Available online at www.sciencedirect.com

ScienceDirect

Electrical Power and Energy Systems 30 (2008) 383–392

ELECTRICAL POWER & ENERGY SYSTEMS

www.elsevier.com/locate/ijepes

Semidefinite programming for optimal power flow problems ☆

Xiaoqing Bai [a,*], Hua Wei [a], Katsuki Fujisawa [b], Yong Wang [c]

[a] College of Electric Engineering, Guangxi University, Guangxi, PR China
[b] Department of Mathematical Sciences, Tokyo Denki University, Saitama, Japan
[c] Department of Mathematics and Statistics, University of North Carolina at Charlotte, Charlotte, NC, United States

Received 29 April 2007; received in revised form 13 December 2007; accepted 18 December 2007

Abstract

This paper presents a new solution using the semidefinite programming (SDP) technique to solve the optimal power flow problems (OPF). The proposed method involves reformulating the OPF problems into a SDP model and developing an algorithm of interior point method (IPM) for SDP. That is said, OPF in a nonlinear programming (NP) model, which is a nonconvex problem, has been accurately transformed into a SDP model which is a convex problem. Based on SDP, the OPF problem can be solved by primal–dual interior point algorithms which possess superlinear convergence. The proposed method has been tested with four kinds of objective functions of OPF. Extensive numerical simulations on test systems with sizes ranging from 4 to 300 buses have shown that this method is promising for OPF problems due to its robustness.

Keywords: Optimal power flow; Interior point method; Semidefinite programming

1. Introduction

Since the early 60s of the last century, optimal power flow (OPF) as a powerful tool for power system optimization problems has attracted many researchers over the world [1,2]. Numerous algorithms have been developed in this area based on linear programming (LP), quadratic programming, Newton's method, nonlinear programming (NLP) and decomposition method [3]. Recently, NLP-based algorithms using interior point methods (IPM) have also been applied to OPF problems successfully [4–7]. However, as power systems are getting more complex, the OPF problems turn to be more difficult to handle. Although the theory on IPM for NLP has been well developed, many issues remain open when building the links between the modeling and the associated algorithms. First of all, the NLP-based OPF has the convergence problem due to its nonconvex nature. Moreover, in order to use IPM for NLP, the Jacobian matrices (the first-order partial derivatives) and the Hessian matrices (the second-order partial derivatives) have to be derived for each specific problem. As a result, it is not convenient to develop a general and uniform software solution for the NLP problems using IPM.

The semidefinite programming (SDP) [8,9] has been one of the most active fields in numerical optimization for over a decade. Many well-known algorithms with uniform frameworks have been exploited [10]. It has been proven that the SDP is convex and the primal–dual interior point algorithms for SDP may possess superlinear convergence theoretically [11]. Moreover, the major advantage for the SDP-based IPM is the avoidance of deriving and computing the Jacobian matrices and the Hessian matrices for each particular problem.

☆ This work was supported in part by National Natural Science Foundation of China under Grant 50467001, Guangxi Science Foundation under Grant 0448025, Doctoral Fund of Ministry of Education of China under Grand 20060593002, and Guangxi Education Department under Grant GJK No. 200420.
* Corresponding author. Tel.: +86 771 3232264.
E-mail addresses: baixq@gxu.edu.cn (X. Bai), weihua@gxu.edu.cn (H. Wei), fujisawa@rnc.r.dendai.ac.jp (K. Fujisawa), ywang73@uncc.edu (Y. Wang).

0142-0615/$ - see front matter © 2008 Published by Elsevier Ltd.
doi:10.1016/j.ijepes.2007.12.003

Different practical problems have been solved successfully in various domains, such as control and system theory [12,13], signal processing and communications [14–16], and combinatorial optimization [17–19]. In [20,21], the SDP has been used to solve power dispatch problems, which are typical constrained economic dispatching problems (CED). To our knowledge, these two paper were pioneering in applying the SDP technique to power systems. However, the works in [20,21] did not involve power flow equations and the bus voltage constraints.

It is significant and challenging to extend the SDP to solve OPF problems which are know for their inherent complexity and practicality. Motivations of this study stem from the following three aspects:

1. The classical OPF problem is a nonconvex NLP [22], which solution is more complicated than the CED problems mentioned earlier. On the other hand, the SDP belongs to convex optimization [23], and can guarantee global optimal solution using IPM. Therefore, it is worth to study how to properly reformulate the classical OPF to a SDP model, due to many advantages of the SDP technique including its convexity and uniform algorithm implemental framework of software suite.
2. Once an OPF problem is reformulated to a SDP one using the quadratic model [24] in the case of the rectangular form, the resulting SDP model should be convex, and the solution quality can be guaranteed by using IPM for the SDP [23]. In detail, by applying $X = x^{\mathrm{T}}x$ where x is a row vector and trace operator of matrices which will be discussed in Section 3, the nonlinear quadratic items of the OPF will be replaced with the relevant elements of the variable matrix X in SDP.
3. Mature techniques such as IPM for LP are available to solve the SDP problems. Furthermore, the solution techniques developed for LP which actually is a special case of SDP are also applicable to SDP [23]. Therefore, IPM for LP, which can be used to solve convex optimization problems in polynomial time, can be implemented for SDP [25]. Moreover, IPM enhancements for LP can also be used in SDP, which renders IPM for SDP as efficient as that for LP theoretically [26].

2. Formulation of OPF problem

The OPF problem is a large-scale nonlinear optimization problem. It can be formulated in polar, rectangular, or mixture of polar and rectangular forms. In this study, the rectangular version of the OPF problem is adopted to take the advantages that the power flow equations are quadratic polynomials without trigonometric functions, which can then yield the SDP models straightforward. The objective functions to be minimized are chosen out of active or reactive power loss of transmission lines, fuel cost, and total system active power loss. Therefore, the OPF problem can be formulated as follows:

$$\text{minimize}\begin{cases} P_{\text{Loss2}} = -\sum\limits_{i\in S_B}\sum\limits_{j\in S_B} G_{ij}[(f_i - f_j)^2 + (e_i - e_j)^2] \\ Q_{\text{Loss}} = \sum\limits_{i\in S_B}\sum\limits_{j\in S_B} B_{ij}[(f_i - f_j)^2 + (e_i - e_j)^2] \\ F_{\text{Cost}} = \sum\limits_{i\in S_G}(a_{fi} + a_{li}P_{Gi} + a_{qi}P_{Gi}^2) \\ P_{\text{Loss1}} = \sum\limits_{i\in S_G} P_{Gi} \end{cases} \tag{1}$$

subject to:

1. Power flow equations:

$$\begin{cases} P_{Gi} - \sum\limits_{j\in S_B}[e_i(e_jG_{ij} - f_jB_{ij}) + f_i(f_jG_{ij} + e_jB_{ij})] = P_{Di} \\ Q_{Ri} - \sum\limits_{j\in S_B}[f_i(e_jG_{ij} - f_jB_{ij}) - e_i(f_jG_{ij} + e_jB_{ij})] = Q_{Di} \end{cases}, \quad i \in S_B \tag{2}$$

2. Constraint of reference bus:

$$e_s = 1.05; \quad f_s = 0 \tag{3}$$

3. Limits of active and reactive power:

$$\underline{P}_{Gi} \leqslant P_{Gi} \leqslant \overline{P}_{Gi}, \quad i \in S_G \tag{4}$$

$$\underline{Q}_{Ri} \leqslant Q_{Ri} \leqslant \overline{Q}_{Ri}, \quad i \in S_R \tag{5}$$

4. Limits of voltage at each bus:

$$\underline{V}_i^2 \leqslant (e_i^2 + f_i^2) \leqslant \overline{V}_i^2, \quad i \in S_B \tag{6}$$

where

a_{fi}, a_{li}, a_{qi}: cost coefficients of thermal plant, i, respectively,
e_i, f_i: real and imaginary part of voltage at bus i,
$\dot{V}_i$: voltage at bus i,
G_{ij}, B_{ij}: real and imaginary part of transfer admittance between buses i and j,
P_{Gi}, Q_{Ri}: dispatchable active and reactive power at bus i,
P_{Di}, Q_{Di}: active and reactive power demand at bus i,
S_B, S_G, S_R: set of buses (n_B), thermal plants (n_G) and reactive power sources (n_B), respectively,
s: identification serial number of reference bus in the system,
$(\underline{\circ})$, $(\overline{\circ})$: lower and upper limits of variables or functions.

The set of (1)–(6) is known as the classical OPF problem.

3. Semidefinite programming by IPM

The semidefinite programming [9] is concerned with choosing a positive semidefinite matrix to optimize a linear function which is subject to linear constraints. In other words, the well-known linear programming problem is generalized by replacing the vector of variables with a symmetric matrix and the nonnegative constraints with a positive semidefinite constraint. This generalization nevertheless inherits several important properties from its vector counterpart: it is convex, has a rich duality theory (although not as strong as the one of linear programming), and theoretically admits efficient solution procedures based on iterating interior points.

3.1. Semidefinite programming

A semidefinite programming problem can be in different forms [9]. In this paper, we only consider the following standard form [8], namely, the primal one and its dual:

$$\text{Primal}: \begin{cases} \min A_0 \bullet X \\ \text{s.t. } A_i \bullet X = b_i, \quad i = 1, \dots, m \\ \quad X \succ 0 \end{cases}$$

$$\text{Dual}: \begin{cases} \max b^{\mathrm{T}} y \\ \text{s.t. } A_0 - \sum_{i=1}^{m} y_i A_i \succ 0 \end{cases} \tag{7}$$

where

1. $(y, b) \in \mathbb{R}^m, X \in \mathbb{R}^{n\times n}$, and $A_i \in \mathbb{R}^{n\times n}$, $i = 0, \dots, m$;
2. symbol $\bullet$ is the trace operator. For example, $A_1 \bullet X = \sum_{i=1}^{n}\sum_{j=1}^{n} A_{1ij} X_{ji}$;
3. $X \succ 0$ and $A_0 - \sum_{i=1}^{m} y_i A_i \succ 0$ mean that X and $A_0 - \sum_{i=1}^{m} y_i A_i$ are positive semidefinite, respectively.

3.2. IPM for SDP

All the improvements developed for IPM as applied to linear programming are also useful when they are extended to SDP. For example, the primal–dual interior point method (PDIPM) is used to solve the SDP problem successfully [8,27]. More clearly, PDIPM generates a sequence of iterations which approximate the central path and converge to the primal and dual solutions [28].

3.2.1. Primal–dual interior point algorithms

Let $\ln\det X$ be the logarithmic barrier for the determinant of the matrix X. Replacing the primal problem in (7) with the relaxed barrier version [8] yields:

$$\begin{cases} \min A_0 \bullet X - \mu \ln\det X \\ \text{s.t. } A_i \bullet X = b_i, \quad i = 1, \dots, m \end{cases} \tag{8}$$

The Lagrangian function is then given by:

$$L(X, y) \equiv A_0 \bullet X - \mu \ln\det X - \sum_{i=1}^{m} y_i (b_i - A_i \bullet X)$$

Applying the first-order Karush–Kuhn–Tucker (KKT) optimality conditions leads to:

$$\begin{cases} \nabla L_X^\mu = A_0 - \sum_{i=1}^{m} A_i y_i(\mu) - \mu X(\mu)^{-1} = 0 \\ \nabla L_{yi}^\mu = b_i - A_i \bullet X(\mu) = 0, i = 1, \dots, m \end{cases}$$

Now introducing a symmetric matrix Z yields the following nonlinear system:

$$\begin{cases} A_i \bullet X(\mu) = b_i, \quad i = 1, \dots, m \\ \sum_{i=1}^{m} y_i(\mu) A_i + Z(\mu) = A_0 \\ X(\mu) Z(\mu) = \mu I \end{cases} \tag{9}$$

where the last equation is equivalent to $Z(\mu) = \mu X(\mu)^{-1}$. Then, the same approach above is applied to the dual:

$$\begin{cases} \max b^{\mathrm{T}} y - \mu \ln\det Z \\ \text{s.t. } \sum_{i=1}^{m} y_i A_i + Z = A_0 \end{cases} \tag{10}$$

The dual Lagrangian now is (with the Lagrange multiplier being a symmetric matrix):

$$L(X, y, Z) = b^{\mathrm{T}} y - \mu \ln\det Z - X \bullet \left(A_0 - \sum_{i=1}^{m} y_i A_i - Z\right)$$

Applying the KKT optimality conditions, we get:

$$\begin{cases} \nabla L_X^\mu = A_0 - \sum_{i=1}^{m} A_i y_i(\mu) - Z(\mu) = 0 \\ \nabla L_{yi}^\mu = b_i - A_i \bullet X(\mu) = 0, \quad i = 1, \dots, m \\ \nabla L_Z^\mu = v Z^{-1}(\mu) - X(\mu) = 0 \end{cases}$$

which is essentially the same as the one obtained from the primal problem.

3.2.2. Central path

In PDIPM [29], the central path (μ) is composed of points defined by system (9):

$$\Upsilon(\mu) \equiv \{X(\mu), y(\mu), Z(\mu) \in R^{n\times n} \times R^m \times R^{n\times n} : \mu > 0\}$$

The central path plays a crucial role in computing an optimal solution. For any complementary gap $\mu > 0$, there exists a unique $X(\mu), y(\mu), Z(\mu) \in R^{n\times n} \times R^m \times R^{n\times n}$ satisfying (10). Thus the central path (μ) structures a smooth curve in the space of $X(\mu), y(\mu), Z(\mu) \in R^{n\times n} \times R^m \times R^{n\times n}$. Moreover, it is obvious that $X(\mu)\bullet Z(\mu) = n\mu$ can be held since $X(\mu)Z(\mu) = \mu I$. Therefore, $\mu = X(\mu)\bullet Z(\mu)/n$. In addition, the central path (μ) converges to an optimal solution of the SDP, i.e., $(X(\mu), y(\mu), Z(\mu))$ converges to an optimal solution of the SDP as $\mu \to 0$. Thus, the PDIPM traces path numerically as μ decreases toward 0.

3.2.3. Search direction

A search direction $(\Delta X, \Delta y, \Delta Z)$ [30] is taken so that $(X^K + \Delta X, y^K + \Delta y, Z^K + \Delta Z)$ can coincide with the target point $(X(\beta\mu^k), y(\beta\mu^k), Z(\beta\ \mu^k))$ on the central path with $\mu = \beta\mu^k$, which leads to the following system by applying Newton's method to (9):

$$\begin{cases} A_i \bullet (X^k + \Delta X) = b_i, \quad i = 1, \dots, m \\ \sum_{i=1}^{m} (y_i^k + \Delta y) A_i + (Z^k + \Delta Z) = A_0 \\ (X^k + \Delta X)(Z^k + \Delta Z) = \beta\mu^k I \end{cases} \tag{11}$$

System (11) is almost a linear system except the term $\Delta X \Delta Z$ in the last equality. Neglecting the nonlinear term and introducing an auxiliary matrix $\Delta\widetilde{Z}$, the system of nonlinear equations above can be reduced into the following system of linear equations:

$$\begin{cases} B\Delta y = r \\ \Delta X = P + \sum_{i=1}^{m} A_i \Delta y_i \\ \Delta\widetilde{Z} = (X^k)^{-1}(R - Z^k \Delta X) \\ \Delta Z = (\Delta\widetilde{Z} + \Delta\widetilde{Z}^{\mathrm{T}})/2 \end{cases} \tag{12}$$

where

$$B_{ij} = [(X^k)^{-1} A_i Z^k] \bullet A_j, \quad i,j = 1,\ldots,m$$
$$r_i = -d_i + A_i \bullet [(X^k)^{-1}(R - PZ^k)], \quad i = 1,\ldots,m$$
$$P = \sum_{i=1}^{m} A_i y_i^k - A_0 - X^k$$
$$d_i = c_i - A_i \bullet Z^k, \quad i = 1,2,\ldots,m$$
$$R = \beta\mu^k I - X^k Z^k$$

3.2.4. The framework of the PDIPM algorithm for SDP

The general framework of the PDIPM algorithm for SDP is described as follows:

Initialization: set the tolerance $\varepsilon = 10^{-5}$, the centering parameter $\beta = 0.2$ and the step length reduction factor $\gamma = 0.85$. Note that they actually can be held by $0 < \beta < 1$ and $0 < \gamma < 1$. The more detailed discussion about how to properly choose β and γ is presented in Part E of Section 4. Choose the initial point (X^0, y^0, Z^0) with $X^0 \succ 0, Z^0 \succ 0$.

Begin $k = 0, 1,\ldots;$

Step 1: $\mu^k = X^k \bullet Z^k / n$. If $\mu^k < \varepsilon$ and (X^k, y^k, Z^k) approximately satisfies feasibility, (X^k, y^k, Z^k) as an approximate optimal solution is printed out and stop;

Step 2: A search direction $(\Delta X, \Delta y, \Delta Z)$ is computed toward a target point $(X(\beta\mu^k), y(\beta\mu^k), Z(\beta\mu^k))$ on the central path (μ) with $\mu = \beta\mu^k$ by solving (12).

Step 3: The step lengths α_p and α_d are computed such that $X^k + \alpha_p \Delta X$ and $Z^k + \alpha_d \Delta Z$ to X^{k+1} and Z^{k+1} remain in the interior of the cone of positive definite matrices. Let L be the lower triangular matrix from the Cholesky factorization of $X^k = LL^{\mathrm{T}}$ and let $P\Lambda P^{\mathrm{T}}$ be the eigenvalue decomposition of $L^{-1}\Delta\ XL^{-\mathrm{T}}$. Then

$$\alpha_p = \gamma \min\{1, \bar{\alpha}_p\} \quad 0 < \gamma < 1 \tag{13}$$

where:

$$\bar{\alpha}_p = \begin{cases} -1/\lambda_{\min}, & \text{if } \lambda_{\min} < 0 \\ +\infty, & \text{otherwise} \end{cases}$$

Here $\lambda_{\min}$ is the minimum diagonal value of Λ. The set lengths α_d can be computed in the same way.

Step 4: Update the current point by

$$(X^{k+1}, y^{k+1}, Z^{k+1}) = (X^k + \alpha_p \Delta X, y^k + \alpha_d \Delta y, Z^k + \alpha_d \Delta Z) \tag{14}$$

End

4. Implementation of SOPF

The solution using the SDP technique to solve the OPF problem is called SOPF. In principle, the OPF problem can be reformulated into SDP shown in (7) using its quadratic style, and then solved by PDIPM. Afterward, the solution of the classical OPF can be mapped from the solution of SDP for OPF in a straightforward way. More detailed discussions about SOPF are presented below.

4.1. Process of SOPF

Once the relationship between the classical OPF problem and OPF in SDP style is confirmed by the analysis in the later two subsections, the left task is as follows:

Step 1: Set data to appropriate places in A which are coefficient matrices in (31) according to the analysis in Section 4.1.1 and 4.1.2.

Step 2: Invoke IPM solver for SDP as presented in Section 3.

Step 3: Map solutions to the classical OPF from the result of Step 2 to be described in Section 4.1.3.

4.1.1. OPF formulated as quadratic problem

For the classical OPF problem shown in (1)–(6), the object functions (F_{Cost}, P_{Loss2}, and Q_{Loss}) are quadratic, and (P_{Loss1}) is linear, respectively; all the constraints are quadratic or linear.

Let $d_{Gi} (i \in S_G)$ and d_{Ri} $(i \in S_R)$ be auxiliary variables which are always equal to one, i.e., $d_{Gi} = 1$ and $d_{Ri} = 1$. Obviously, there exists $d_{Gi}^2 = 1$ and $d_{Ri}^2 = 1$ since $d_{Gi} = 1$ and $d_{Ri} = 1$. On the other hand, also exists $P_{Gi} = P_{Gi} d_{Gi}$ and $Q_{Ri} = Q_{Ri} d_{Ri}$. Therefore, each linear term about P_{Gi} and Q_{Ri} in the classical OPF problem can be reformulated into the quadratic terms about P_{Gi} nd Q_{Ri} with $P_{Gi} d_{Gi}$ and $Q_{Ri} d_{Ri}$, respectively.

Meanwhile, by introducing a slack variable vector $[u_G, l_G, u_R, l_R, u_B, l_B] \in \mathbb{R}^{2n_G + 2n_R + 2n_B}$, inequality constraints (4)–(6) can be transformed into the equality constraints as $(u_G^2, l_G^2, u_R^2, l_R^2, u_B^2, l_B^2) \geqslant 0$. Thus, the classical OPF (1)–(6) can be equivalently formulated as a quadratic-objective and quadratic-constraint problem (15)–(28) as follows:

$$\text{minimize}\begin{cases} P_{\text{Loss2}} = -\sum_{i=1}^{n_B}\left[2(f_i^2+e_i^2)\sum_{j=1}^{n_B}(G_{ij}-G_{ii}) + 4\sum_{j=i+1}^{n_B}(f_if_j+e_ie_j)G_{ij}\right] \\ Q_{\text{Loss}} = \sum_{i=1}^{n_B}\left[2(f_i^2+e_i^2)\sum_{j=1}^{n_B}(B_{ij}-B_{ii}) - 4\sum_{j=i+1}^{n_B}(f_if_j+e_ie_j)B_{ij}\right] \\ F_{\text{Cost}} = \sum_{i\in S_G}(a_{fi}d_{Gi}^2 + a_{li}P_{Gi}d_{Gi} + a_{qi}P_{Gi}^2) \\ P_{\text{Loss1}} = \sum_{i\in S_G} P_{Gi}d_{Gi} \end{cases} \tag{15}$$

subject to:

1. Power flow equations:

$$\sum\nolimits_{j=1}^{n_B}(-e_ie_jG_{ij}+e_if_jB_{ij}-f_if_jG_{ij}-f_ie_jB_{ij}) = P_{Di}, \quad (i\in S_B/S_G) \tag{16}$$

$$\sum\nolimits_{j=1}^{n_B}(-f_ie_jG_{ij}+f_if_jB_{ij}+e_if_jG_{ij}+e_ie_jB_{ij}) = Q_{Di}, \quad (i\in S_B/S_R) \tag{17}$$

$$P_{Gi}d_{Gi}+\sum_{j=1}^{n_B}(-e_ie_jG_{ij}+e_if_jB_{ij}-f_if_jG_{ij}-f_ie_jB_{ij}) = P_{Di}, \quad (i\in S_G) \tag{18}$$

$$Q_{Ri}d_{Ri}+\sum_{j=1}^{n_B}(-f_ie_jG_{ij}+f_if_jB_{ij}+e_if_jG_{ij}+e_ie_jB_{ij}) = Q_{Di}, \quad (i\in S_R) \tag{19}$$

2. Constraint of reference bus:

$$f_s^2 = 0 \tag{20}$$
$$e_s^2 = 1.05^2 \tag{21}$$

3. Limits of active and reactive power:

$$P_{Gi}d_{Gi}+u_{Gi}^2 = \overline{P}_{Gi}, \quad i\in S_G \tag{22}$$
$$P_{Gi}d_{Gi}-l_{Gi}^2 = \underline{P}_{Gi}, \quad i\in S_G \tag{23}$$
$$Q_{Ri}d_{Ri}+u_{Ri}^2 = \overline{Q}_{Ri}, \quad i\in S_R \tag{24}$$
$$Q_{Ri}d_{Ri}-l_{Ri}^2 = \underline{Q}_{Ri}, \quad i\in S_R \tag{25}$$

4. Limits of amplitude at each bus:

$$e_i^2+f_i^2+u_{Bi}^2 = \overline{V}_i^2, \quad i\in S_B \tag{26}$$
$$e_i^2+f_i^2-l_{Bi}^2 = \underline{V}_i^2, \quad i\in S_B \tag{27}$$

5. Auxiliary variables

$$d_{Gi}^2 = 1, \quad i\in S_G$$
$$d_{Ri}^2 = 1, \quad i\in S_R \tag{28}$$

4.1.2. The OPF in quadratic form reformulated as SDP

Now the classical OPF problem has been formulated as a quadratic problem (15)–(28) which can be transformed into a SDP problem of form (7). For simplicity, TEST-4, a simple power system shown in Fig. 1 is used to explain the data matrix structures, where the generators G_1 and G_2 at the buses 1 and 4, respectively, are dispatchable

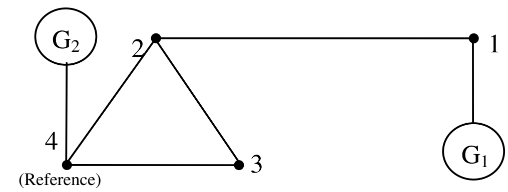

Fig. 1. TEST-4: a simple power system.

sources, the buses 2 and 3 have scheduled active and reactive power, and the bus 4 is the reference bus.

All the variables in the problem (15)–(28) for TEST-4 are grouped into a single vector $x = [x_1, x_2, x_3, x_4, x_5, x_6]$, where

1. Group of active resources:
   $x_1 = [P_{G1} \quad d_{G1} \quad P_{G2} \quad d_{G2}]$
2. Group of slack variables for active resources:
   $x_2 = [u_{G1} \quad l_{G1} \quad u_{G2} \quad l_{G2}]$
3. Group of reactive resources:
   $x_3 = [Q_{R1} \quad d_{R1} \quad Q_{R2} \quad d_{R2}]$
4. Group of slack variables for reactive resources:
   $x_4 = [u_{R1} \quad l_{R1} \quad u_{R2} \quad l_{R2}]$
5. Group of buses:
   $x_5 = [e_1 \quad f_1 \quad e_2 \quad f_2 \quad e_3 \quad f_3 \quad e_4 \quad f_4]$
6. Group of slack variables for buses:
   $x_6 = [u_{B1} \quad l_{B1} \quad u_{B2} \quad l_{B2} \quad u_{B3} \quad l_{B3} \quad u_{B4} \quad l_{B4}]$.

Afterward, the SDP variable can be defined by:

$$X = x^{\mathrm{T}}x \tag{29}$$

Obviously, the matrix $X \in \mathbb{R}^{n\times n}$ is positive definite or semidefinite [31], i.e., $X \succ 0$. And in this example, it has dimension $n = 2\times 2\times n_G + 2\times 2\times n_R + 2\times 2\times n_B = 32$.

The matrix X is shown in (30). For simplicity, some elements in X are replaced with ellipses, indicating that the relevant coefficients of those elements are always zeroes. Therefore, the matrix X can be treated as a block-diagonal symmetric matrix. This novel block-diagonal symmetric matrix variable X can take advantage of the sparsity technique of SDP [32], so that the computing efficiency can be improved to some extent.

$$X = \begin{bmatrix} X_1 & \cdots & & \cdots & & \cdots \\ \vdots & X_2 & \ddots & & \ddots & \\ & \ddots & X_3 & \ddots & & \vdots \\ \vdots & & \ddots & X_4 & \ddots & \\ & \ddots & & \ddots & X_5 & \vdots \\ \cdots & & \cdots & & \ddots & X_6 \end{bmatrix}_{(32\times 32)} \tag{30}$$

where

$$X_1=\begin{bmatrix} P_{G1}^2 & P_{G1}d_{G1} & P_{G1}P_{G2} & P_{G1}d_{G2} \\ P_{G1}d_{G1} & d_{G1}^2 & P_{G2}d_{G1} & d_{G1}d_{G2} \\ P_{G1}P_{G2} & P_{G2}d_{G1} & P_{G2}^2 & P_{G2}d_{G2} \\ P_{G1}d_{G2} & d_{G1}d_{G2} & P_{G2}d_{G2} & d_{G2}^2 \end{bmatrix}_{(4\times4)}$$

$$X_2=\begin{bmatrix} u_{G1}^2 & u_{G1}l_{G1} & u_{G1}u_{G2} & u_{G1}l_{G2} \\ u_{G1}l_{G1} & l_{G1}^2 & l_{G1}u_{G2} & l_{G2}l_{G1} \\ u_{G1}u_{G2} & l_{G1}u_{G2} & u_{G2}^2 & l_{G2}u_{G2} \\ u_{G1}l_{G2} & l_{G2}l_{G1} & l_{G2}u_{G2} & l_{G2}^2 \end{bmatrix}_{(4\times4)}$$

$$X_3=\begin{bmatrix} Q_{R1}^2 & Q_{R1}d_{R1} & Q_{R1}Q_{R2} & Q_{R1}d_{R2} \\ Q_{R1}d_{R1} & d_{R1}^2 & Q_{R2}d_{R1} & d_{R1}d_{R2} \\ Q_{R1}P_{R2} & Q_{R2}d_{R1} & Q_{R2}^2 & Q_{R2}d_{R2} \\ Q_{R1}d_{R2} & d_{R1}d_{R2} & Q_{R2}d_{R2} & d_{R2}^2 \end{bmatrix}_{(4\times4)}$$

$$X_4=\begin{bmatrix} u_{R1}^2 & u_{R1}l_{R1} & u_{R1}u_{R2} & u_{R1}l_{R2} \\ u_{R1}l_{R1} & l_{R1}^2 & l_{R1}u_{R2} & l_{R2}l_{R1} \\ u_{R1}u_{R2} & l_{R1}u_{R2} & u_{R2}^2 & l_{R2}u_{R2} \\ u_{R1}l_{R2} & l_{R2}l_{R1} & l_{R2}u_{R2} & l_{R2}^2 \end{bmatrix}_{(4\times4)}$$

$$X_5=\begin{bmatrix} e_1^2 & e_1f_1 & e_2e_1 & f_2e_1 & e_3e_1 & f_3e_1 & e_4e_1 & f_4e_1 \\ e_1f_1 & f_1^2 & e_2f_1 & f_2f_1 & e_3f_1 & f_3f_1 & e_4f_1 & f_4f_1 \\ e_1e_2 & f_1e_2 & e_2^2 & f_2e_2 & e_3e_2 & f_3e_2 & e_4e_2 & f_4e_2 \\ e_1f_2 & f_1f_2 & e_2f_2 & f_2^2 & e_3f_2 & f_3f_2 & e_4f_2 & f_4f_2 \\ e_1e_3 & f_1e_3 & e_2e_3 & f_2e_3 & e_3^2 & f_3e_3 & e_4e_3 & f_4e_3 \\ e_1f_3 & f_1f_3 & e_2f_3 & f_2f_3 & e_3f_3 & f_3^2 & e_4f_3 & f_4f_3 \\ e_1e_4 & f_1e_4 & e_2e_4 & f_2e_4 & e_3e_4 & f_3e_4 & e_4^2 & f_4e_4 \\ e_1f_4 & f_1f_4 & e_2f_4 & f_2f_4 & e_3f_4 & f_3f_4 & e_4f_4 & f_4^2 \end{bmatrix}_{(8\times8)}$$

$$X_6=\begin{bmatrix} u_{B1}^2 & l_{B1}u_{B1} & u_{B2}u_{B1} & l_{B2}u_{B1} & u_{B3}u_{B1} & l_{B3}u_{B1} & u_{B4}u_{B1} & l_{B4}u_{B1} \\ u_{B1}l_{B1} & l_{B1}^2 & u_{B2}l_{B1} & l_{B2}l_{B1} & u_{B3}l_{B1} & l_{B3}l_{B1} & u_{B4}l_{B1} & l_{B4}l_{B1} \\ u_{B1}u_{B2} & l_{B1}u_{B2} & u_{B2}^2 & l_{B2}u_{B2} & u_{B3}u_{B2} & l_{B3}u_{B2} & u_{B4}u_{B2} & l_{B4}u_{B2} \\ u_{B1}l_{B2} & l_{B1}l_{B2} & u_{B2}l_{B2} & l_{B2}^2 & u_{B3}l_{B2} & l_{B3}l_{B2} & u_{B4}l_{B2} & l_{B4}l_{B2} \\ u_{B1}u_{B3} & l_{B1}u_{B3} & u_{B2}u_{B3} & l_{B2}u_{B3} & u_{B3}^2 & l_{B3}u_{B3} & u_{B4}u_{B3} & l_{B4}u_{B3} \\ u_{B1}l_{B3} & l_{B1}l_{B3} & u_{B2}l_{B3} & l_{B2}l_{B3} & u_{B3}l_{B3} & l_{B3}^2 & u_{B4}l_{B3} & l_{B4}l_{B3} \\ u_{B1}u_{B4} & l_{B1}u_{B4} & u_{B2}u_{B4} & l_{B2}u_{B4} & u_{B3}u_{B4} & l_{B3}u_{B4} & u_{B4}^2 & l_{B4}u_{B4} \\ u_{B1}l_{B4} & l_{B1}l_{B4} & u_{B2}l_{B4} & l_{B2}l_{B4} & u_{B3}l_{B4} & l_{B3}l_{B4} & u_{B4}l_{B4} & l_{B4}^2 \end{bmatrix}_{(8\times8)}$$

With the definition and the structure of X, the problem (15)–(28) for TEST-4 can be easily reformulated as the primal SDP of (7) as following:

$$\begin{aligned} \min \quad & F = A_0 \bullet X \\ \text{s.t.} \quad & A_i \bullet X = b_i, \quad i = 1,\ldots,30 \\ & X \succ 0 \end{aligned} \tag{31}$$

where the coefficient matrices A_i have the same dimension as X, and the relationships between i and the constraints are below:

$A_1, A_2 \Longleftrightarrow$ Constraint (16), $A_3, A_4 \Longleftrightarrow$ Constraint (17),
$A_5, A_6 \Longleftrightarrow$ Constraint (18), $A_7, A_8 \Longleftrightarrow$ Constraint (19),
$A_9 \Longleftrightarrow$ Constraint (20), $A_{10} \Longleftrightarrow$ Constraint (21),
$A_{11}, A_{12} \Longleftrightarrow$ Constraint (22), $A_{13}, A_{14} \Longleftrightarrow$ Constraint (23),
$A_{15}, A_{16} \Longleftrightarrow$ Constraint (24), $A_{17}, A_{18} \Longleftrightarrow$ Constraint (25),
$A_{19}, A_{20}, A_{21}, A_{22} \Longleftrightarrow$ Constraint (26),
$A_{23}, A_{24}, A_{25}, A_{26} \Longleftrightarrow$ Constraint (27),
$A_{27}, A_{28}, A_{29}, A_{30} \Longleftrightarrow$ Constraint (28).

The structures as well as more detailed discussions about the construction of the matrices A_i ($i = 0, 1, \ldots, 30$) are described as follows:

$$A_i=\begin{bmatrix} W_{Gi} & 0 & & \cdots & & 0 \\ 0 & W_{SGi} & 0 & & \ddots & \\ & 0 & W_{Ri} & 0 & & \vdots \\ \vdots & & 0 & W_{SRi} & 0 & \\ & \ddots & & 0 & W_{Bi} & 0 \\ 0 & & \cdots & & 0 & W_{SBi} \end{bmatrix}_{(32\times32)} \tag{32}$$

1. Construction of A_0
For simplicity, the object function of minimizing the fuel cost (F_{Cost}) is shown here, and the other three object functions (P_{Loss2}, Q_{Loss}, and P_{Loss1}) can be constructed in the same way. With the definition for X in (30) and the structure of A_i in (32), A_0 is

$$A_0=\begin{bmatrix} W_{G0} & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}_{(32\times32)}$$

where

$$W_{G0}=\begin{bmatrix} a_{q1} & \frac{1}{2}a_{l1} & 0 & 0 \\ \frac{1}{2}a_{l1} & a_{f1} & 0 & 0 \\ 0 & 0 & a_{q2} & \frac{1}{2}a_{l2} \\ 0 & 0 & \frac{1}{2}a_{l2} & a_{f2} \end{bmatrix}_{(4\times4)}$$

2. Construction of A_i, $i = 1, \ldots, 30$
The constraints matrices are constructed in a similar way to A_0's. For illustration purpose, we describe the derivations of A_5 for (18) which originally is an equation constraint, A_{11} for (22) which originally is an inequality constraint, and A_{27} for (28) which is the auxiliary variable constraints, respectively.
(1) A_5
The constraint (18) is the active power flow equation of the ith generator in the system. For the generator G_1 in TEST-4, it is A_5, and there exists:

$$\begin{aligned} A_5 \bullet X &= P_{G1}d + \sum_{j=1}^{4}(-e_1e_jG_{1j} + e_1f_jB_{1j} - f_1f_jG_{1j} - f_1e_jB_{1j}) \\ &= P_{D1} \end{aligned}$$

More precisely, the structure of A_5 is

$$A_5=\begin{bmatrix} W_{G5} & 0 & & \cdots & & 0 \\ 0 & 0 & & \ddots & & \\ & & 0 & & \ddots & \vdots \\ \vdots & \ddots & & 0 & 0 & \\ & & \ddots & 0 & W_{B5} & 0 \\ 0 & & \cdots & & 0 & 0 \end{bmatrix}_{(32\times32)}$$

where

$$W_{G5} = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{(4\times4)}$$

$$W_{B5} = -\frac{1}{2}\times\begin{bmatrix} 2G_{11} & 0 & G_{12} & B_{12} & G_{13} & B_{13} & G_{14} & B_{14} \\ 0 & 2G_{11} & -B_{12} & G_{12} & -B_{13} & G_{13} & -B_{14} & G_{14} \\ G_{12} & -B_{12} & 0 & 0 & 0 & 0 & 0 & 0 \\ B_{12} & G_{12} & 0 & \vdots & \cdots & \cdots & \cdots & \vdots \\ G_{13} & -B_{13} & 0 & \vdots & \ddots & & & \vdots \\ B_{13} & G_{13} & 0 & \vdots & & \ddots & & \vdots \\ G_{14} & -B_{14} & 0 & \vdots & & & \ddots & \vdots \\ B_{14} & G_{14} & 0 & 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}_{(8\times8)}$$

(2) A_{11}
The constraint (22) is the maximum active power of the ith generator in the system. For the generator G_1 in TEST-4, there exists:

$$A_{11} \bullet X = P_{G1}d + u_{G1}^2 = \overline{P}_{G1}$$

More precisely, the structure of A_{11} is

$$A_{11} = \begin{bmatrix} W_{G11} & 0 & \cdots & \cdots & 0 \\ 0 & W_{SG11} & & \ddots & \\ \vdots & & 0 & & \vdots \\ \vdots & \ddots & & \ddots & \\ 0 & \cdots & \cdots & & 0 \end{bmatrix}_{(32\times32)}$$

where

$$W_{G11} = \frac{1}{2}\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{(4\times4)} \quad W_{SG11} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{(4\times4)}$$

(3) A_{27}
The constraint (28) is for the auxiliary variables which are equal to one. For the generator G_1 in TEST-4, it is A_{27}, and since $d_{G1} = 1$, there exists $d_{G1}^2 = 1$. Thus the structure of A_{27} is

$$A_{27} = \begin{bmatrix} W_{G27} & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 \end{bmatrix}_{(32\times32)}$$

where

$$W_{G27} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}_{(4\times4)}$$

Other constraints in the OPF quadratic form can be reformulated into $A_i \bullet X = b_i$ in the same way as that for A_5, A_{11}, or A_{27}.

3. Construction of b_i, $i = 1,\ldots,30$
The vector b is constructed using the constants located in the right hand side of the constraints (16)–(28) in TEST-4, namely,

$$\begin{aligned} b^{\mathrm{T}} = [&P_{D2}\ P_{D3}\ Q_{D2}\ Q_{D3}\ P_{D1}\ P_{D4}\ Q_{D1}\ Q_{D4}\ 0\ 1.05^2 \\ &\overline{P}_{G1}\ \overline{P}_{G2}\ \underline{P}_{G1}\ \underline{P}_{G2}\ \overline{Q}_{R1}\ \overline{Q}_{R2}\ \underline{Q}_{R1}\ \underline{Q}_{R2}\ldots\ldots \\ &\overline{V}_1^2\ \overline{V}_2^2\ \overline{V}_3^2\ \overline{V}_4^2\ \underline{V}_1^2\ \underline{V}_2^2\ \underline{V}_3^2\ \underline{V}_4^2\ 1\ 1\ 1\ 1] \end{aligned}$$

4.1.3. Mapping solutions to the classical OPF's

The solution of OPF in SDP (31) can be mapped to the solution of the classical OPF (1)–(6) in a straightforward way. more precisely, upon obtaining an optimal solution X^* of (31) and referring to the structure of X in (30), the optimal solution of (1)–(6) can be obtained in many ways since X^* contains enough information. The following procedure is for the purpose of illustration:

1. Active power $P_{Gi}(i \in S_G)$ and reactive power Q_{Gi} $(i \in S_R)$ equal to the square roots of the corresponding diagonal elements P_{Gi}^2 and Q_{Ri}^2 in the blocks X_1 and X_3 of (30), respectively.
2. Real part $e_i(i \in S_B)$ of voltage equal to the square roots of the corresponding diagonal elements e_i^2 in the block X_5 of (30). As for imaginary part $f_i(i \in S_B)$ of voltage, $f_i = e_if_i/e_i$, where e_if_i is the right-hand adjoining element to e_i^2 in X_5 of (30).

   In this mapping procedure, the elements of the matrix X^* mentioned above are more than enough to obtain the solution of (1)–(6). Therefore, no all elements are involved.

4.2. Choice of the initial value

In general, the better the initial values are selected, the faster the convergence of an algorithm will be. In [33], the initial values of the OPF variables are set as the averages of their limits which are feasible. However, it has been observed that the proposed algorithm in this study is not quite sensitive to the initial values whether they are feasible or unfeasible [28]. Experimentally, it works very well that the initial values are given by data whose order of magnitude is the same as that of an optimal solution (X^*, y^*, Z^*). Both ways for choosing the initial values lead to almost the same computing efficiency in this study. Therefore, in this study, the initial values do not have to be set scrupulously. Instead, for simplicity and without loss of any generality, they are chosen as follows:

1. The primal variable matrix $X^{(0)}$ and the slack variable matrix $Y^{(0)}$

$$X^{(0)} = \tau \times I, \quad Y^{(0)} = \tau \times I$$

where is a coefficient whose order of magnitude is the same as that of the optimal solution (X^*, y^*, Z^*), and I denotes the identity matrix which has the same dimension as the matrix X or Y in (30).

2. The dual variable vector $Y^{(0)}$

$$y^{(0)} = 0$$

where 0 denotes a full zero vector which has the same dimension as the vector formed by b_i in (31).

5. Test results and discussions

The proposed algorithm has been implemented in SDPAM [30] using Matlab 7.0 on a 2.8 GHz Pentium 4, Windows PC with 512 MB of RAM for six different test systems. The details of the six test systems, including a simple 4-bus system and five IEEE standard systems, are shown in Table 1. It should be noted that the OPF problem in SDP formulation can be solved with only two lines of Matlab codes using SDPAM, which actually benefits from the general uniform framework of the SDP software suite.

5.1. Solution quality of OPF using SDP

The OPF problem can be solved by IPM for nonlinear programming with excellent computing performance, and its solution is optimal [24]. However, the Jacobian matrices and the Hessian matrices have to be derived for each specific problem using NLP. Therefore, it is not convenient to develop a general uniform software solution for OPF problems using NLP.

On the other hand, the OPF problem can also be solved by IPM for semidefinite programming, and the numerical simulation in this study has demonstrated that its solution quality can be guaranteed like as for NLP. For the sake of space, we list in Table 2 only the results of TEST-4 by SDP and NLP which, as can be seen, are identical.

Table 3 shows the CPU time and the number of iterations for the optimal solutions by SOPF for the four kinds of objective functions of all six test systems.

It should be point out that OPF using SDP can not compete against OPF using NLP in terms of CPU times so far. Therefore, it is worthwhile to explore more efficient algorithms for SDP in the future.

Table 1
Test systems size, control variables and inequality constraints

Name of system	Number of buses/lines	Controllable/state variables (P_{Gi},Q_{Ri}/e_i,f_i)	Slack variables	Number of constrains
TEST-4	4/4	1, 2/4, 4	14	27
IEEE-14	14/20	5, 3/14, 14	44	75
IEEE-30	30/45	6, 6/30, 30	84	147
IEEE-57	57/78	4, 7/57, 57	136	253
IEEE-118	118/179	16, 54/118, 118	376	615
IEEE-300	300/409	21, 69/300, 300	780	1383

Table 2
Solutions by SDP and NLP (PER UNIT)

	By SDP		By NLP	
	Node	Value	Node	Value
Active	1	0.56818	1	0.56818
Power	4	0.30000	4	0.30000
Reactive	1	0.10048	1	0.10048
Power	4	0.27613	4	0.27613
Voltage	1	$0.98513 + 0.01257i$	1	$0.98513 + 0.01257i$
	2	$0.95976 - 0.09846i$	2	$0.95976 - 0.09846i$
	3	$1.08660 + 0.17117i$	3	$1.08660 + 0.17117i$
	4	$1.05 + 0i$	4	$1.05 + 0i$
Fuel cost($)		421.7741		421.7741

Table 3
CPU time (s), iterations

Name of systems	P_{Loss2}	Q_{Loss2}	F_{Cost}	P_{Loss1}
TEST-4	0.0156/9	0.0156/11	0.0156/9	0.0156/10
IEEE-14	0.0781/15	0.0781/14	0.0781/9	0.0938/14
IEEE-30	0.3906/17	0.3438/15	0.3125/12	0.3594/25
IEEE-57	2.2813/24	1.7656/18	1.6406/19	1.2188/17
IEEE-118	12.72/24	12.781/25	9.0938/22	6.9219/21
IEEE-300	121.38/31	122.90/27	102.05/26	154.31/21

5.2. Convergence of OPF by SDP

The complementary gap is a very important measure to judge the optimality of solutions, and its change reflects the characteristic of the algorithm. Fig. 2 shows how it reduces to zero (here, tolerance is $\varepsilon = 10^{-5}$) with iterations for the case IEEE-300 under four different kinds of objective functions. Generally speaking, the complementary gap should decrease to zero with iterations monotonously and rapidly for an algorithm to be efficient. For this reason, Fig. 2 clearly demonstrates the efficiency of the proposed algorithm. As for the remaining important feature of the algorithm, Fig. 3 shows the relative optimal values of four objective functions converge with iterations.

Just like IPM for NLP, tuning the centering parameter β introduced in (12) and the reduction factor γ introduced in (13) of IPM for SDP can exploit the tradeoff between the accuracy of optimal solutions and the computation times.

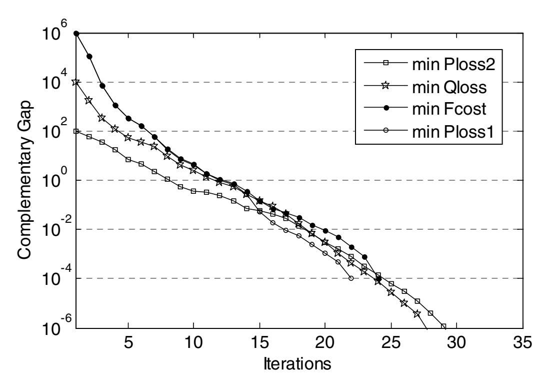

Fig. 2. Complementary gaps with iterations for IEEE-300 system.

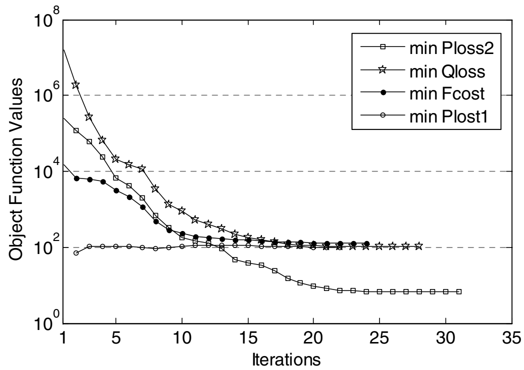

Fig. 3. Objective functions with iterations for IEEE-300 system.

The related discussion, however, is not included in this paper again due to space limitation.

6. Conclusions and future work

The solution to the OPF problem using semidefinite programming has been presented in this paper. The proposed method has been tested on six test systems with sizes ranging from small (4 buses) to large (300 buses), and the results are satisfactory. To the best of our knowledge, the SDP technique has not been used previously to solve the OPF problem.

The OPF is a very complex nonlinear and nonconvex problem which has been solved by various optimization methods. IPM is one of the best algorithms for the OPF problem in NLP forms [8]. The SDP is a technique for convex programming that can be implemented using IPM as well. Therefore, once a classical OPF problem is transformed into the semidefinite programming model, it turns to be a convex problem, and then can be solved using IPM for SDP and benefits from the SDP technique, as shown in the extensive numerical simulations presented in this paper.

However, the proposed method could not solve extremely large systems due to limitation in computer resources such as CPU and memory. Therefore, developing more powerful algorithms with sparsity technique and parallel computing technique for OPF using SDP will be an attractive research topic in the future.

Acknowledgements

The authors wish to thank Professor Masakazu Kojima from Department of Mathematical and Computing Sciences, Tokyo Institute of Technology and his SDPA project team. Thanks also go to Dr. Makoto Yamashita, member of the SDPA team from Department of Industrial Engineering and Management in Kanagawa University, Japan, for discussion on semidefinite techniques and on SDPAM which is one of the best MATLAB-based SDP solvers in the world [10].

References

[1] Carpentier J. Contribution á létude de Dispatching Economique. Bull Soc Fr Electric 1962;3:431–47.

[2] Dommelll HW, Tinney WF. Optimal power flow solutions. IEEE Trans PAS 1968;87:1866–76.

[3] Momoh JA, Koessler RJ, Bond MS, Stott B, Sun D, Papalexopoulos A, et al. Challenges to optimal power flow. IEEE Trans Power Syst 1997;12:444–55.

[4] Granville S. Optimal reactive dispatch through interior point methods. IEEE Trans Power Syst 1994;9:136–46.

[5] Wei H, Sasaki H, Yokoyama R. An application of interior point quadratic programming algorithm to power system optimization problems. IEEE Trans Power Syst 1996;11:260–7.

[6] Wu Y-C, Debs AS, Marsten RE. A direct nonlinear predictor-corrector primal–dual interior point algorithm for optimal power flows. Presented at power industry computer application conference, Scottsdale, AZ, USA; 1993.

[7] Quintana VH, Torres GL, Medina-palomo J. Interior-point methods and their applications to power systems: a classification of publications. IEEE Trans Power Syst 2000;15:170–6.

[8] Adler I, Alizadeh F. Primal–dual interior point algorithms for convex quadratically constrained and semidefinite optimization problems. Technical report, Rutcor, Rutgers University, New Brunswick, NJ; 1995.

[9] Todd MJ. Semidefinite optimization. Acta Numer 2001;10: 515–60.

[10] Mittelmann HD. An independent benchmarking of SDP and SOCP solvers. Math Program 2003;B:407–30.

[11] Wolkowicz H. Handbook of applied optimization. Oxford University Press; 2001.

[12] P. Parrilo. Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization. PhD thesis. California: Caltech, Pasadena; 2000.

[13] Parrilo P. Semidefinite programming relaxations and algebraic optimization in control. Eur J Control 2003;9:307–21.

[14] Alkire B, Vandenberghe L. Handling nonnegative constraints in spectral estimation. Technical report, Electrical Engineering Department, UCLA, Los Angeles, CA; 2000.

[15] Alkire B, Vandenberghe L.Convex optimization problems involving finite autocorrelation sequences. Technical report, Electrical Engineering Department, UCLA, Los Angeles, CA; 2001.

[16] Xiao L, Boyd S. Fast linear iterations for distributed averaging. Technical report, Stanford University, Stanford, CA; 2003.

[17] Benson SJ, Ye Y, Zhang X. Solving large scale sparse semidefinite programs for combinatorial optimization. SIAM J Optim 2000;10:443–61.

[18] Goemans MX, Rendl F. Semidefinite programming in combinatorial optimization. Math Program 1997;79:143–61.

[19] Goemans MX. Semidefinite programming and combinatorial optimization. Documenta mathematica, Extra Volume ICM 1998. p. 657–66.

[20] Fuentes-Loyola R, Quintana VH. Medium-term hydrothermal coordination by semidefinite programming. IEEE Trans Power Syst 2003;18:1515–22.

[21] Madrigal M, Quintana VH. Semidefinite programming relaxations for {0,1}-power dispatch problems. Presented at proceeding of IEEE-PES 1999 Summer Meeting, Edmonton, Al., Canada; 1999.

[22] Jabr RA. A primal–dual interior-point method to solve the optimal power flow dispatching problem. Optim Eng 2003;4:309–36.

[23] Alizadeh F, Haeberly J-PA, Overton ML. Primal–dual interior-point methods for semidefinite programming: convergence rates, stability and numerical results. Computer Science Department, Technical Report 721, New York University, May 31; 1996.

[24] Wei H, Sasaki H, Kubokawa J, Yokoyama R. A interior point nonlinear programming for optimal power flow problems with a novel data structure. IEEE Trans Power Syst 1998;13:870–7.

[25] Mizuno S, Kojima M, Todd MJ. Infeasible-interior-point primal–dual potential reduction algorithms for linear programming. SIAM J Optim 1995;5:52–67.
[26] Alizadeh F. Combinatorial optimization with interior point methods and semidefinite matrices. PhD Thesis, University of Minnesota; 1991.
[27] Wright S. Primal–dual interior-point methods. Publisher: Soc for Industrial & Applied Math, June 1997.
[28] Fujisawa K, Fukuda M, Kojima M, Nakata K. Numerical evaluation of SDPA (Semidefinite Programming Algorithm). Technical report series B: operations research, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, September 1998.
[29] Yamashita M, Fujisawa K, Kojima M. SDPARA: SemiDefinite programming algorithm PARAllel version. Parallel Comput 2003;29:1053–67.
[30] Satoshi Matsuyama SN, Katsuki Fujisawa, Kazuhide Nakata, Masakazu Kojima. SDPA-M (SemiDefinite Programming Algorithm in MATLAB) user's manual — Version 1.00. Technical report series B: operations research, Department of Mathematical and Computing Sciences, Tokyo Institute of Technology, Tokyo, January 2000.
[31] Lov'asz L, Schrijver A. Cones of matrices and set-functions and 0–1 optimization. SIAM J Optim 1991;1:166–90.
[32] Fujisawa K, Kojima M, Nakata K. Exploiting sparsity in primal–dual interior-point methods for semidefinite programming. Math Program 1997;79:235–53.
[33] Wei H, Sasaki H, Kubokawa J, Yokoyama R. Large scale hydrothermal optimal power flow problems based on interior point nonlinear programming. IEEE Trans Power Syst 2000;15:396–403.