Embedding sensitivity theory in ordinal optimization for decentralized optimal power flow control

Shieh-Shing Lin\textsuperscript{a,\textdagger}, Shih-Cheng Horng\textsuperscript{b}, Ch'i-Hsin Lin\textsuperscript{c}

\textsuperscript{a} Department of Electrical Engineering, St. John's University, 499, Sec. 4 Tam King Road, Tamsui, Taipei 25135, Taiwan, ROC
\textsuperscript{b} Department of Computer Science and Information Engineering, Chuyang University of Technology, Taichung, Taiwan, ROC
\textsuperscript{c} Department of Electronics Engineering, Kaou Yuan University, Kaouhsiung, Taiwan, ROC

A R T I C L E   I N F O

Article history:
Received 11 May 2011
Received in revised form 27 August 2011
Accepted 5 September 2011
Available online 22 October 2011

Keywords:
Sensitivity theory
NP-hard optimization problem
Ordinal optimization
Decentralized optimal power flow control
Power systems

A B S T R A C T

In this paper, the decentralized optimal power flow with continuous and discrete control variables problem is firstly formulated as an NP-hard optimization problem - Block Additive constrained with Continuous and Discrete variables (BACD) problem. Secondly, an algorithm of embedding sensitivity theory (ST) in ordinal optimization (OO), abbreviated as STOO, is proposed for solving this NP-hard optimization problem. The STOO algorithm consists of three stages and three models of performance evaluation. The proposed method not only copes with the computational complexity due to huge solution space but also obtains a good enough solution with high probability guaranteed by the OO theory. Finally, this work demonstrates the computational efficiency of the STOO algorithm via various tests on the IEEE 118-bus and 244-bus systems partitioned into four subsystems using a 4-PC network and compares the results with those obtained using other heuristic methods, Genetic Algorithm, Tabu Search, Ant Colony Optimization and Simulated Annealing. Test results show the validity, robustness and excellent computational efficiency of the STOO algorithm for obtaining a good enough solution.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

The network topology systems constructed by interconnected subsystems are mostly large-scale, such as power systems, communication systems and transportation systems. Operations management and state control are usually key processes of these systems. A centralized control center is typically employed to manage the operations of the whole system. Because the popularity of the computer communication technologies, the manner of decentralized management and control has recently become a novel trend. However, solving a nonlinear constrained optimization problem such as the state estimation for a large-scale power system is computationally intractable due to its high dimension and nonlinearity.

Power system state estimation (PSSE) is a very large-scale, sparse and non-linear problem in power systems. Various numerical approaches had been proposed for solving these problems [1,2]. The purpose of the PSSE is to obtain the best estimate of the power system states, including voltage magnitude and phase angles of all buses, given a redundant set of measurements. These measurements contain bus voltage magnitudes, power transmission line flows and node power injections. Problems of PSSE with constraints can be formulated as a class of constrained weighted least squares problems based on their mathematical formulations. Numerous solution methods had been developed to solve these problems [3,4]. PSSE with continuous variables problem is a classical convex programming problem. Nonlinear programming algorithms are typically used to solve such problem [5]. Recently, several efficient methods are successfully developed to solve the PSSE problem with equality and/or inequality constraints in power systems [6,7]. However, the above studies focus on continuous variables only, while few attempt to solve PSSE problem with continuous and/or discrete variables. In practice, continuous and/or discrete variables exist in PSSE problem such as the optimal power flow problems with discrete variables [8]. These problems are NP-hard optimization problems due to the huge solution space of discrete variable settings. Methods of computational intelligence provide a possible solution for NP-hard optimization problems, such as the evolution strategy (ES) [9], evolutionary programming (EP) [10], Genetic Algorithm (GA) [11], Tabu Search (TS) [12], Ant Colony Optimization (ACO) [13] and Simulated Annealing (SA) [14]. Nonetheless, NP-hard optimization problems with both continuous and discrete variables are still difficult to solve due to huge solution space.

In this paper, we firstly formulate the decentralized optimal power flow with continuous and discrete control variables problem as a NP-hard optimization problem - Block Additive constrained with Continuous and Discrete variables (BACD) problem. To cope with the computational complexity due to huge solution space, a recently developed optimization theory called ordinal optimization...
(OO) [15] is employed to solve for a good enough solution with high probability instead of searching the best solution for sure. Using limited computing time to solve for a good enough solution of the NP-hard optimization problem is the core concept of OO theory. By relaxing the definition of optimality and softening the goal of optimization, OO makes the NP-hard optimization problems easier as well as the solution process faster. The OO theory has been successfully applied to solve the NP-hard problems such as the wafer probe testing process [16], G/G/1/K polling systems with k-limited service discipline [17], resource allocation of grid computing system [18], and cyclic service of the centralized broadband wireless networks [19]. The first step of OO theory is to evaluate each solution roughly using a crude model, and then based on which to select a subset of good enough solutions. Each solution in the good enough subset will be evaluated using the exact model, and the best one is the final good enough solution. Thus, the purpose of this paper is to present an algorithm of embedding sensitivity theory (ST) [20] in ordinal optimization (OO), abbreviated as STOO, for solving the developed mathematical formulation of the decentralized optimal power flow with continuous and discrete control variables problem – Block Additive constrained with Continuous and Discrete variables (BACD) problem is the first contribution of this paper. Application of the STOO algorithm to solve the BACD problem of the IEEE 118-bus and 244-bus systems partitioned into four subsystems using a 4-PC network is another contribution of this paper.

The remainder of this paper is organized as follows. Section 2 describes the problem statement and mathematical formulation of the considered NP-hard optimization problem – BACD problem. Section 3 presents the STOO algorithm for solving the considered problem. Section 4 tests the STOO algorithm on two power systems, the IEEE 118-bus and 244-bus systems partitioned into four subsystems, and compares the results with those obtained using other heuristic approaches. Finally, Section 5 draws a conclusion.

2. Problem statement and mathematical formulation

First, we introduce the notations for the considered NP-hard problem – BACD problem. Consider the th area or subsystem, \( i = 1, \ldots, I \).

<table>
<thead>
<tr>
<th>BACD</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( I )</td>
<td>Number of interconnected subsystems.</td>
</tr>
<tr>
<td>( x_i )</td>
<td>The vector of continuous state variables, which can be voltage magnitudes and phase angles in power systems and ( x^T = [x_1^T, \ldots, x_I^T] ).</td>
</tr>
<tr>
<td>( x_{is} )</td>
<td>The vector of boundary states of other subsystems connecting with subsystem ( i ).</td>
</tr>
<tr>
<td>( a_{di} )</td>
<td>Discrete variables in the considered NP-hard problem, which can be switching shunt capacitor banks and transformer taps in power systems.</td>
</tr>
<tr>
<td>( g_i(x_i, x_{is}, a_{di}) )</td>
<td>The equality constraints.</td>
</tr>
<tr>
<td>( h_i(x_i) \leq 0 )</td>
<td>The inequality constraints.</td>
</tr>
<tr>
<td>( f(x, a_{di}) )</td>
<td>Objective function of BACD problem, ( f(x, a_{di}) = \sum f_i(x_i, a_{di}) ).</td>
</tr>
<tr>
<td>( A_d )</td>
<td>Solution space of ( a_{di} ) and ( A_d = \cup_{i=1}^{I} A_{di} ).</td>
</tr>
<tr>
<td>( u_i )</td>
<td>Lagrange multiplier sub vector of ( u ), ( u^T = [u_1, \ldots, u_I]^T ), corresponding to the equality constraints.</td>
</tr>
<tr>
<td>( \varepsilon )</td>
<td>Positive real numbers.</td>
</tr>
</tbody>
</table>

This work considered a network topology power system constructed by \( I \) interconnected subsystems. Fig. 1 shows an example constructed by four interconnected subsystems, where a single bar is a state indicated the complex voltage of a power network bus, and a line connecting two bars represents the physical connection such as a transmission line in power systems. It should be noticed that \( x_{i1} \) is a sub vector of \( x_i \). In Fig. 1, the vector of boundary states of Subsystem 1 is \( x_{i1} = (x_{11}, x_{12}, x_{13}, x_{14}, x_{15}) \), and the vector of boundary states of the other subsystems connecting with Subsystem 1 is \( x_{i1} = (x_{11}, x_{22}, x_{13}, x_{41}, x_{42}, x_{43}) \).

The mathematical formulation of considered BACD problem is described in the following:

\[
\begin{align*}
\min_{x_i, a_{di}} & \quad \sum_{i=1}^{I} f_i(x_i, a_{di}) \quad \text{(1a)} \\
\text{s.t.} & \quad g_i(x_i, x_{is}, a_{di}) = 0, \quad i = 1, \ldots, I \quad \text{(1b)} \\
& \quad h_i(x_i) \leq 0, \quad i = 1, \ldots, I \quad \text{(1c)} \\
& \quad a_{di} \in A_{di}, \quad i = 1, \ldots, I \quad \text{(1d)} 
\end{align*}
\]

The goal of the BACD problem is to find an optimal continuous variable \( x_i \) and discrete variable \( a_{di} \) from the solution space \( A_{di} \), \( i = 1, \ldots, I \), such that the objective function \( f(x, a_{di}) = \sum f_i(x_i, a_{di}) \) is minimized, while satisfying the equality constraints (1b) and the inequality constraints (1c). In the equality constraints (1b), \( x_{is} \) denotes the vector of boundary states of other subsystems connecting with subsystem \( i \).

3. Embedding sensitivity theory in ordinal optimization for solving the BACD problem

The BACD problem is difficult to solve by the existing exhaustive searching methods [9–14] due to huge size of solution space. For example, problem (1a), (1b), (1c), (1d) is a distributed nonlinear constrained optimization problem (NCOP) with high dimension for a given \( a_{di} \). To evaluate the performance of a \( a_{di} \), we need to solve one simulation run of distributed NCOP. Regarding the huge size of \( A_{di} \), if the whole system has \( b = 40 \) discrete variables and each variable has \( p = 4 \) possible discrete values, there are \( p^b = 4^{40} \approx 10^{24} \) possible solutions of \( a_{di} \)’s. Thus, if we employ the existing exhaustive searching method to search the optimal solution \( a_{di} \) in \( A_{di} \), we need to solve more than \( 10^{24} \) simulation runs of distributed NCOP. Because the huge size of \( A_{di} \), it is computationally intractable to solve for an optimal solution of BACD problem shown in (1a), (1b), (1c), (1d) using conventional exhaustive searching method. Therefore, to cope with the computationally expensive, we propose an algorithm of embedding ST in OO to obtain a good enough solution instead of optimal \( a_{di} \).
Combined with the successive quadratic programming (SQP) method [21,22], we firstly formulate the BACD problem as a standard form of ordinal optimization problems. The SQP method is one of the most popular and robust algorithms for nonlinear continuous optimization. This method is based on solving a series of subproblems designed to minimize a quadratic model of the objective subject to a linearization of the constraints. The SQP method was widely employed in the previous researches [23,24] and the recent papers [8,25]. For a given discrete sample $a_i$, the SQP method used the following iteration to solve for an optimal continuous variable $x_i$ of (1a), (1b), (1c), (1d):

$$x_i(t + 1) = x_i(t) + \alpha_i(t) \Delta x_i(t)$$

(2)

where $\alpha$ is an iteration index, $\alpha_i(t)$ is a positive step-size, and $\Delta x_i(t)$ is the optimal descent direction obtained by solving the BACD's quadratic approximate programming problem as follows.

$$\min_{\Delta x_i(t)} \sum_{i=1}^{M} \frac{1}{2} \Delta x_i^T \text{diag} \left[ \nabla^2_r f_i(x, a_i) \right] \Delta x_i + \nabla_x f_i(x, a_i) \Delta x_i$$

s.t.

$$\begin{align*}
& g_i(x, x_i, a_i) + \nabla_x g_i(x, x_i, a_i) \Delta x_i + \nabla_{x_i} g_i(x, x_i, a_i) \Delta x_i = 0, \quad i = 1, \ldots, l \\
& h_i(x_i) + \nabla_{x_i} h_i(x_i) \Delta x_i \leq 0, \quad i = 1, \ldots, l \\
& a_i \in A_i, \quad i = 1, \ldots, I
\end{align*}$$

(3a), (3b), (3c), (3d)

where $\nabla^2_r f_i(x, a_i)$ denotes the Hessian, $\nabla_x f_i(x, a_i)$ and $\nabla_{x_i} g_i(x, x_i, a_i)$ denote the gradients, $I_i$ is an identity matrix with suitable dimension for subsystem $i$, and $\delta > 0$ is a small positive scalar but enough to make the Hessian positive definite. In each iteration of (2), a new quadratic approximate programming problem (3a), (3b), (3c), (3d) will be solved using the solution obtained from the previous iteration.

When all the discrete samples $a_i \in A_i, i = 1, \ldots, I$ are considered, the BACD's quadratic approximate programming problem (3a), (3b), (3c), (3d) is stated as follows.

$$\begin{align*}
& \min_{a_i \in A_i} \{ \min_{\Delta x_i(t)} \sum_{i=1}^{M} \frac{1}{2} \Delta x_i^T \text{diag} \left[ \nabla^2_r f_i(x, a_i) \right] \Delta x_i + \nabla_x f_i(x, a_i) \Delta x_i + \nabla_{x_i} g_i(x, x_i, a_i) \Delta x_i + \nabla_{x_i} g_i(x, x_i, a_i) \Delta x_i = 0, \\
& h_i(x_i) + \nabla_{x_i} h_i(x_i) \Delta x_i \leq 0, \quad i = 1, \ldots, I
\end{align*}$$

(4)

For the sake of simplicity in presentation, problem (4) can be further rewritten as

$$\begin{align*}
& \text{Quadratic BACD (aq)} \\
& = \left\{ \min_{a_i \in A_i} \{ \sum_{i=1}^{M} \frac{1}{2} \Delta x_i^T \text{diag} \left[ \nabla^2_r f_i(x, a_i) \right] \Delta x_i + \nabla_x f_i(x, a_i) \Delta x_i \Delta x_i + \nabla_{x_i} g_i(x, x_i, a_i) \Delta x_i + \nabla_{x_i} g_i(x, x_i, a_i) \Delta x_i = 0, \\
& h_i(x_i) + \nabla_{x_i} h_i(x_i) \Delta x_i \leq 0, \quad i = 1, \ldots, I \} \right. \}
\end{align*}$$

(5)

The mathematical formulation of problem (5) is the same to the standard forms treated in the OO theory [15]. Therefore, combined with the SQP method, we have reformulated the BACD problem as a standard form of ordinal optimization problems.

### 3.1. The sensitivity theory based goal softening method

The concept of OO theory is using a computationally easy estimated model to select a good enough subset of solutions with high probability instead of searching the best solution for sure. A crude model is firstly used to evaluate the estimated performances of all solutions in the candidate solution set and rank them to select a set of top ranked solutions. If a more refined crude model is employed, there will be more actual good solutions contained in the selected set of top ranked solutions. The aim of OO theory is to decrease the searching space gradually, which is carried out in the following three stages [15]: (a) Uniformly select $N$ settings from $A_N$ as the Representative Set, $RS$, which is the candidate settings of the Good enough Subset, $GS$. (c) Estimate and rank the $s$ settings in $SS$ using an accurate model, then select the top $k, k \geq 1$ settings ($k$ is called the numbers of alignment in OO theory [15]). The OO theory guarantees that for $N$ settings in (a) and a crude model with moderate noise in (b), the top one setting (i.e., $k = 1$) selected from (c) within $s$ settings must belong to the $GS$ with probability around 0.95, where $GS$ represents a collection of the top $5\%$ actually good solutions among $N$. According to this claim, we firstly evaluate the estimated performances of all discrete solutions in the candidate solution set using a crude model and then select a set of good enough solutions, $SS$. Since there are more good enough solutions in $SS$, we use the accurate model to evaluate the estimated performances of the $s$ solutions in $SS$. The best one among these $s$ solutions is the good enough solution that we seek.

In [16], we proposed an OO theory based two-stage algorithm to solve for a good enough solution of the stochastic simulation optimization problem with huge input-variable space. Then the OO theory based two-stage algorithm was applied to the problem of reducing overkills and retests in a wafer probing test process and obtained successful results [16]. Therefore, based on the OO theory shown above, the proposed methodology consists of three stages stated below. First, we employ a crude model to select $N$ discrete settings of $a_i$ from $A_N$ to construct the Representative Set, $RS$. Second, a more refined crude model is proposed and the sensitivity theory based goal softening method is utilized to construct the Selected Subset, $SS$. Finally, an exact model is employed to evaluate the performances of $s$ candidate of the more refined crude model, and the best one $a_i$ with smallest objective value $\sum_{i=1}^{s} \mathbf{f}_i$ will be the good enough solution that we seek.

#### 3.1.1. Crude model

Because Eq. (1b) contained the discrete variables, we employ a crude model which relaxes the inequality constraints and regards the discrete variables as continuous variables. Therefore, formulation (5) can be rewritten as

$$\begin{align*}
& \min_{a_i \in A_i} \{ \sum_{i=1}^{M} \frac{1}{2} \Delta x_i^T \text{diag} \left[ \nabla^2_r f_i(x, a_i) \right] \Delta x_i + \nabla_x f_i(x, a_i) \Delta x_i + \nabla_{x_i} g_i(x, x_i, a_i) \Delta x_i + \nabla_{x_i} g_i(x, x_i, a_i) \Delta x_i = 0, \\
& h_i(x_i) + \nabla_{x_i} h_i(x_i) \Delta x_i \leq 0, \quad i = 1, \ldots, I
\end{align*}$$

(6)

The mathematical formulation shown in (6) is a type of purely continuous variables distributed nonlinear constrained optimization problems (NCPs). We can employ the previous developed method called parallel dual type method [25] to solve for $x_i$ and $a_i, i = 1, \ldots, I$. Furthermore, the optimal round off integer values of $a_i, i = 1, \ldots, I$ (round off to the upper or lower bound of $a_i, i = 1, \ldots, I$) are fixed to the nearest discrete values value $a_i, i = 1, \ldots, I$. Suppose there are $n_i$ discrete variables in subsystem $i$ and each discrete variable has $p$ possible settings, the total number of discrete variables is $\sum_{i=1}^{I} n_i$ in the whole system and the cardinality of solution space is $p^{\sum_{i=1}^{I} n_i}$. Hence, combining the yet fixed sub optimal closer integer values of $2^k a_i, i = 1, \ldots, I$ in all subsystems to form these $2^{\sum_{i=1}^{I} n_i}$ possible $a_i$'s, say, $a_i(j), j = 1, \ldots, 2^{\sum_{i=1}^{I} n_i}$, and $a_i(j)$, the sub vector of $a_i(j)$ corresponding to subsystem $i$, is one of the $2^n$ $a_i(j)$'s from subsystem $i$ to construct the Representative...
Set. RS. We label these \(N\) discrete solutions of \(a_d\) as \(a_d(j), j = 1, \ldots, N\). Therefore, the cardinality of solution space is reduced from \(2^\sum_{i=1}^{m}\) to \(N\). Next, we regard the sensitivity theory based goal softening method as a more refined crude model which is used to construct the Selected Subset. SS.

3.1.2. More refined crude model

Since the approximately solution is obtained by relaxing the inequality constrains of the roughly model, it is necessary to involve the inequality constrains in next stage. After inclusion of the inequality constrains, Eq. (6) can be stated as below.

\[
\begin{align*}
\min \left\{ \frac{1}{2} \Delta x_i^T \text{diag}[\nabla^2 f(x_i, a_i)] \Delta x_i + \Delta x_i^T \Delta u_i + \nabla_x f^T(x_i, a_i) \Delta x_i \right\} \\
\left| g(x_i, x_k, a_i) + \nabla_x g(x_i, x_k, a_i) \Delta x_i + \nabla_x g_i^T(x_i, x_k, a_i) \Delta x_i \right| = 0, \\
h(x_i) + \nabla_x h_i^T(x_i) \Delta x_i \leq 0, \quad g_i \leq a_i \leq a_i \}
\end{align*}
\]

(7)

Now, we are ready to employ the sensitivity theory based goal softening method to build the Selected Subset. SS. The sensitivity theory states that the sensitivity or the gradient of \(f\) with respect to the value changed on the equality constraint function \(g(x_i, x_k, a_i) + \nabla_x g(x_i, x_k, a_i) \Delta x_i + \nabla_x g_i^T(x_i, x_k, a_i) \Delta x_i \Delta u_i = 0, \quad i = 1, \ldots, I\) equals to the value of negative Lagrange multiplier, \(u_i\). We will estimate the performance of these \(N\) &s obtained in the roughly model and select the top ranked s, say 45, \(a_d\)'s [26].

The sensitivity corresponding to the BACD problem can be obtained as follow.

\[
\Delta f\left(a_d(l)\right) \equiv f_i^T \nabla a_i g(x_i, x_k, a_i) + \nabla_x g(x_i, x_k, a_i) \Delta a_i + \nabla_x g_i^T(x_i, x_k, a_i) \Delta x_i \Delta a_i + \nabla_x g_i^T(x_i, x_k, a_i) \Delta x_i \Delta a_i \]

(8)

where \(\Delta f\left(a_d(l)\right) = f_i^T (x_i, a_i') - f_i^T (x_i, a_d'(l))\).

Proof of (8).

For the sake of simplicity in explaining the formal proof of (8), the coefficients of the matrix \(\frac{1}{2} \Delta x_i^T \text{diag}[\nabla^2 f(x_i, a_i)] + \Delta u_i \Delta x_i\) are assumed to be very small that we can neglect. We employ the duality-based method [24] to solve Eq. (7). The duality-based method solves the following dual problem instead of solving Eq. (5) directly.

\[
\max \sum_{i=1}^{I} \phi_i(u_i)
\]

(9)

where \(\phi_i(u_i)\) is the dual function as below.

\[
\min \left\{ \frac{1}{2} \Delta x_i^T \text{diag}[\nabla^2 f(x_i, a_i)] \Delta x_i + u_i^T g(x_i, x_k, a_i) + \nabla_x g(x_i, x_k, a_i) \Delta x_i + \nabla_x g_i^T(x_i, x_k, a_i) \Delta x_i \right\}
\]

(10)

where \(\Omega = \{ \Delta x_i, a_i, h(x_i) + \nabla_x h_i^T(x_i) \Delta x_i \leq 0, \quad g_i \leq a_i \leq a_i \}
\]

(11)

According the duality theorem of linear programming and the first-order necessary conditions, we can obtain

\[
\nabla_x g^T(x_i, x_k, a_i) + u_i^T (\nabla_x g(x_i, x_k, a_i)) = 0
\]

(12a)

\[
u_i^T (\nabla_x g_i^T(x_i, x_k, a_i)) = 0
\]

(12b)

Multiply Eqs. (12a) and (12b) by \(\Delta x_i\) and \(\Delta x_i\) respectively, from the right-hand side and add them together, we yield

\[
\nabla_x g^T(x_i, x_k, a_i) \Delta x_i + u_i^T (\nabla_x g(x_i, x_k, a_i)) \Delta x_i + u_i^T (\nabla_x g_i^T(x_i, x_k, a_i)) \Delta x_i = 0
\]

(13)

Since \(f_i(x_i', a_i') = f_i(x_i, a_d'(l)) + \nabla_x f(x_i, a_i) \Delta x_i\), then

\[
\Delta f\left(a_d(l)\right) = f_i(x_i, a_i') - f_i(x_i, a_d'(l))
\]

(14)

The equality constraint function \(g_i(x_i, x_k, a_i) + \nabla_x g(x_i, x_k, a_i) \Delta x_i + \nabla_x g_i^T(x_i, x_k, a_i) \Delta x_i = 0, \quad i = 1, \ldots, I\) can be rewritten as

\[
g_i(x_i, x_k, a_i) + \nabla_x g(x_i, x_k, a_i) \Delta x_i + \nabla_x g_i^T(x_i, x_k, a_i) \Delta x_i = 0
\]

(15)

Since \(\Delta x_i'\) and \(a_i'\) are continuous optimal solutions, \(g_i(x_i', x_k', a_i') + \nabla_x g(x_i', x_k', a_i') \Delta x_i' + \nabla_x g_i^T(x_i', x_k', a_i') \Delta x_i' = 0\) will be zero. From (15), we obtain

\[
\nabla_x g(x_i, x_k, a_i) \Delta x_i + \nabla_x g_i^T(x_i, x_k, a_i) \Delta x_i = 0
\]

(16)

Substitute (14) and (16) into (13), we obtain the result

\[
\Delta f\left(a_d(l)\right) = f_i(x_i, a_i') - f_i(x_i, a_d'(l))
\]

(17)

We process the BACD problem separately in every subsystem \(i, j = 1, \ldots, I\) using the sensitivity theory based goal softening method. The subsystem \(i\) will send the \(2^n\) pairs of \((a_d(l), \Delta f\left(a_d(l)\right))\)'s to the source subsystem. Therefore, these \(2^\sum_{i=1}^{m}\) possible \(a_d\)'s are of \(a_d(j), j = 1, \ldots, 2^\sum_{i=1}^{m}\) in the source subsystem. The sub vector of \(a_d(j)\) corresponding to subsystem \(i, j = 1, \ldots, I\), is one of the \(2^n\) \(a_d\)'s coming from subsystem \(i\). According to the linear property of the sensitivity theory [20], we have \(\Delta f\left(a_d(l)\right) = \sum_{i=1}^{I} \Delta f(a_d(i))\), where \(\Delta f(a_d(i))\) is one of the \(2^n\) \(\Delta f\left(a_d(l)\right)\)'s coming from subsystem \(i\). Then, the source subsystem will rank these \(2^\sum_{i=1}^{m}\) \(a_d(j)\)'s based on the corresponding values of \(\|\Delta f\left(a_d(j)\right)\|\) such that the smaller \(\|\Delta f\left(a_d(j)\right)\|\) has higher rank. We choose the top ranked \(s a_d(j)\)'s and relabeled them as \(a_d(j)\), \(m = 1, \ldots, s\). The source subsystem will send the corresponding sub vectors, \(a_d(j)\), \(m = 1, \ldots, s\) to subsystem \(i, j = 1, \ldots, I\). Finally, we further reduce the size of the candidate solution set from \(2^\sum_{i=1}^{m}\) to \(s\) and build the Selected Subset SS, \(a_d(j)\), \(m = 1, \ldots, s\).

3.1.3. Exact model

An exact model is used to evaluate the performance of \(s a_d(j)\)'s in the more refined crude model, and the best one \(a_d(j)\) with smallest objective value \(\sum_{i=1}^{I} f_i(x_i(a_d(j)))\) will be the good enough solution that we seek. The details of exact model are described below.

The \(a_d\) is replaced by the fixed \(a_d(j)\) in the exact model, and the considered BACD problem becomes a purely continuous variables distributed NCOP. We proceed with picking the best \(a_d(j)\) among \(a_d(j), \ldots, a_d(j)\) as follows. While receiving the corresponding sub vectors of \(s\) discrete solutions resulted in the more refined crude model from the source subsystem, all subsystems will cooperate to solve these \(s\) purely continuous variables distributed NCOPs. Let \(x^*(a_d(j)) = (x_i^*(a_d(j)), \ldots, x_i^*(a_d(j)))\) denote the optimal solution of the continuous variables distributed NCOP with the fixed \(a_d(j)\). Once these \(s\) continuous variables distributed NCOPs are solved, each subsystem will send the obtained \(s\) pairs of \((a_d(j), f_i(x_i^*(a_d(j))))\) to the source subsystem. The source subsystem just calculates the objective value of the overall system.
for the given $a_d(j_m)$ by taking the sum $\sum_{i=1}^{m}f_i(x_i(a_d(j_m)))$. Let $a_d(j_m)$ denote the $m$th discrete variable among $a_d(j_1), \ldots, a_d(j_l)$, and $a_d(j_m)$ denote the minimum $a_d(j_m)$, which is determined by

$$a_d(j_m) = \arg\left\{ \min_{a_{d(i)} \in \mathbb{S}} \sum_{i=1}^{l} f_i(x_i(a_d(j_m))) \right\}$$

(17)

Thus, the $a_d(j_m)$ associated with the $x_i(a_d(j_m))$ will be the good enough solution that we seek.

3.2. Algorithm of embedding ST in OO

Now, we state the algorithm of embedding ST in OO for BACD problem as following.

The STOO algorithm:

- **Step 1.** The source subsystem firstly commands all subsystems to start. 
- **Step 2.** After receiving the command from the source subsystem, all subsystems relax the inequality constraints and treat the discrete variables as the continuous variables. Next, all subsystems employ the parallel dual type method to reduce the size of the primitive candidate solution set, from $A_0$ to $N a_d$ and construct the Representative Set $\mathbb{N}$.
- **Step 3.** All subsystems compute $\Delta f_i(a_d(l))$ by (8) and send the $2^m$ pairs of $(a_d(l), \Delta f_i(a_d(l)))$, $l = 1, \ldots, 2^m$ to the source subsystem.
- **Step 4.** After receiving the $2^m$ pairs of $(a_d(l), \Delta f_i(a_d(l)))$, $l = 1, \ldots, 2^m$, from subsystem $i, i = 1, \ldots, l$, the source subsystem picks the best $s$ from the $N a_d$’s based on the more refined crude model. Relabeled these picked $s a_d$’s as $a_d(j_m)$, $m = 1, \ldots, s$ and construct the Selected Subset, $\mathbb{S}$, then send $a_d(j_m)$, $m = 1, \ldots, s$ to all subsystems.
- **Step 5.** After receiving the $s$ sub vectors $a_d(j_m)$, $m = 1, \ldots, s$ from the source subsystem, all subsystems start to solve these $s$ distributed NCOPs using parallel dual type method. Once the $s$ distributed NCOPs are solved, send the $s$ pairs of $(a_d(j_m), f_i(x_i(a_d(j_m))))$, $m = 1, \ldots, s$ to the source subsystem.
- **Step 6.** After receiving the $s$ pairs of $(a_d(j_m), f_i(x_i(a_d(j_m))))$, $m = 1, \ldots, s$, from subsystems $i, i = 1, \ldots, l$, the source subsystem computes the sum $\sum_{i=1}^{l} f_i(x_i(a_d(j_m)))$ for each $m = 1, \ldots, s$. Determine $a_d(j_m)$ by (17) and send $a_d(j_m)$ to all subsystems.
- **Step 7.** Once receiving the good enough sub vector $a_d(j_m)$ from the source subsystem, all subsystems stop and output the solution $(x_i(a_d(j_m)), a_d(j_m))$.

4. Applications

With the increasing use of distributed intelligent devices and the demand of separated power network managing, distributed control of a large-scale power system becomes more and more important in real application. Decentralized optimal power flow with continuous and discrete control variables problems are important in power system research, and nervous solution methods have been proposed [27, 28]. In decentralized optimal power flow problems, the large-scale power system is separated to several subsystems. By coordinating the control of generators and taps in a subsystem, the cost of the network can be optimized. The decentralized optimal power flow problems with continuous and discrete control variables can be formulated a type of BACD problem stated below.

$$\min_{a_d} \sum_{i=1}^{l} f_i(x_i, a_d) = \sum_{j=1}^{b} \left[ \sum_{i=1}^{l} (\gamma_j + \phi_i + \phi_j) (\lambda_i, a_d) + \gamma_j \right]$$

(18a)

subject to:

$$g_i(x_i, a_d) = 0, \quad i = 1, \ldots, l$$

(18b)

$$h_i(x_i) \leq 0, \quad i = 1, \ldots, l$$

(18c)

$$a_d \in A_{d}, \quad i = 1, \ldots, l$$

(18d)

Fig. 2. Diagram of the IEEE 118-bus system partitioned into four subsystems.
where \( x_i \) denotes the continuous state and control variables vector; \( a_d \) denotes the discrete variable solution, such as switching shunt capacitor banks and transformer taps, come from the solution space set \( A_d \) and \( a_d \in A_d \); \( f_i(x_i, a_d) \) is a function of \( P_{G_i}(x_i, a_d) \) which denotes the objective function with minimum total distributed generation cost criterion of the subsystem \( i \), \( i = 1, \ldots, l \); \( P_{G_i}(x_i, a_d) \) is a function of \( x_i \) and \( a_d \) which denotes real power generation of the generation buses \( G_i \), \( G_i \) is the \( j \)th generation bus in subsystem \( i \); \( |J_i| \) denotes the cardinality of the set \( J_i \); the coefficients \( y_{ij}, \theta_{ij}, \varphi_{ij} \) of the generation cost curve are various for different generation bus \( P_{G_i} \). Note that \( g_i(x_i, x_j, a_d) = 0 \) denotes the flow balance constraints, such as real and reactive power mismatch, and \( h_i(x_i) \leq 0 \) denotes the functional inequality constraints, such as security constraints on line flows for specified limit [29]. The mathematical formulations shown in (18) belong to the type of BACD problem shown in (1a), (1b), (1c), (1d). Thus, we can use the proposed method to solve it.

The STOO algorithm is implemented in a deregulated environment using 4-PC network. We demonstrate the computational efficiency and the goodness of the obtained good enough solution by comparing with four heuristic methods: Genetic Algorithm (GA) [11], Tabu Search (TS) [12], Ant Colony Optimization (ACO) [13] and Simulated Annealing (SA) [14]. The STOO algorithm is employed to solve the BACD problem on two power systems, the IEEE 118-bus system partitioned into four subsystems (A1, A2, A3 and A4) shown in Fig. 2, and the IEEE 244-bus system partitioned into four subsystems (B1, B2, B3 and B4) shown in Fig. 3. Each subsystem is indicated by a closed-dash contour and associated with a PC.

Table 1 shows the number of buses, number of transmission lines and number of generation buses in each subsystem. It should be noted that the values of conductance of the transmission lines in the IEEE 244-bus system are much higher than that of the IEEE 118-bus system on average. The discrete variables include the switching capacity bank and transform tap ratio setting. Each switching capacitor is equipped with 4 capacitor banks, and the capacity of a bank is 14MVAR. In each transformer tap, there are 32 discrete steps so that each step is 5/8% of the nominal transformer tap ratio. Table 2 lists the four different settings of discrete variables for Cases (a)-(d) of the IEEE 118-bus system, Cases (A)-(D) of the IEEE 244-bus system, and the number of switching capacitors and transformers in each subsystem. Because each capacitor has four capacitor banks and each transform tap has 32 discrete steps, the corresponding sizes of the discrete solution of \( a_d \) in \( A_d \) are \( 4^{10} \times 32^{10}, 4^{18} \times 32^{18}, 4^{26} \times 32^{26}, \) and \( 4^{34} \times 32^{34} \) in Case (a), (b), (c) and (d), respectively. Besides, the corresponding sizes of the discrete solution of \( a_d \) in \( A_d \) are \( 4^{20} \times 32^{20}, 4^{46} \times 32^{46}, 4^{52} \times 32^{52}, \) and \( 4^{68} \times 32^{68} \) in Cases (A), (B), (C) and (D), respectively. Obviously, it is indeed computationally intractable to solve for a local optimal solution of the BACD problem using a global searching technique.

For each power system and discrete variable setting, we have tested 1000 simulation runs in each test case. The STOO algorithm is applied to solve the BACD problem of these eight cases in the 4-PC network. It should be noted that all the test results are simulated in a Pentium IV 2.66 GHz processor and 1.25 GB RAM. The program is implemented in Borland C++ and the TCP/IP is served.

**Table 1**

<table>
<thead>
<tr>
<th>System</th>
<th>IEEE 118-bus</th>
<th>IEEE 244-bus</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subsystems</td>
<td>A1</td>
<td>A2</td>
</tr>
<tr>
<td>Generator buses</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Buses</td>
<td>32</td>
<td>28</td>
</tr>
<tr>
<td>Transmission lines</td>
<td>49</td>
<td>33</td>
</tr>
</tbody>
</table>

**Fig. 3.** Diagram of the IEEE 244-bus system partitioned into four subsystems.
as the communication protocol in a 4-PC network. Subsystem A1 and B1 are assigned to be the source subsystems of the IEEE 118-bus and IEEE 244-bus systems, respectively. We set $\varepsilon = 10^{-4}$ in the parallel dual-type method and $s = 45$ for the cardinality of Selected Subset, SS [26].

Table 2 shows the final objective value and corresponding CPU time consumed in the 4-PC network in each case obtained by the proposed method and the four heuristic methods in the IEEE 118-bus and IEEE 244-bus systems. The CPU time is the amount of time starting from the communication overhead in the 4-PC network until the source subsystem determines the good enough discrete solution and sends the corresponding sub vector to each subsystem.

In addition, the four heuristic methods: GA, TS, ACO, and SA associated with exact model alone are also implemented to solve the same BACD problem in the same 4-PC network. These test results demonstrate that the computational efficiency of the STOO algorithm increases as the complexity of the test system increases. Fig. 4 displays the progressions of the objective value with respect to the CPU time for the STOO algorithm and the four heuristic methods in Case (C) of the IEEE 244-bus system. The average CPU time consumed by our algorithm is approximately 1.825 s, it can serve as an effective control process for real time application purpose. Obviously, the four heuristic methods associated with exact model alone are very time-consuming to

as the communication protocol in a 4-PC network. Subsystem A1 and B1 are assigned to be the source subsystems of the IEEE 118-bus and IEEE 244-bus systems, respectively. We set $\varepsilon = 10^{-4}$ in the parallel dual-type method and $s = 45$ for the cardinality of Selected Subset, SS [26].

Table 2 shows the final objective value and corresponding CPU time consumed in the 4-PC network in each case obtained by the proposed method and the four heuristic methods in the IEEE 118-bus and IEEE 244-bus systems. The CPU time is the amount of time starting from the communication overhead in the 4-PC network until the source subsystem determines the good enough discrete solution and sends the corresponding sub vector to each subsystem.

In addition, the four heuristic methods: GA, TS, ACO, and SA associated with exact model alone are also implemented to solve the same BACD problem in the same 4-PC network. These test results demonstrate that the computational efficiency of the STOO algorithm increases as the complexity of the test system increases. Fig. 4 displays the progressions of the objective value with respect to the CPU time for the STOO algorithm and the four heuristic methods in Case (C) of the IEEE 244-bus system. The average CPU time consumed by our algorithm is approximately 1.825 s, it can serve as an effective control process for real time application purpose. Obviously, the four heuristic methods associated with exact model alone are very time-consuming to

### Table 2

<table>
<thead>
<tr>
<th>Number of switching capacitors and transformers.</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IEEE 118-Bus Cases</td>
<td>(a)</td>
<td>(b)</td>
</tr>
<tr>
<td>Subsystems</td>
<td>A1</td>
<td>A2</td>
</tr>
<tr>
<td>Switching Capacitors</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Transformers</td>
<td>4</td>
<td>2</td>
</tr>
</tbody>
</table>

### Table 3

<table>
<thead>
<tr>
<th>Cases</th>
<th>IEEE 118-bus</th>
<th></th>
<th></th>
<th>IEEE 244-bus</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(a)</td>
<td>(b)</td>
<td>(c)</td>
<td>(d)</td>
</tr>
<tr>
<td>STOO algorithm</td>
<td>Time (s.)</td>
<td>0.648</td>
<td>0.649</td>
<td>0.650</td>
</tr>
<tr>
<td>Obj. ($/hr)</td>
<td>5790.53</td>
<td>5781.10</td>
<td>5769.26</td>
<td>12021.86</td>
</tr>
<tr>
<td>Tabu Search</td>
<td>Time (s.)</td>
<td>235</td>
<td>230</td>
<td>255</td>
</tr>
<tr>
<td>Obj. ($/hr)</td>
<td>6516.66</td>
<td>6245.32</td>
<td>6156.37</td>
<td>12098.69</td>
</tr>
<tr>
<td>O.E.F.</td>
<td>361.65</td>
<td>353.39</td>
<td>391.31</td>
<td>548.04</td>
</tr>
<tr>
<td>T.F.E.</td>
<td>0.1254</td>
<td>0.0803</td>
<td>0.0671</td>
<td>0.0563</td>
</tr>
<tr>
<td>Genetic Algorithm</td>
<td>Time (s.)</td>
<td>240</td>
<td>250</td>
<td>255</td>
</tr>
<tr>
<td>Obj. ($/hr)</td>
<td>6469.76</td>
<td>6267.87</td>
<td>6155.80</td>
<td>12347.65</td>
</tr>
<tr>
<td>O.E.F.</td>
<td>369.37</td>
<td>384.21</td>
<td>391.31</td>
<td>529.54</td>
</tr>
<tr>
<td>T.F.E.</td>
<td>0.1173</td>
<td>0.0842</td>
<td>0.0670</td>
<td>0.0271</td>
</tr>
<tr>
<td>Ant Colony Optimization</td>
<td>Time (s.)</td>
<td>255</td>
<td>235</td>
<td>250</td>
</tr>
<tr>
<td>Obj. ($/hr)</td>
<td>6392.75</td>
<td>6171.01</td>
<td>6122.72</td>
<td>12542.40</td>
</tr>
<tr>
<td>O.E.F.</td>
<td>392.52</td>
<td>361.10</td>
<td>383.62</td>
<td>535.71</td>
</tr>
<tr>
<td>T.F.E.</td>
<td>0.1040</td>
<td>0.0927</td>
<td>0.0630</td>
<td>0.0433</td>
</tr>
<tr>
<td>Simulated Annealing</td>
<td>Time (s.)</td>
<td>255</td>
<td>250</td>
<td>245</td>
</tr>
<tr>
<td>Obj. ($/hr)</td>
<td>6505.66</td>
<td>6274.23</td>
<td>6154.07</td>
<td>12330.82</td>
</tr>
<tr>
<td>O.E.F.</td>
<td>392.52</td>
<td>384.21</td>
<td>375.92</td>
<td>548.04</td>
</tr>
<tr>
<td>T.F.E.</td>
<td>0.1235</td>
<td>0.0851</td>
<td>0.0667</td>
<td>0.0257</td>
</tr>
</tbody>
</table>
obtain the optimal solutions. Since the proposed method consists of three stages and three models of performance evaluation, we can obtain a good enough solution with smaller objective value and consumed less CPU time than four heuristic methods. The efficiency of the STOO algorithm is significant. These comparisons demonstrate the computational efficiency of the STOO algorithm and the goodness of the obtained good enough solutions. The feasibility and the goodness of the obtained good enough solutions in all test cases confirm the robustness of the STOO algorithm.

Remark 1: It is difficult for most of the traditional constrained optimization methods to solve the problems involved discrete and integer variables. Various heuristic search approaches have been developed to solve these problems, such as Simulated Annealing (SA), Tabu Search (TS), Genetic Algorithm (GA), and Ant Colony Optimization (ACO). SA and TS are trajectory-based meta-heuristic methods with strong intensification, only a fraction of search space might be visited and there is a risk of being trapped in a local optimum. On the other hand, GA and ACO are population-based meta-heuristic techniques with strong diversification, the algorithms converge very slowly because solutions jump around some potentially optimal solutions. However, a big challenge in developing global optimization approaches is to compromise the contradictory requirements, including accuracy, robustness and computation time. It is difficult to meet all these requirements by concentrating on a sole heuristic search approach. To cope with the computationally intractable NP-hard optimization problems, we have proposed a method of embedding ST in OO to solve the considered problem. Since the neighbors of sub-optimal continuous variables will frequently result in a high probability of sub-optimal discrete variables, the sub-optimal values of continuous variables are rounded off to the nearest values of discrete variables. Then, the sensitivity theory based goal softening method is used to significantly reduce the search space. Thus, we can obtain a good enough solution using reasonable computation time.

Remark 2: Basically, the OO theory is not a method on its own but rather a supplement to existing optimization methods. Instead of finding the best solution, the OO theory concentrates on finding a good enough solution with high probability yielding a significantly reduced computation time. Thus, the bad side of the proposed method is that the obtained good enough solution is not exactly the global optimal solution. That is, the computation time can be reduced at the expense of global optimality. However, for most practical real-time applications in power systems, such as the decentralized optimal power flow control problems, we would rather obtain a good enough solution within reasonable computation time than get the global optimal solution using incredibly long time. Thus, compared to the exhaustive search methods associated with exact model alone to evaluate each solution, the proposed STOO firstly tends to select a good enough \(x_i^*(α/C^3)\) from the enormous \(A_i\), then yields the good enough solution \(x_j^*(α/C^3)\) using limited computation time.

5. Conclusion

This paper presented an algorithm of embedding sensitivity theory in ordinal optimization to solve a class of NP-hard problem. The proposed method was applied to solve numerous examples of decentralized optimal power flow with continuous and discrete variables problems on the IEEE 118-bus and 244-bus systems using a 4-PC network and compared the results with four heuristic approaches: GA, TS, ACO and SA methods. The STOO algorithm obtained a good enough solution with smaller objective value and consumed less CPU time than four heuristic methods. Test results demonstrated the validity, robustness and excellent computational efficiency of the STOO algorithm for obtaining a good enough solution.

Acknowledgments

This research work is supported in part by the National Science Council in Taiwan under Grant NSC100–2221E–324–006 and NSC100–2221–E–129–010.

References

[17] Horng SC, Lin SY. Ordinal optimization of G/G/1/K polling systems with k-
[18] Horng SC. Ordinal optimization based approach to the optimal resource
allocation of grid computing system. Math Comput Model 2011;54(1-
[19] Horng SC, Lin SS. Optimal cyclic service of the centralized broadband wireless
networks with k-limited discipline. Simul Model Pract Theory
[21] Bazaraa MS, Sherali HD, Shetty CM. Nonlinear programming: theory and
[22] Luenberger DC, Ye Y. Linear and nonlinear programming. 3rd ed. New
York: Springer; 2008.
[23] Lin CH, Lin SY, Lin SS. Improvements on the dual-type method used in solving
[24] Lin SS, Lin CH, Horng SC. A parallel dual-type algorithm for a class of quadratic
programming problems and applications. Expert Syst Appl
[25] Lin SS, Horng SC. A more general parallel dual-type method and application to
[26] Zhong J, Xie M, Wu FF. Ordinal optimization for power systems. In:
Proceedings of the 2006 IEEE power system conference and expos, Atlanta,
[27] Khodr HM, Martinez-Crespo J. Integral methodology for distribution systems
reconfiguration based on optimal power flow using Benders decomposition
distribution systems reconfiguration based on OPF and solved by
[29] Debs AS. Modern power systems control and operation. Boston: Kluwer