defined as follows:

$$\begin{aligned} \alpha_{p_k} &= \min\left(1, \gamma \min_{\Delta s_k < 0} \frac{-s_k}{\Delta s_k}\right) \\ \alpha_{d_k} &= \min\left(1, \gamma \min_{\Delta \pi_k < 0} \frac{-\pi_k}{\Delta \pi_k}\right) \end{aligned} \tag{10}$$

with $\gamma \in (0,1)$ being a security factor to ensure strict positivity of slack variables and their respective dual variables.

Step 3. Verify convergence. A local optimum of the problem (1) is found and the optimization process is finished when it meets:

$$\begin{aligned} \max\{\max(h(x_k), \|g(x_k)\|_\infty)\} &\leq \varepsilon_1 \\ \frac{\left\|{f_x}_k + {\lambda_k}^T {g_x}_k + {\pi_k}^T {h_x}_k\right\|_\infty}{1 + \|x_k\|_2 + \|\lambda_k\|_2 + \|\pi_k\|_2} &\leq \varepsilon_1 \\ \frac{\rho_k}{1 + \|x_k\|_2} &\leq \varepsilon_2 \\ \frac{|f(x_k) - f(x_{k-1})|}{1 + |f(x_k)|} &\leq \varepsilon_2 \end{aligned} \tag{11}$$

where $\boldsymbol{\varepsilon_1}$ and $\boldsymbol{\varepsilon_2}$ are predetermined tolerances, and ρ_k is the residue of the complementarity conditions, known as complementarity gap, obtained by

$$\rho_k = {s_k}^T \pi_k \tag{12}$$

If convergence is not obtained, the barrier parameter is reduced based on the average decrease of the complementarity gap:

$$\mu_{k+1} = \sigma_k \frac{\rho_k}{nh} \tag{13}$$

where $\sigma_k \in (0,1)$ is called central parameter. A way to update σ_k is $\sigma_{k+1} = \max\{0.99\sigma_k, 0.1\}$. Finally, set $k = k + 1$ and go to step 1.

B. Predictor corrector algorithm (PC)

The main idea of the PC algorithm proposed in [2] and other higher-order algorithms [3, 4] is to improve the accuracy to which the Newton direction approximates the nonlinear KKT system equations.

At each iteration of the PC algorithm, the Newton direction is obtained by solving two systems of linear equations, respectively known as predictor and corrector steps. They involve the same single coefficient matrix with two different right-hand sides:

$$F_v(v_k)\Delta v^k = -\underbrace{\begin{bmatrix} L_x^{\mu^T} \\ [s]\pi \\ g(x) \\ h(x)+s \end{bmatrix}}_{RHS\ 1} + \underbrace{\begin{bmatrix} 0 \\ \mu e - [\Delta s]\Delta\pi \\ 0 \\ 0 \end{bmatrix}}_{RHS\ 2} \tag{14}$$

where

$$\Delta v^k = -\underbrace{\Delta v_{af}^k}_{predictor} + \underbrace{\Delta v_{cen}^k + \Delta v_{seg}^k}_{corrector} \tag{15}$$

The idea is to first compute the Newton direction Δv_{af}, called *affine-scaling* or predictor direction provided by the first right-h and side of (14),

$$F_v(v_k)\Delta v_{af}^k = -\begin{bmatrix} L_x^{\mu^T} \\ [s]\pi \\ g(x) \\ h(x)+s \end{bmatrix} \tag{16}$$

The *affine-scaling* direction Δv_{af} is then used to approximate the Δ-terms on the second right-hand side of (14) and to dynamically estimate the barrier parameter μ_{af}.

To estimate μ_{af}, step lengths $\alpha_{p_{af}}$ and $\alpha_{d_{af}}$ are obtained in the predictor direction by using (10). Afterwards, the complementarity gap is estimated by means of:

$$\rho_{af}^k = \left(s_k + \alpha_{p_{af}}^k \Delta s_{af}^k\right)^T \left(\pi_k + \alpha_{d_{af}}^k \Delta\pi_{af}^k\right) \tag{17}$$

Finally, the estimation of μ_{af}. is obtained from:

$$\mu_{af_k} = \sigma_{af_k} \frac{\rho_{af_k}}{nh} \tag{18}$$

where the central parameter is determined by:

$$\sigma_{af_k} = \min\left\{\left(\frac{\rho_{af}^k}{\rho_k}\right)^2, 0.2\right\} \tag{19}$$

The corrector direction Δv_{co} is obtained by solving the system (14) with the second right-hand side,

$$F_v(v_k)\Delta v_{co}^k = -\begin{bmatrix} 0 \\ [\Delta s_{af}]\Delta\pi_{af} - \mu_{af} e \\ 0 \\ 0 \end{bmatrix} \tag{20}$$

Finally, full Newton direction Δv is obtained by the sum of the predictor and corrector directions. The additional effort in the PC algorithm is in the extra linear system solution to compute the *affine-scaling* direction, and to estimate μ_{af}. This extra effort per iteration, results in a reduction in the number of iterations that, in general, translates into overall computational time savings.

C. Multiple Centrality Corrections algorithm (MCC)

The MCC algorithm [3] uses the predictor direction Δv_{af}^k obtained in the PC algorithm to evaluate one or more corrector terms Δv_{co}^k, in order to achieve two main goals: (i) improving the centrality of the next iteration, and (ii) increasing the step lengths. The motivation for the first goal is to increase the chances for a longer step to be taken in the following iteration. The argument for the second one is to obtain a faster reduction of the infeasibilities; overall, the aim is to accelerate the convergence rate.

The affine-scaling direction and its respective step lengths are calculated the same as the PC algorithm. To calculate the corrector direction, first the step lengths are enlarged by an increment δ:

$$\tilde{\alpha}_{p,d}^k = \min\left(\alpha_{p_{af},d_{af}}^k + \delta, 1\right) \tag{21}$$

According to [10], a trial point is defined as:

$$\tilde{v} = v^k + \tilde{\alpha}_{p,d}^k \Delta v_{af} \tag{22}$$

By definition the trial point violates the condition of strict positivity. Therefore, the corrector term Δv_{co} must compensate the negative components so that the trial point, $\tilde{v}$, returns to a neighborhood of the central path.

Then a vector $\tilde{w}$ is defined as the product of the complementarity conditions $[\tilde{s}]\tilde{\pi}$.

Next, we identify the components of $\tilde{w}$ that do not belong to the interval $\left(\beta_{min}\mu_{af}^k, \beta_{max}\mu_{af}^k\right)$, where β_{min} and β_{max} are given relative threshold values and $\beta_{min} = 1/\beta_{max}$. These components are called outliers complementarity products. The effort in the corrector steps is focused on correcting only outliers to improve the centrality of the trial point. To this end, the components of $\tilde{w}$ are projected on a hypercube $H = \left[\beta_{min}\mu_{af}^k, \beta_{max}\mu_{af}^k\right]^{2nh}$ to define the following target:

$$w = \begin{cases} \beta_{min}\mu_{af}^k, & if\ \tilde{w} < \beta_{min}\mu_{af}^k, \\ \beta_{max}\mu_{af}^k, & if\ \tilde{w} < \beta_{max}\mu_{af}^k \\ \tilde{w} & otherwise. \end{cases} \tag{23}$$

Then a corrector term is obtained as the solution of the following linear system:

$$F_v(v_k)\overbrace{\begin{bmatrix} \Delta x^{m_k} \\ \Delta s^{m_k} \\ \Delta\lambda^{m_k} \\ \Delta\pi^{m_k} \end{bmatrix}}^{\Delta v^{m_k}} = \begin{bmatrix} 0 \\ w - \tilde{w} \\ 0 \\ 0 \end{bmatrix} \tag{24}$$

Full Newton direction Δv is calculated as the sum of the predictor and corrector directions. A new step length in the direction Δv is determined, and new values for the primal and dual variables are computed.

The correcting process can be repeated a desired number of times, in which case the direction Δv becomes each time a new predictor, $\Delta v_{af} = \Delta v$.

According to [3], new centrality corrections can be implemented until the step lengths in the full Newton direction Δv do not increase sufficiently compared with the step lengths found earlier for the predictor direction. Following this procedure, we stop correcting if:

$$\alpha_p \geq \alpha_{p_{af}} + \phi\delta \;\; and \;\; \alpha_d \geq \alpha_{d_{af}} + \phi\delta \qquad (25)$$

where ϕ is the minimum acceptable increase of step length. Then, besides condition (25) it is necessary to limit the number of centrality corrections (M) per iteration.

In [10] the increase of step lengths δ is dynamically selected as follows:

$$\delta = \left(1 - min\left(\alpha_{p_{af}}, \alpha_{d_{af}}\right)\right)/M \qquad (26)$$

Additionally, δ should be not less than 0.1 neither greater than 0.2

D. Weigthed Multiple Centrality Corrections algorithm (WMCC)

The WMCC algorithm [4] is based on overcoming a drawback of the PC algorithm. Cartis in [14] shows that for certain starting points the corrector is always orders of magnitude larger than the predictor. Whilst the predictor points towards the optimum, the second-order corrector points away from it.

As the total direction is given by $\Delta v = \Delta v_{af} + \Delta v_{co}$, it is influenced almost exclusively by the corrector term, hence it is not accurate and we could get a wrong forward direction. As a result, the step lengths in the direction Δv are small.

To overcome this deficiency Colombo [4] proposed to reduce the influence of the corrector direction through a weighting coefficient, namely:

$$\Delta v = \Delta v_{af} + \omega \Delta v_{co} \qquad (27)$$

where the weighting coefficient $\omega \in (0,1]$.

A sequence of multiple centrality correctors can be generated, and for each of them we choose the optimal weight $\hat{\omega}$ which maximizes the step length in primal and dual spaces, that is:

$$\Delta v = \Delta v_{af} + \hat{\omega} \Delta v^{m_k} \qquad (28)$$

The total direction Δv becomes a predictor direction Δv_{af} for the next centrality corrector Δv^{m_k}. In [4] the optimal weight coefficient $\hat{\omega}$ in (28) is selected by a line search in the interval $[\omega_{min}, \omega_{max}] = [\alpha_p \alpha_d, 1]$, where α_p and α_d are the step lengths in the predictor direction Δv_{af}.

For this purpose, nine uniformly distributed points in the interval $[\alpha_p \alpha_d, 1]$ are selected in [4]. Then, for each of these points, the step lengths α_p^{ω} and α_d^{ω} are evaluated. When a larger step length α_p^{ω} or α_d^{ω} is obtained, the corresponding ω is stored as $\hat{\omega}_p$ or $\hat{\omega}_d$ respectively, allowing two different weightings for directions in the primal and dual spaces.

Step lengths, used to determine the trial point (22), are calculated by using

$$\tilde{\alpha}_{p,d} = \min\left(1.5\alpha_{p_{af}, d_{af}} + 0.3, 1\right) \qquad (29)$$

New centrality corrections are accepted if $\alpha_p \geq 1.01\alpha_{p_{af}}$, $\alpha_d \geq 1.01\alpha_{d_{af}}$, and $(m_k < M)$.

4. Numerical Experiments.

The IPM-based OPF formulations described above have been coded in Matlab R2011A and tested on a set of five power networks ranging in size from 14 to 3120 buses. Simulations are performed on Windows PC with a core i7 of 2.8 GHz processor, 8 GB of RAM and with the default parameters shown in Table 1. Solution times below include the whole solution process, except for data reading and dumping of results.

Table 1. Default parameters

Parameter	μ_0	σ_0	γ	τ	M
Value	1	0.2	0.99995	0.25	5
Parameter	β_{min}	β_{max}	ε_1	ε_2	ϕ
Value	0.1	10	1E-04	1E-06	0.1

The variables x are initialized from the solution of a power flow, the other variables are initialized as proposed in [9]. Table 2 shows the size of the power systems and their respective OPF problem.

Table 2. Sizes of power systems and OPF problems

System	n_b	n_g	n_l	n_x	n_{eq}	n_{iq}
bus14	14	5	20	38	29	88
bus118	118	54	186	344	237	824
bus300	300	69	411	738	601	1698
bus2383wp	2383	327	2896	5420	4898	11604
bus3120sp	3120	298	3693	6836	6373	14554

Table 3 shows the number of iterations and solution times for the systems analyzed. The biggest difference in the number of iterations arises between PD and PC IPMs, so the superiority of higher-order IPMs can be claimed. WMCC IPM converges in fewer iterations and computational time, and increases its performance for large systems.

Concerning the 3120-bus system, Figure 1 shows the step lengths (minimum of the primal and dual step length) for each iteration by different IPMs (for PD IPM only the step lengths of the first 19 iteration are showed for a total of 29 iterations). As seen in Figure 1, WMCC IPM usually takes longer step lengths than other IPMs, requiring consequently a smaller number of iterations to achieve convergence.

Table 3. Number of iterations and solution times (iter./sec.)

System	PD	PC	MCC	WMCC
bus14	12/0.028	6/0.016	6/0.017	6/0.018
bus118	14/0.099	9/0.068	9/0.071	8/0.068
bus300	16/0.22	10/0.146	9/0.14	9/0.145
bus2383wp	29/3.034	18/1.932	15/1.779	13/1.663
bus3120sp	29/4.573	19/2.68	16/2.379	12/2.012
Total	100/7.956	62/4.841	55/4.386	48/3.906

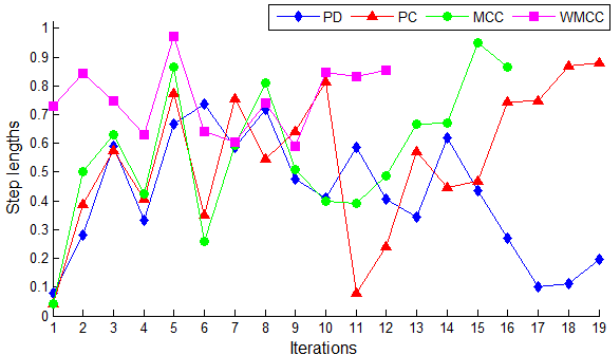

Figure 1. Step length by diferent IPMs

Table 4 reports the step lengths taken by the MCC IPM to solve the 2383 bus system. It can be seen that, in all 15 iterations at least one centrality correction is performed. Only in the 8th and 9th iterations are the maximum 5 centrality corrections performed. Note that performing one or more centrality corrections significantly improves step lengths, which leads to better convergence.

Figure 2 shows the weighting parameter of the corrector step for an iteration of the 3120 bus system by using WMCC IPM. The initial primal and dual step lengths are $\alpha_{p0} = 0.731$ and $\alpha_{d0} = 0.653$, and the search interval is [0.477,1]. The results of the line search gives $\hat{\omega}_p = 0.674$ associated to $\alpha_p^{\omega} = 0.809$, and $\hat{\omega}_d = 0.608$ associated to $\alpha_d^{\omega} = 0.729$, respectively. These step lengths are greater than those obtained initially without weighting, resulting in faster convergence.

Since the OPF formulations developed in this work use the complex matrix notation, initially proposed in [12] and used in MATPOWER program [13], a comparative analysis is performed between our WMCC IPM with the PD IPM of MATPOWER, called MIPS. The results, presented in Table 5, show that our WMCC IPM achieves a reduction of 64.44% in total number of iterations and 56.81% for total solution time.

5. Conclusiones.

In this paper we have presented and compared the performance of PD, PC, MCC and WMCC IPMs. These IPMs have been applied to solve the optimal power flow problem associated with generation cost function on test systems of different sizes. We have shown that the IPMs that use multiple centrality corrections, such as MCC and WMCC IPMs, achieve better performance, especially for large-scale OPF problems. We have also concluded that the WMCC IPM, being the most efficient of all implemented algorithms, clearly outperforms the PD IPM of MATPOWER.

Table 4. Step length using MCC IPM

Iteration		α_{af}	Number of centrality corrections (m)				
			1	2	3	4	5
1	Primal	0.52	0.56	0.532			
	Dual	0.277	0.35	0.489			
2	Primal	0.227	0.304	0.331	0.367		
	Dual	0.209	0.294	0.358	0.363		
3	Primal	0.41	0.56	0.684	1	1	
	Dual	0.498	0.584	0.687	0.744	0.803	
4	Primal	1	1				
	Dual	0.87	0.871				
5	Primal	0.686	0.72				
	Dual	0.817	0.819				
6	Primal	0.37	0.401				
	Dual	0.834	0.005				
7	Primal	0.711	0.728	0.747	0.767		
	Dual	0.823	0.854	0.89	0.895		
8	Primal	0.336	0.366	0.405	0.456	0.5	0.574
	Dual	0.518	0.568	0.608	0.645	0.673	0.615
9	Primal	0.393	0.407	0.428	0.459	0.501	0.543
	Dual	0.795	0.846	0.916	0.951	0.964	0.382
10	Primal	0.788	0.791				
	Dual	0.44	0.556				
11	Primal	0.52	0.536	0.641			
	Dual	0.631	0.749	0.712			
12	Primal	0.42	0.445				
	Dual	0.985	0.99				
13	Primal	0.831	0.877	0.939	0.97	0.971	
	Dual	0.682	0.7	0.728	0.763	0.803	
14	Primal	1	1				
	Dual	0.931	0.999				
15	Primal	0.897	0.997				
	Dual	0.998	1				

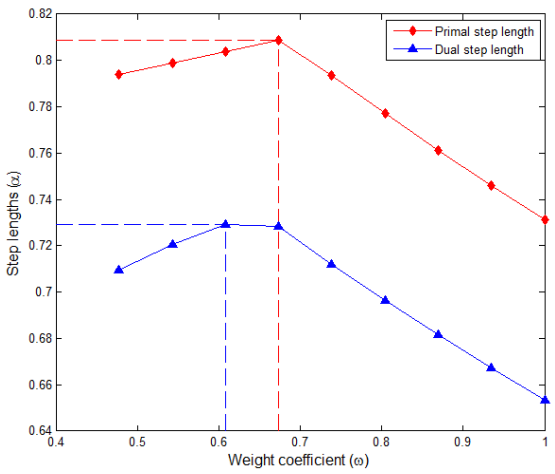

Figure 2. Weighted corrector direction using WMCC

Table 5. Comparison between MATPOWER MIPS results and the implemented WMCC algorithm.

System	MATPOWER		WMCC	
	Iter.	t(sec.)	Iter.(%)	t(%)
bus14	15	0.062	60.00%	71.06%
bus118	19	0.155	57.89%	56.20%
bus300	26	0.358	65.38%	59.37%
bus2383	33	3.157	60.61%	47.32%
bus3120	42	5.313	71.43%	62.13%
Total	135	9.045	64.44%	56.81%

References

[1] N. Megiddo, "Pathways to the optimal set in linear programming," in *Progress in Mathematical Programming Interior-point and related methods*, ed: Springer-Verlag New York, Inc., 1988, pp. 131-158.

[2] S. Mehrotra, "On the Implementation of a Primal-Dual Interior Point Method," *SIAM Journal on Optimization*, vol. 2, pp. 575-27, 1992.

[3] J. Gondzio, "Multiple centrality corrections in a primal-dual method for linear programming," *Computational Optimization and Applications*, vol. 6, pp. 137-156, 1996.

[4] M. Colombo and J. Gondzio, "Further development of multiple centrality correctors for interior point methods," *Comput. Optim. Appl.*, vol. 41, pp. 277-305, 2008.

[5] S. Granville, "Optimal reactive dispatch through interior point methods," *Power Systems, IEEE Transactions on*, vol. 9, pp. 136-146, 1994.

[6] J. L. Martínez, A. Gómez, and V. H. Quintana, "Reactive-Power Optimization by Interior-Point Methods: Implementation Issues," in *Power System Computation Conference. (PSCC)*, Dresde, 1996, pp. 844-850

[7] J. L. Martínez, A. Gómez, and V. H. Quintana, "Transmission power loss reduction by interior-point methods: implementation issues and practical experience," *Generation, Transmission and Distribution, IEE Proceedings-*, vol. 152, pp. 90-98, 2005.

[8] Y. Wu, A. S. Debs, and R. E. Marsten, "A direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flows," *Power Systems, IEEE Transactions on*, vol. 9, pp. 876-883, 1994.

[9] G. L. Torres and V. H. Quintana, "An interior-point method for nonlinear optimal power flow using voltage rectangular coordinates," *Power Systems, IEEE Transactions on*, vol. 13, pp. 1211-1218, 1998.

[10] G. L. Torres and V. H. Quintana, "On a nonlinear multiple-centrality-corrections interior-point method for optimal power flow," *Power Systems, IEEE Transactions on*, vol. 16, pp. 222-228, 2001.

[11] H. Zhiguang and J. Quanyuan, "Nonlinear weighted multiple centrality corrections interior point method for optimal power flow," in *Sustainable Power Generation and Supply, 2009. SUPERGEN '09. International Conference on*, 2009, pp. 1-6.

[12] R. D. Zimmerman, "AC Power Flows, Generalized OPF Costs and their Derivatives using Complex Matrix Notation," *Matpower Technical Note 2*, 2011.

[13] R. D. Zimmerman, C. E. Murillo-Sanchez, and R. J. Thomas, "MATPOWER: Steady-State Operations, Planning, and Analysis Tools for Power Systems Research and Education," *Power Systems, IEEE Transactions on*, vol. 26, pp. 12-19, 2011.

[14] C. Cartis, "Some disadvantages of a Mehrotra-type primal-dual corrector interior point algorithm for linear programming," *Appl. Numer. Math.*, vol. 59, pp. 1110-1119, 2009.