1997年度 定期總會 및 秋季學術大會 論文集 1997.11.29

최적조류계산의 분산처리기법에 관한 연구

김 호 웅° 김 발 호 김 정 훈
홍익대학교 전기제어공학과

An Approach to Implementing Distributed Optimal Power Flow

Kim, Ho-Woong° Kim, Bal-ho Kim, Jung-Hoon
Dept. of Electrical & Control Engineering. Hong-ik University

Abstract

This paper presents a mathematical approach to implementing distributed optimal power flow (OPF), wherein a regional decomposition technique is adopted to parallelize the OPF. Three mathematical decomposition coordination methods are introduced first to implement the proposed distributed scheme: the Auxiliary Problem Principle (APP), the Predictor-Corrector Proximal Multiplier Method (PCPM), and the Alternating Direction Method (ADM). Then two alternative schemes for modeling distributed OPF are introduced; the Dummy Generator-Dummy Generator (DGDG) scheme and Dummy Generator-Dummy Load (DGDL) scheme.

We present the mathematical analyses of the proposed approach, and demonstrate the approach on several test systems, including IEEE Reliability Test Systems and parts of the ERCOT (Electric Reliability Council of Texas) system.

1. Introduction

Conventionally, the OPF problem has been one of static optimizations involving a large-scale system, and formulated as a centralized continuous optimization problem. Recently, forces such as increasing competition, advances in generation technology, interests in deregulation, and the advent of new planning strategies have put pressure on electric utilities to become more efficient and to improve the efficiency from non-generation technologies such as supervisory control and data acquisition (SCADA). As a consequence, the role of OPF is changing and the importance for real-time computation, communication, and data control is greatly increasing.

As power systems are operated more closely to their ultimate ratings, it is becoming necessary to incorporate contingency constraints into the formulation, and more rapid updates of telemetered data and faster solution times are becoming necessary to better track changes in the system. The inclusion of security constraints greatly increases the computational difficulty of the OPF. However, the computational and communicational requirements for OPF are at the limit of current centralized implementations [1], and the requirement for faster and more frequent solutions has encouraged the recent development of a number of new OPF technologies, and the consideration of parallel implementations using decentralized processors.

2. Review of Parallel/Distributed OPF

Since Dantzig and Wolfe [2] proposed their decomposition principle for linear programming in 1960, an extensive work on large-scale mathematical programming has followed. (See [3] and its references.) Recently, motivated by this influential work, various approaches have been taken to parallelize power system problems including reactive power optimization problem and constrained economic dispatch problem [4, 5].

Initial applications of parallel computing to power systems problems used array computers which are equipped with specialized processors for performing vector computations efficiently [1]. Sundarraj et al. [6] demonstrated a distributed decomposition of constrained economic dispatch on a hypercube multiprocessor using Dantzig-Wolfe decomposition method.

While there has been some other works and progress in parallelizing power systems problems (see the discussion and references in [1]), major efforts has concentrated on parallelizing individual steps such as Jacobian factorization, and furthermore current implementations are centralized, making use of large mainframe computers.

In [7], Kim and Baldick proposed an approach to parallelizing optimal power flow (OPF) that is suitable for distributed implementation and is applicable to very large inter-connected power systems. In the approach, the OPF is solved in a decentralized framework, consisting of each region, a local processor would perform its own OPF for the region and its border. Regions interact by adjusting flows between regions depending on the prices quoted for inter-regional interchanges.

3. Distributed Optimal Power Flow

In our distributed scheme, we use the regional decomposition technique. This paper is an extension of [7]. In this paper, we introduce three mathematical decomposition coordination methods which are amenable for implementing distributed OPF: the Auxiliary Problem Principle (APP), the Predictor-Corrector Proximal Multiplier Method (PCPM), and the Alternating Direction Method (ADM). The theoretical background of the proposed methods and the formulation of distributed OPF will be first given in the following section. Then two implementation methodologies, called DGDG and DGDL schemes, proposed in our study will be described, followed by a case study.as in [7], where the regions buy and sell electricity from adjacent regions at prices that are coordinated by negotiations between adjacent regions. All the variables and constraints are the same as defined in [7].

3.1 OPF Formulation

With the same definition on variables and constraints as in [7], the OPF problem can be written as

$$\min_{(x,y)\in A,(y,z)\in B} \{ f_a(x) + f_b(z) \}, \tag{1}$$

where we assume that the cost functions f_a and f_b are convex approximations to the actual cost functions in each region and that there is a unique solution to (1). We decompose problem (1) into regions by duplicating the border variables and imposing coupling constraints between the two variables.

First, define the copies of y to be y_a and y_b, assigned to the regions a and b, respectively. Then the problem (1) is equivalent to:

$$\min_{\substack{(x,y_a)\in A\\(y_b,z)\in B}} \left\{ f_a(x) + f_b(z) + \frac{\gamma}{2}\|y_a - y_b\|^2 : y_a - y_b = 0 \right\}. \tag{2}$$

The quadratic term added to the objective does not affect the solution since the constraint $y_a - y_b = 0$ will make the quadratic term equal to zero at any solution; however, when we decompose the problem, this term will significantly aid in convergence.

3.2 Decomposition

Next we apply the three decomposition algorithms described in the previous section to obtain sub-problems for a distributed OPF implementation.

Algorithm APP [8]

First, with the use of auxiliary problem principle [8], we can solve (2) by solving a sequence of problems of the form:

$$(x^{k+1}, y_a^{k+1}) = \arg\min_{(x,y_a)\in A} \Big\{ f_a(x) + \frac{\beta}{2}\|y_a - y_a^k\|^2 + \gamma y_a^{\dagger}(y_a^k - y_b^k) + (\lambda^k)^{\dagger}(y_a) \Big\}, \tag{3}$$

$$(z^{k+1}, y_b^{k+1}) = \arg\min_{(y_b,z)\in B} \Big\{ f_b(z) + \frac{\beta}{2}\|y_b - y_b^k\|^2 - \gamma y_b^{\dagger}(y_a^k - y_b^k) - (\lambda^k)^{\dagger}(y_b) \Big\}, \tag{4}$$

$$\lambda^{k+1} = \lambda^k + \alpha(y_a^{k+1} - y_b^{k+1}), \tag{5}$$

where the superscript k is the iteration index, α and β are positive constants. Some sufficient conditions for this iterative scheme to converge to a solution of (2) are presented in [7]. The initial conditions $x^0, y_a^0, y_b^0, z^0, \lambda^0$ can be any convenient starting point such as a previous solution or flat start. The value of the Lagrange multiplier λ_i at iteration k is an estimate of the cost to maintain the constraint $y_{ai} - y_{bi} = 0$. If y_i represents, for example, power flow from region a to b along a particular line, then λ_i is the "shadow-cost" on the interchange of power along that line. If some region must import power to satisfy local demand, then the initial conditions for the border flows can be set to reflect the generation deficiency; however, this is not necessary for convergence since the dummy generators can be arranged to supply the imports necessary for a feasible initial solution.

Notice that the terms $\frac{\beta}{2}\|y_a - y_a^k\|^2 + \gamma y_a^T(y_a^k - y_b^k) + (\lambda^k)^T(y_a)$ in the objective of (3) can be interpreted as being the sum of cost functions of generators placed at the border buses in region-a. The cost functions of these *dummy generators* include costs for real and reactive power generation, voltage support, and phase angle control. A similar interpretation applies for the terms in (4). The costs are quadratic and depend on the values of the Lagrange multipliers as well as on previous values of the iterates.

Algorithm PCPM [11]

Similarly, employing the algorithm proposed by Chen [11], we obtain the following regional OPF problems:

$$(x^{k+1}, y_a^{k+1}) = \arg\min_{(x, y_a)\in A}\left\{f_a(x) + \frac{\beta_k}{2}\|y_a - y_a^k\|^2 + (\bar{\lambda}^{k+1})^T(y_a)\right\}, \quad (6)$$

$$(z^{k+1}, y_b^{k+1}) = \arg\min_{(y_b, z)\in B}\left\{f_b(z) + \frac{\beta_k}{2}\|y_b - y_b^k\|^2 + (\bar{\lambda}^{k+1})^T(y_b)\right\}, \quad (7)$$

$$\lambda^{k+1} = \lambda^k + \frac{1}{\beta_k}(Ax^{k+1} - z^{k+1}), \text{ (corrector step)} \quad (8)$$

$$\bar{\lambda}^{k+1} = \lambda^k + \frac{1}{\beta_k}(Ax^k - z^k), \text{ (predictor step)} \quad (9)$$

Algorithm ADM [16]

In the same manner, the Algorithm-ADM produces the following subproblems

$$(x^{k+1}, y_a^{k+1}) = \arg\min_{(x, y_a)\in A}\left\{f_a(z) + \frac{\gamma}{2}\|y_a - y_b^k\|^2 + (\lambda^k)^T y_a\right\}, \quad (10)$$

$$(z^{k+1}, y_b^{k+1}) = \arg\min_{(y_b, z)\in B}\left\{f_b(z) + \frac{\gamma}{2}\|y_b - y_a^k\|^2 + (\lambda^k)^T y_b\right\}, \quad (11)$$

$$\lambda^{k+1} = \lambda^k + \gamma(y_a^{k+1} - y_b^{k+1}). \quad (12)$$

3.3 Distributed Algorithm

A natural implementation of the Algorithm-APP and Algorithm-PCPM is given in Figure 1. It is noted that Algorithm-ADM requires the regional OPFs for region-a and region-b to be performed sequentially.

The Telemeter and Dispatch steps require intra-regional communication of data and control signals. The loop termination criterion requires global communication, while the Exchange step only requires communication between adjacent regions. In the case of multiple regions, each region will solve an OPF for its core and border variables.

Initialize x^0, y_a^0, y_b^0, z^0, λ^0 ;
$k = -1$;
Telemeter load and topology data from each region to its processor ;
Repeat {
  Increment k ;
  In parallel, solve the regional OPF for region-a and for region-b ;
  Exchange y_a^k and y_b^k between regional processors ;
  Update λ^{k+1} ;
    } Until y_a^k and y_b^k converge to within tolerance ;
Dispatch generators according to OPF solution.

Figure 1: Distributed implementation of parallel OPF.

4. Implementation of Distributed OPF

We formulate the problem by introducing hypothetical generating unis and loads which we call *Dummy generators* and *Dummy loads*, respectively. The dummy generators are designed to produce or consume electric power in accordance with the terms in the objective function of, for example, (3) or (4) that do not appear in the objective of (2). The dummy generators mimic the effects of the external part of the system through a cost for supply of real and reactive power, voltage support, etc. Dummy loads are used in an alternative way to implement Algorithm-ADM.

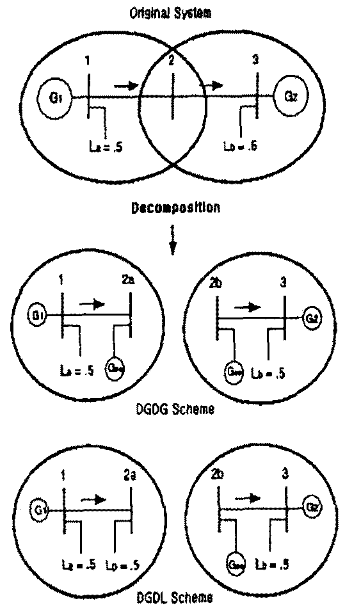

Figure 2 : Decomposition with DGDG and DGDL Schemes

In the following sections, two alternative models for distributed OPF are introduced. The first scheme puts two dummy generators for each transmission tie-line between a pair of regions. One of the dummy generators is placed in each region. We call this scheme *Dummy Generator- Dummy Generator* (DGDG) scheme. (Algorithm-APP and algorithm-PCPM.) In the second alternative, called *Dummy Generator-Dummy Load* (DGDL) scheme, one of the dummy generators in DGDG scheme is replaced with a dummy load. (Algorithm-ADM)

To impliment the distributed algorithm, we first duplicate the border buses, and the vector of the border variables, and then put either dummy generators or dummy loads on the duplicated border buses in each region. Each individual region solves an OPF that includes its own region and the borders it shares with other interconnected regions. The price information for the borders are then exchanged between adjacent regions for updating the multipliers on the constraints.

4.1 DGDG Scheme

In this scheme, dummy generators are put on the duplicated border buses in region-a and region-b to represent the power flow between a pair of interconnected regions. In the iterative algorithm, the generation levels of the dummy generators are determined by the updated cost functions and the demand-supply relationship. We interpret a positive output as injected power to the border bus, while negative output is interpreted as a demand at the border bus where the dummy generator is connected.

It is noted that the dummy generators in region-a and region-b are not both expected to produce or consume power at the same time; if a dummy generator in region-a produces power, then the corresponding dummy generator in region-b is supposed to produce negative power (equivalently, consume positive power) in equal amount at the same production cost so that the optimal solution to (2) is not affected by the dummy generators.

For instance, in Figure 2, assume that the dummy generator which placed on the border bus in region-a (Bus-2a) might produce negative power, while the dummy generator put on the border bus in region-b (Bus-2b) produces positive power, in order for the negative power and positive power to be canceled out at the optimum. In this case, with the dummy generator G_{DA}, region-a may reduce its core power generation due to the positive power from G_{DB}, resulting in a power flow from region-a to region-b.

4.2 DGDL Scheme

In this scheme, dummy generator(s) are put on the border buses in one region, while dummy load(s), set equal to the magnitude of the power output from the corresponding dummuy generator(s), are put on the border buses of the other region so that the power output from the dummy generator and the dummy load are canceled out at the optimum. As in DGDG scheme, the production cost functions of the dummy generators are updated in each iteration, using different update rule from that in DGDG scheme. In principle, DGDL scheme is best modeled as a sequential computation algorithm, though, under specific situations, it can be modeled as a parallel computation scheme.

In our implementation, first, the regional OPFs for region-a and region-b are executed with no dummy generator or dummy load. Then a dummy generator is put in the region experiencing higher Lagrange multipliers on the borders (e.g., region-b in our case study) to produce power. The iteration begins with the regional OPF for the region with the dummy generator, followed by the regional OPF for the region with the dummy load. The production cost function of the dummy generator is assumed to be:

$$F(P_{GD}) = \beta \cdot P_{GD} + \gamma_{GD} \cdot P_{GD}^2, \qquad (13)$$

where, P_{GD} is the power produced by the dummy generator, γ_{GD} is on the order of $1/N$ times the average of γ_{G_i} over all the generators in the region-a, given by,

$$\gamma_{GD} = \frac{1}{N^2} \sum_{i=1}^{N} \gamma_{G_i}, \qquad (14)$$

where N is the number of generators in region-a and γ_{G_i} is the quadratic cost coefficient of generator i. The detailed choice of γ_{GD} is problem dependent.

In each iteration, the coefficient β is updated by the following formula:

$$\beta^{k+1} = \frac{\lambda_a^k + \lambda_b^k}{2} - 2 \cdot \eta \cdot P_{GD}^k, \qquad (15)$$

where λ_a^k and λ_b^k are the Lagrange multipliers at the border buses in region-a and region-b, respectively, η is a problem dependent parmeter governing the rate of convergence.

To illustrate the implementation of DGDG and DGDL schemes, we present an example below.

4.3 Sample Application of the Schemes

The example system is given in Figure 2, which contains three buses, two generators meeting 1 pu of system demand. Demand is assumed to be real power for expositional convenience. In addition, all transmission lines are assumed to be lossless. The cost function for each generator is given by:

$$f_a = \frac{1}{2} P_a^2 \quad \text{(for generator } G_1\text{)} \qquad (16)$$

$$f_b = \frac{1}{2} P_b^2 \quad \text{(for generator } G_2\text{)} \qquad (17)$$

There is local demand in region-a of $L_a = 0.5$ at Bus-1 and local demand in region-b of $L_b = 0.5$ at Bus-3. Bus-2 is the border bus, Bus-1 is the core bus for region-a and Bus-3 is the core bus for region-b.

Problem (Centralized scheme):

We minimize the total production cost. For notational consistence with that in the preceding sections, we use x for P_a and z for P_b. Then the problem can be written as:

$$\min_{x+z=1} \left\{ \frac{1}{2} x^2 + z^2 \right\}. \qquad (18)$$

The optimum solution to this problem occurs at $x = \frac{2}{3}$ and $z = \frac{1}{3}$, yielding $\frac{1}{3}$ as the objective value. One may see that the generation at Bus-1 exceeds the local demand while at Bus-3 to satisfy demand-supply relationship (i.e., the equality constraint).

We will show that the same solution could be obtained with the implementation of our parallel computation schemes, DGDG (Algorithm-APP and Algorithm-PCPM) and DGDL (Algorithm-ADM).

Implementation of Algorithm-APP (DGDG Scheme)

Step 1 : Duplication

Duplicate the border bus (Bus-2) to get Bus-2a and Bus-2b, then put the dummy generator G_{DA} on the Bus-2a, and G_{DB} on Bus-2b. Introduce a border variable y, and duplicate it to yield y_a and y_b for region-a and region-b, respectively, where y can be interpreted as a power flow passing through the border bus (Bus-2). Consequently, the duplicated border variavles y_a and y_b pertain to the generation levels of the dummy generators.

Step 2 : Regional OPF

Divide the central optimization problem, (18), into two regional OPF problems as in Figure 2. The regional OPFs are given by:

$$OPF_a \quad \min_{x+y_a=L_a} \left\{ \frac{1}{2} x^2 + \frac{\beta}{2} \|y_a - y_a^k\|^2 + \gamma y_a^T (y_a^k - y_b^k) + (\lambda^k)^T (y_a) \right\} \qquad (19)$$

$$OPF_b \quad \min_{x+y_a=L_a} \left\{ z^2 + \frac{\beta}{2} \|y_b - y_b^k\|^2 + \gamma y_b^T (y_a^k - y_b^k) - (\lambda^k)^T (y_b) \right\} \qquad (20)$$

$$\lambda^{k+1} = \lambda^k + \alpha (y_a^{k+1} + y_b^{k+1}) \qquad (21)$$

Step 3 : Parallel computation

i) Solve OPF_a. The solutions x^{k+1} and y_a^{k+1} are given by,

$$x^{k+1} = \frac{\beta \cdot (L_a - y_a^k) + \gamma \cdot (y_a^k - y_b^k) - \lambda^{k+1}}{1 + \beta}$$

$$y_a^{k+1} = L_a - x^{k+1}$$

ii) Solve OPF_b. The solutions z^{k+1} and y_b^{k+1} are given by,

$$z^{k+1} = \frac{\beta \cdot (L_b - y_b^k) + \gamma \cdot (y_a^k - y_b^k) - \lambda^{k+1} + 2}{2 + \beta}$$

$$y_b^{k+1} = L_b - z^{k+1}$$

iii) Update λ.

$$\lambda^{k+1} = \lambda^k + \alpha (y_a^{k+1} - y_b^{k+1})$$

iv) Repeat i) to iii) until convergence criteria are met.

The results of the first few iterations are in Table 1, where α = .375, β = .750, γ = .375 were used.

Implementation of Algorithm-PCPM (DGDG Scheme)

Step 1 : Duplication

Duplicate the border bus (Bus-2) to get Bus-2a and Bus-2b, then put the dummy generator G_{DA} on the Bus-2a, and G_{DB} on Bus-2b as done in Algorithm-APP.

Step 2 : Regional OPF

Divide the central optimization problem, (18), into two regional OPF problems as in Figure 2. The regional OPFs are given by:

$$OPF_a \quad \min_{x+y_a=L_a} \left\{ \frac{1}{2} x^2 + \frac{1}{2\gamma} \|y_a - y_a^k\|^2 + (\lambda^k)^T (y_a) \right\} \qquad (22)$$

$$OPF_b \quad \min_{x+y_a=L_a} \left\{ z^2 + \frac{1}{2\gamma} \|y_b - y_b^k\|^2 - (\lambda^k)^T (y_b) \right\} \qquad (23)$$

$$\lambda^{k+1} = \lambda^k + \gamma (y_a^{k+1} + y_b^{k+1}) \qquad (24)$$

Step 3 : Parallel computation

i) Compute $\bar{\lambda}^{k+1}$. (Predictor Step.)

$$\bar{\lambda}^{k+1} = \lambda^k + \alpha (y_a^{k+1} - y_b^{k+1})$$

ii) Solve OPF_a. The solutions x^{k+1} and y_a^{k+1} are given by,

$$x^{k+1} = \frac{\gamma}{1+\gamma} \left\{ \frac{1}{\gamma} \cdot (L_a - y_a^k) - \bar{\lambda}^{k+1} \right\}$$

$$y_a^{k+1} = L_a - x^{k+1}$$

iii) Solve OPF_b. The solutions z^{k+1} and y_b^{k+1} are given by,

$$z^{k+1} = \frac{\gamma}{1+2\gamma} \left\{ \frac{1}{\gamma} \cdot (L_b - y_b^k) + \bar{\lambda}^{k+1} + 2 \right\}$$

$$y_b^{k+1} = L_b - z^{k+1}$$

iv) Update λ. (Corrector Step.)

$$\lambda^{k+1} = \bar{\lambda}^k + \gamma (y_a^{k+1} - y_b^{k+1})$$

v) Repeat i) to iv) until convergence criteria are met.

The results with first few iterations are in Table 2, where stage-fixed γ = .525 was used.

Implementation of Algorithm-ADM (DGDL Scheme)

Step 1 : Duplication

Duplication the border bus (Bus-2) into Bus-2a and Bus-2b, and put a dummy generator on the Bus-2b. (It has been already known that the dual value at Bus-2b is higher than that at Bus-2a.) Notice that no border variable is introduced in this case.

Step 2 : Regional OPF

Divide the master problem, (18), into two subproblems as done in DGDG scheme. The regional OPFs are then given by;

$$OPF_a \quad \min_{\{x=L_a+y_a\}} \left\{ \frac{1}{2} x^2 \right\} \qquad (25)$$

$$OPF_b \quad \min_{\{z+y_b=L_b\}} \left\{ z^2 + \beta \cdot y_b + \gamma_{GD} \cdot y_b^2 \right\} \qquad (26)$$

Step 3 : Parallel computation

i) Slove OPF_b. The solutions z^{k+1} and y_b^{k+1} are given by,

$$z^{k+1} = \frac{\beta^{k+1} + 2\gamma_{CD} \cdot L_b}{2 + 2\gamma_{CD}}$$

$$y_b^{k+1} = L_b - z^{k+1}$$

ii) Solve OPF_a. The solution x^{k+1} is given by,

$$x^{k+1} = L_a + y_b^k$$

iii) Update β.

$$\beta^{k+1} = \frac{\lambda_a^k + \lambda_b^k}{2} - d \cdot P_{CD}^k$$

iv) Repeat i) to iii) until convergence critera are met.

The results for the first few iterations and the final solutions are given in Table 3. For this simple problem the DGDL scheme has better convergence property than the DGDG scheme. However, the convergence rate depends strongly on the choice of parameters, α, β, γ, and the characteristics of the problems. In the following section we show results for DGDG and DGDL applied to OPF problems.

5. Case Sudies

In this section, several case studies are performed to demonstrate the proposed distributed OPF algorithms. The objectives of the case studies are, first, to discover the viability of the algorithms in practical implementation and, second, to test and compare the overall performance of the algorithms. Performance comparisons are based on the cputimes and number of iterations required for desired accuracy.

For the case studies, a state-of-the art Interior-Point OPF code (INTOPF) [19] were employed. Non-contingency constrained AC OPFs were performed for all cases with real and reactive generator limits and line and voltage constraints imposed. All computations were performed on a Sun Sparc-20 workstation, while parallel (distributed) computations with INTOPF code were implemented on several Sparc-20 and Ultra-Sparc workstations.

Table 1 : Algorithm-APP (DGDG Scheme)

k	x	z	y_a	y_b	λ^{k+1}	λ_a^k	λ_b^k	$\Vert y_a - y_b \Vert$
0	.5000	.5000	.0000	.0000	.7500	.5000	1.0000	.0000
1	.6428	.4090	-.1428	.0909	.7306	.6428	.8181	.0519
2	.6818	.3701	-.1818	.1298	.7110	.6818	.7402	.0519
3	.6873	.3524	-.1873	.1475	.6961	.6873	.7048	.0398
4	.6838	.3438	-.1838	.1561	.6857	.6838	.6876	.0276
5	.6790	.3393	-.1790	.1606	.6788	.6790	.6787	.0183
12	.6667	.3333	-.1667	.1667	.6667	.6667	.6667	.0000

Table 2 : Algorithm-PCPM (DGDG Scheme)

k	x	z	y_a	y_b	λ^{k+1}	λ_a^k	λ_b^k	$\Vert y_a - y_b \Vert$
0	.5000	.5000	.0000	.0000	.7500	.5000	1.0000	.0000
1	.6428	.4090	-.1428	.0909	.6915	.6428	.8181	.0519
2	.6595	.3559	-.1595	.1440	.6741	.6595	.7119	.0155
3	.6645	.3400	-.1645	.1599	.6688	.6645	.6801	.0046
4	.6660	.3353	-.1660	.1646	.6673	.6660	.6707	.0013
5	.6664	.3339	-.1664	.1660	.6668	.6664	.6678	.0004
9	.6666	.3333	-.1666	.1666	.6666	.6666	.6666	.0000

Table 3 : Algorithm-ADM (DGDL Scheme)

k	x	z	y_a	y_b	λ_a^k	λ_b^k	β^{k+1}	$\Vert \lambda_a^k - \lambda_b^k \Vert$
0	.5000	.5000	.0000	.0000	.5000	1.0000	.7500	.5000
1	.5714	.4285	.0714	.0714	.5714	.8571	.6071	.2857
2	.6122	.3877	.1122	.1122	.6122	.7755	.5255	.1633
3	.6355	.3644	.1355	.1355	.6357	.7286	.4786	.0932
4	.6489	.3511	.1489	.1489	.6489	.7022	.4522	.0533
5	.6565	.3434	.1565	.1565	.6565	.6869	.4369	.0304
11	.6666	.3333	.1666	.1666	.6666	.6666	.4166	.0000

Table 4 : Case study systems.

No	Buses	Regions	Core Buses	Ties	lines	load
1	50	2	24,24	2	80	50
2	78	3	24,24,24	6	126	74
3	108	4	24,24,24,24	12	186	100
4	238	2	118,118	2	376	76
5	360	3	118,118,118	6	570	126
6	376	2	271,105	3	574	157
7	753	4	271,,105,128,237	12	1100	209
8	1459	6	271,105,128,237,,365,325	28	2145	395
9	1777	8	271,105,128,237,365,325,74,213	59	2581	462

5.1 Case Study Systems

Data from two IEEE Reliability Test System and eight Texas utilities were used to demonstrate the performance of the algorithm. Table 4 summarizes the test systems. The first column denotes the system identification number, which will be used throughout the paper instead of real names, the second column shows the total number of buses in each system, while the third and fourth columns show the number of regions and the number of core buses in each region. The fifth column shows the number of tie-lines that interconnect the regions, while the sixth column shows the total number of transmission lines in each complete system. The last column shows the total per unit loads in the systems. The five smaller systems consist of two, three, or four copies of two IEEE Test Systems, while the four Texas systems use data from two to eight Texas utilities.

The objective to be minimized is the production cost for active and reactive power. The cost of reactive power is assumed to be 10^{-6} of the active power cost for each generator, while real power costs were adopted from [20] and [21]. The constants α, β, and γ were tuned for each system to improve convergence.

5.2 Stopping criterion

We chose the maximum mismatch between the border variables as the stopping criterion. To select the tolerance on the maximum mismatch, we experimented with the performance of the algorithm. We found that the choice 0.03 per unit maximum mismatch yielded a solution with total costs that were within 0.1% of the optimal production costs from the serial algorithm. Typically, the mismatches on most buses were much smaller than 0.03 per unit.

5.3 Test Results

Selected case study results are presented in this section. To compare the overall performance of the algorithms, the total cputimes and iteration counts are tabulated. Several figures based on the results from the Algorithm-APP are also provided. Finally, the speed-ups and efficiency of the Algorithm-APP are discussed.

The cputime results from the undecomposed and the parallel implementation of INTOPF code are summarized in Tables 5 and 6, respectively, where all the cputimes include the overheads necessary for reading data and communicating among processors. As seen in Table 5, the cputimes and the number of buses have almost a linear relationship. Table 6 shows that the first iteration of the INTOPF algorithm takes much more cputime than each subsequent iteration.

Table 7 compares the estimated efficiencies of the algorithms.

6. Conclusion

6.1 Distributed OPF issues

We have presented an effective parallel algorithm that can achieve significant speed-up over serial implementations. In a distributed environment there are overheads that may reduce the possible speed-up. However, even if speed-ups of the OPF computation itself were less than ideal, there would still be three powerful incentives to explore a distributed implementation. First, as we have remarked, institutional arrangemaents may prevent the pooling of data.

Second even if pooling of data were possible, communication bottlenecks at a central control center may prove a major obstacle for centralized multi-utility OPF. For real-time applications, particularly, a distributed implementation using our approach will therefore be much more attractive than a central implementation.

We note that most traditional approaches to parallelizing OPF involve a master process assigning tasks to slave processes. Telemetered data is passed from the master process to the assigned slave process, making communication overhead heavy for distributed implementation. For this reason, the traditional approaches are unlikely to be practical for on-line applications.

A distributed implimentation has a third important advantage over a centralized implimentation (whether serial or parallel.) A communication failure between regions can be handled more gracefully by a group of decentralized processors than in a centralized implementation because each regional processor can attend to the local needs of its region, perhaps with increased generation costs, even while inter-regional communication is interrupted.

Table 5 : Cputime for undecomposed OPF with INTOPF (sec)

System Number	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9
Base case	1.9	2.4	4.2	7.2	11.7	17.6	37.3	66.2	89.5

Table 6 : Cumulative cputime for parallel OPF with INTOPF (sec)

System Number	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9
Iteration=1	1.4	1.4	1.4	1.7	1.7	6.5	6.5	7.2	7.2
Iteration=5	2.2	2.3	2.2	3.3	3.2	11.4	11.3	12.3	12.7
Iteration=10	3.1	3.1	3.2	5.2	5.4	17.7	18.1	18.9	19.3
Iteration=20	5.1	5.0	5.1	8.7	8.7	30.1	30.3	31.6	32.6

Table 7 : Comparison of Efficiency (%)

System Number	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9
Algorithm-APP	20.7	24.2	35.0	69.2	76.5	53.3	61.7	65.3	57.9
Algorithm-PCPM	18.9	22.5	37.2	66.3	71.4	51.7	59.5	59.2	54.5
Algorithm-ADM	20.2	25.1	36.2	71.8	72.7	52.5	61.5	62.6	55.4

6.2 Algorithms for Implimenting Distfibuted OPF

In this paper, three decomposition algorithms based on the augmented Lagrangian method were introduced to implement the distributed OPF, namely Algorithm-APP, Algorithm-PCPM, and Algorithm-ADM, respectively.

In addition, to formulate the regional OPF problem, two alternative models are introduced. The first scheme, called DGDG scheme, puts two dummy generators for each transmission tie-line between a pair of regions. One of the dummy generators in DGDG scheme is replaced with a dummy load. Algorithm-ADM was formulated with this scheme. Note that in Algorithm-ADM one can employ dummy generators instead of dummy loads, but our experience shows that in the problems where many tie-lines exist, adopting dummy loads makes programming easier and shows better convergence property in general.

Based on the case study results, Algorithm-APP has a great advantage in number of iterations, while Algorithm-ADM looks very competitive in cputime. However, in an efficient implimentation, we would expect the time per itertion to be almost independent of the algorithms. Therefore Algorithm-APP, which requires fewer iterations to converge, can be expected to perform better overall than Algorithm-PCPM and Algorithm-ADM.

6.3 Direction of Future Study

Our future study is first to explore ways to improve convergence of the algorithms. The critical issue is then how many iterations are necessary before the Lagrange multipliers and border variables converge. The quadratic term introduced in (2) and approximated in (3), (4) is designed to tie the copies of the border variables together more strongly than just using a linear constraint in (2). The reason is that the quadratic term strongly convexifies the problem. The effect is to enhance the rate of convergence. An important challenge is to theoretically analyze the improvement in convergence speed due to the quadratic term. Clearly, careful choice of regions will also enhance the convergence of the algorithm. Since the inter-regional communication requirements will be relatively small under essentially any choice of regional decomposition, the main goal in choosing the regional decomposition will be to enhance convergence.

Finally, incorporation of contingency constraints will also be be studied. We will investigate ways to represent security constraints and to solve the SCOPFs efficiently and reliably in distributed manner.

References

[1] Task Force of the Computer and Analytical Methods Subcommittee of the Power Systems engineering Committee. Parallel processing in power systems computation. *IEEE Transactions on Power Systems*, 7(2):629-637, May 1992.

[2] G. B. Dantzig and P. Wolfe. Decomposition principle for linear programs. *Operations Research*, 8, January 1960.

[3] M. D. Mesarovic, D. Macko, and Y. Takahara. *Theory of Hierarchical, Multilevel, Systems*. Mathematics in science and engineering. Academic Press, New York, 1970.

[4] N. I. Deeb and S. M. Shahidehpour. Decomposition approach for minimizing real power losses in power systems. *IEE Proceedings, Part C*, 138(1):27-38, January 1991.

[5] Nedal. I. Deeb and S. M. Shahidehpour. Linear reactive power optimization in a large power network using the decomposition approach. *IEEE Transaction on Power Systems*, 5(2):428-438, May 1990.

[6] R.P. Sundarraj, S. Kingsley Gnanendran, and J.K. Ho. Distributed price-directive decomposition applications in power systems. *Paper 94 SM 596-7 PWRS presented at the IEEE Power Engineering Society 1994 Summer Meeting, San Francisco, CA, July 15-19*, 1994.

[7] Balho H. Kim and Ross Baldick. Coarse-grained distributed optimal power flow. *Paper 96 SM 564-5 PWRS presented at IEEE/PES Summer Meeting, Denver, Colorado, July 28-August 1*, 1996.

[8] Guy Cohen. Optimization by decomposition and coordination: A unified approach. *IEEE Transactions on Automatic Control*, AC-23(2):222-232, April 1978.

[9] Guy Cohen. Auxiliary problem principle and decomposition of optimization problems. *Journal of Optimization Theory and Applications*, 32(3):277-305, November 1980.

[10] Guy Cohen and Dao Li Zhu. Decomposition coordination methods in large scale optimization problems. *Advances in Large Scale Systems*, 1:203-266, 1984.

[11] Gong Chen. *Proximal and Decomposition Method in Convex Programming*. Ph.D thesis, University of Maryland Baltimore, 1993.

[12] J. Eckstein. Parallel alternating direction multiplier decomposition of convex programs. *Journal of Optimization Theory and Applications*, 80(1):39-63, January 1994.

[13] D. Garby and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite-element approximation. *Comp. Math. Appl.*, 2:17-40, 1976.

[14] E. V. Tamminen. Sufficient conditions for the existence of multipliers and Lagrangian duality in abstract optimization problems. *Journal of Optimization Theory and Applications*, 82(1):93-104, July 1994.

[15] P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal on Control and Optimization*, 29(1):119-138, January 1991.

[16] J. Eckstein and P.D. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55(3):293-318, 1992.

[17] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, New Jersey, 1970.

[18] Anthony Brooke, David Kendrick, and Alexander Meeraus. *GAMS User's Guide*. The Scientific Press, Redwood City, CA, 1990.

[19] Yu-Chi Wu, Atif S. Debs, and Roy E. Marsten. A direct nonlinear predictor-corrector primal-dual interior point algorithm for optimal power flows. *IEEE Transactions on Power Systems*, 9(2):876-883, May 1994.

[20] Allen J. Wood and Bruce F. Wollenberg. *Power Generation, Operation, and Control*. Wiley, New York, 2nd edition, 1996.

[21] Martin L. Baughman et al. *Electric Utility Resource Planning and Production Costing Projects: Final Report*. Center for Energy Studies, The University of Texas at Austin, 1993.