

PC 클러스터 시스템 기반 병렬 PSO 알고리즘의 최적조류계산 적용

論 文
56-10-1

Application of Parallel PSO Algorithm based on PC Cluster System for Solving Optimal Power Flow Problem

金鍾律^{*} · 文景俊^{*} · 李和錫^{**} · 朴俊灝^{***}
(Jong-Yul Kim · Kyoung-Jun Moon · Haw-Seok Lee · June-Ho Park)

Abstract - The optimal power flow(OPF) problem was introduced by Carpentier in 1962 as a network constrained economic dispatch problem. Since then, the OPF problem has been intensively studied and widely used in power system operation and planning. In these days, OPF is becoming more and more important in the deregulation environment of power pool and there is an urgent need of faster solution technique for on-line application. To solve OPF problem, many heuristic optimization methods have been developed, such as Genetic Algorithm(GA), Evolutionary Programming(EP), Evolution Strategies(ES), and Particle Swarm Optimization(PSO). Especially, PSO algorithm is a newly proposed population based heuristic optimization algorithm which was inspired by the social behaviors of animals. However, population based heuristic optimization methods require higher computing time to find optimal point. This shortcoming is overcome by a straightforward parallel processing of PSO algorithm. The developed parallel PSO algorithm is implemented on a PC cluster system with 6 Intel Pentium IV 2GHz processors. The proposed approach has been tested on the IEEE 30-bus system. The results showed that computing time of parallelized PSO algorithm can be reduced by parallel processing without losing the quality of solution.

Key Words : OPF, PSO Algorithm, PC Cluster System, Parallel Processing

1. 서 론

최적조류계산(Optimal Power Flow : OPF)은 전력계통에서 여러 가지 제약조건을 만족하면서 경제적이고 안전하게 계통을 운용하기 위한 기법이다. 1960년대 초 Dommel과 Tinney[1]에 의해 소개된 이후 많은 연구자들에 의해 연구되고 있는 전력계통의 중요한 분야이며, 최근에는 컴퓨터 기술의 발달에 힘입어 더 많은 연구가 진행되고 있다. 현재 우리나라의 전력산업 환경은 구조 개편으로 인해 경제적, 사회적, 그리고 기술적으로 많은 변화를 겪고 있다. 이러한 변화 속에서 안전하고 높은 신뢰도를 유지하면서 최적으로 계통을 운용하는 것은 점차로 어려운 일이 되고 있다. 따라서 전력계통의 운용과 계획에 가장 기본이 되고 필수적인 최적조류계산에 대한 연구가 요구되고 있으며, 이와 아울러 최적조류계산 기법의 온라인 적용에 대한 필요성 또한 높아지고 있다. 종래의 계산방법에는 비선형 계획법[2], 선형계획법[3], 2차계획법[4] 등의 수치적인 기법 등이 사용되었다. 그러나 이러한 방법들이 전역 최적해를 구하기 위해서는 목적함수가 Convex해야 하나, 최적조류계산 문제는 Non-Convex한

특성을 가지므로 기존의 방법으로 전역 최적해를 구하기가 어려운 단점이 있다. 최근들의 수치적 기법의 단점을 극복하고자 유전알고리즘(GA)[5][6], 진화프로그래밍(EP)[7]과 같은 진화연산 기법들이 최적조류계산 문제에 적용되어 좋은 성능을 보이고 있다. 최근에는 새로운 진화연산 기법의 일종인 PSO(Particle Swarm Optimization) 알고리즘[8]의 적용에 대한 연구가 진행되고 있는데, PSO 알고리즘은 기존 유전알고리즘에 비해 연산과정이 간단하여 계산부담이 상대적으로 적고, 최적해 탐색능력에 영향을 미치는 파라미터가 적어 실제 구현이 용이하다는 장점을 가지고 있다. 그러나, 진화연산기법은 전역 최적해에 대한 우수한 탐색능력을 가지고 있으나, 한 점에서 해를 찾아가는 것이 아니라 여러 개의 점에서 동시에 해를 찾아나가기 때문에 상대적으로 많은 연산시간이 걸린다는 단점이 있다. 특히, 진화연산기법을 이용한 최적조류계산 알고리즘의 온라인 적용을 위하여는 연산시간의 단축은 더욱 중요한 요소로 부각되어진다. 병렬처리 기법은 여러 개의 프로그램들 또는 프로그램의 분할된 부분들을 다수의 프로세서가 분담하여 동시에 처리하는 기술을 말하며, 단일 컴퓨터에 의해 순차적으로 수행하는 것보다 속도를 크게 향상시킬 수 있다. 기존의 병렬 알고리즘 구현은 주로 병렬 컴퓨터를 이용하였으나 병렬 컴퓨터는 매우 고가여서 쉽게 이용할 수 없고, 확장이 용이하지 않다는 단점이 있었다. 이에 비해 PC 클러스터 시스템은 다수의 저렴한 일반 PC들을 고속 네트워크로 연결하여 마치 하나의 컴퓨팅 시스템처럼 사용함으로써 병렬처리 능력을 가진다. 따라서 PC 클러스터 시스템은 기존의 병렬 컴퓨터와 비교해 저가이며, 추후 확장이 용이하다는 장점이 있다 [9].

[†] 교신저자, 正會員 : 한국전기연구원 전력연구단 선임연구원
E-mail : jykim@keri.re.kr
^{*} 正會員 : 한국원자력연구원 선임연구원
^{**} 正會員 : 거제대학교 전기과 부교수
^{***} 正會員 : 부산대학교 전자전기통신공학부 교수
接受日字 : 2007年 5月 17日
最終完了 : 2007年 8月 13日

OPF 문제에 병렬처리 기법을 적용한 최초 연구는 1981년 Romeo에 의해 수행되었다[10]. 이 연구에서는 여러개의 독립된 지역으로 구성된 계통의 경제급전문제를 병렬계산을 이용하여 해결하도록 하였다. 이후 Wu와 Bose[11], Sundarraj[12] 등을 필두로 많은 연구가 진행되었다. 그러나 이러한 연구들은 주로 Jacobian Factorization을 병렬처리 하는데 목적을 두고 있어 실질적인 병렬계산의 효용을 얻지 못하였다. [13]에서 Balho Kim과 Ross Baldick은 지역분할 기법을 응용한 분산처리 기법을 제시하였는데, 이 방법은 종래의 방법처럼 수학적으로 연산부분을 나누는 것이 아니라 지역에 따라 나누어 분산처리 하게 하였다.

진화연산 기법을 이용한 최적화 문제 역시 연산시간 단축을 위해 병렬처리 기법 적용에 대한 연구가 진행되고 있는데, Tanese와 Schonkwiler는 [14],[15]에서 병렬 GA 알고리즘을 이용하여 연산시간을 단축하는 기법을 제시하였으며, 이후 [16],[17]에서는 전력계통의 기동정지 계획 및 경제급전 문제에 적용하여 그 효용성을 입증하였다. 또한, Li와 Wada는 [18]에서 병렬 EP 알고리즘을 OPF 문제에 적용하여 그 효과를 검토하였다.

본 논문에서는 최적해를 찾는 데 소요되는 연산시간을 단축시키기 위해 PC 클러스터 기반 병렬 PSO 알고리즘을 이용한 최적조류계산 기법을 제안하였다. 제안한 기법의 유용성을 보이기 위하여 IEEE 30 모선 계통의 최적조류계산 문제에 적용하였으며, 검토결과 최적해 탐색성능의 저하없이 연산시간을 단축할 수 있음을 확인하였다.

2. 최적조류계산

2.1 최적조류계산의 정의

최적조류계산은 제어변수를 통하여 계통운용과 관련된 제약조건들을 만족시키면서, 전력계통 운용 시 고려해야 하는 목적함수를 최적화하는 문제로써 다음과 같이 나타낼 수 있다.

$$\begin{aligned} \text{Minimize } & f(x, u) & (1) \\ \text{s.t. } & g(x, u) = 0 & (2) \\ & h(x, u) \leq 0 & (3) \end{aligned}$$

여기서, x : 상태변수

- 전압의 위상각
- 부하 모선에서의 전압 크기
- 발전기의 무효전력 출력

u : 제어변수

- 발전기의 유효전력 출력(Slack Bus 제외)
- 발전기 모선 전압
- 변압기의 변압비

$f(\cdot)$: 목적함수

$g(\cdot)$: 등식제약조건

$h(\cdot)$: 부등식제약조건

2.2 최적조류계산 문제의 정식화

본 논문에서는 유효발전 비용의 최소화를 최적조류계산 문제의 목적함수로 설정하였으며, 이를 다음의 식 (4)에 나타내었다.

$$\text{Min } F(P_g) = \sum_{i=1}^{N_g} (a_i + b_i P_{gi} + c_i P_{gi}^2) \quad (4)$$

여기서, $F(P_g)$: 총 발전 비용

P_{gi} : 각 발전기의 유효 전력출력

i : 발전기의 모선 번호

N_g : 계통 전체의 발전기 모선의 수

a_i, b_i, c_i : 각 발전기의 연료계수

최적조류계산에서의 제약조건은 다음과 같다.

1) 등식 제약조건

$$P_i - P_{gi} + P_{di} = 0 \quad i = 1, \dots, N_b \quad (5)$$

$$Q_i - Q_{gi} + Q_{di} = 0 \quad i = 1, \dots, N_b \quad (6)$$

$$P_i = V_i \sum_{j=1}^{N_b} V_j Y_{ij} \cos(\theta_i - \theta_j - \psi_i) \quad i = 1, \dots, N_b \quad (7)$$

$$Q_i = V_i \sum_{j=1}^{N_b} V_j Y_{ij} \sin(\theta_i - \theta_j - \psi_i) \quad i = 1, \dots, N_b \quad (8)$$

여기서, i : 발전기의 모선 번호

P_i : i번째 모선에 주입되는 유효전력

Q_i : i번째 모선에 주입되는 무효전력

P_{gi}, Q_{gi} : i번째 발전기 유효전력 및 무효전력

P_{di}, Q_{di} : i번째 부하 유효전력 및 무효전력

θ_i, θ_j : i와 j번째 모선 전압 위상각

N_b : 모선 총 개수

ψ_{ij} : 모선 i와 j의 상호 어드미턴스 위상각

Y_{ij} : 모선 i와 j의 상호 어드미턴스 크기

2) 부등식 제약조건

i) 운용상의 제약조건

- 선로 조류 제약 (MVA, MW, A)

$$\sqrt{P_i^2 + Q_i^2} \leq S_i^{Max} \quad (9)$$

여기서, P_i : i번째 모선에 주입되는 유효전력

Q_i : i번째 모선에 주입되는 무효전력

S_i^{Max} : i번째 모선에 흐르는 복소전력의 최대허용값

용값

- 부하 모선의 전압크기

$$V_{di\min} \leq V_{di} \leq V_{di\max} \quad (10)$$

여기서, V_{di} : 부하모선의 전압 크기

$V_{di\min}, V_{di\max}$: 부하모선 허용 전압의 최소값
과 최대값

- 발전기의 무효전력 출력용량

$$Q_{gi\min} \leq Q_{gi} \leq Q_{gi\max} \quad (11)$$

여기서, Q_{gi} : 발전기의 무효전력 출력

$Q_{gi\min}, Q_{gi\max}$: 발전기의 무효전력 최소 및 최
대 출력값

- 슬랙 모선의 유효전력 출력

$$P_{g1\min} \leq P_{g1} \leq P_{g1\max} \quad (12)$$

여기서, P_{g1} : 슬랙 모선의 유효전력 출력

$P_{g1\min}, P_{g1\max}$: 슬랙 모선의 유효전력 최소
및 최대 출력값

ii) 제어변수에 관한 제약조건

- 발전기의 유효전력 출력용량

$$P_{gi\min} \leq P_{gi} \leq P_{gi\max} \quad (13)$$

여기서, P_{gi} : 발전기의 유효전력 출력

$P_{gi\min}, P_{gi\max}$: 발전기의 유효전력 최소 및
최대 출력값

- 발전기 모선의 전압크기

$$V_{gi\min} \leq V_{gi} \leq V_{gi\max} \quad (14)$$

여기서, V_{gi} : 발전기 모선의 전압 크기

$V_{gi\min}, V_{gi\max}$: 발전기 모선 전압 최소 및
최대값

- 변압기의 탭 비

$$t_{i\min} \leq t_i \leq t_{i\max} \quad (15)$$

여기서, t_i : 변압기 탭 비

$t_{i\min}, t_{i\max}$: 변압기 탭 비의 최소 및 최대값

- 전력용콘덴서 투입량

$$Sh_{i\min} \leq Sh_i \leq Sh_{i\max} \quad (16)$$

여기서, Sh_i : 전력용콘덴서 투입량

$Sh_{i\min}, Sh_{i\max}$: 전력용콘덴서 투입량 최소 및
최대값

3. PC 클러스터 시스템 기반 병렬 PSO 알고리즘

3.1 PC 클러스터 시스템

1980년대 중반이후 과학과 공학 분야의 발전을 통하여 첨단과학기술과 정보 통신 분야 등의 연구에 있어서 고계산성 컴퓨터의 요구가 증가되고 있다. 기존의 벡터형 슈퍼컴퓨터는 고가이므로 접하기 어려울 뿐만 아니라 발전도 한계를 보이고 있다. 그러나 최근 고성능 네트워크 기술의 비약적인 발전과 반도체 집적기술의 향상으로 클러스터형 슈퍼컴퓨터로 바뀌고 있는 추세이다. PC 클러스터 시스템이란 다수의 PC 또는 workstation을 고속 네트워크로 연결하여 하나의 컴퓨팅 시스템으로 사용하는 것으로 고성능 또는 고가용성을 얻을 수 있도록 하는 기술을 말한다. 이러한 클러스터 시스템은 일반 PC를 이용함으로써 저렴한 가격에 쉽게 구입하여 이용할 수 있으며, 기존의 병렬형 슈퍼컴퓨터보다 수배에서 수십 배 작은 비용으로 동일한 성능의 시스템 구성이 가능하여 가격 대 성능비가 우수하다. 그리고 사용자가 직접 상용부품을 사용하여 업그레이드나 node의 확장이 가능하여 시스템 유지비용이 감소하고 사용이 편리한 PC의 개발환경을 그대로 사용할 수 있다는 장점이 있다. PC 클러스터 시스템은 각 node를 연결하여 SAN(system area network)을 구축하기 위한 네트워크 어댑터와 병렬 프로그래밍을 지원하기 위한 통신 라이브러리로 구성된다. 그림 1은 일반적인 클러스터 시스템의 구조이다. PC 클러스터 시스템의 성능은 각 node를 연결하는 네트워크의 성능, 지원 소프트웨어와 각 node의 성능에 의해서 좌우되므로 PC 클러스터 시스템 구성 시 각 구성 부분을 적절히 선정하는 것이 중요하다.

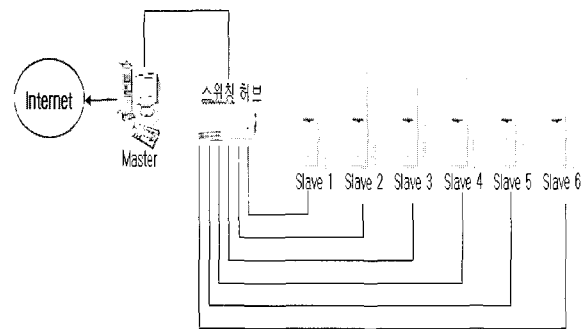


그림 1 PC 클러스터 시스템 구성

Fig. 1 Configuration of PC cluster system

본 연구에서 구축한 PC 클러스터 시스템은 Ethernet switch 기반 Fast ethernet방식의 6-node PC 클러스터 시스템이다. 운영 체제는 Windows 2000 server와 Windows2000 pro를 사용하였으며, NIC으로는 Fast ethernet card를 사용하여 스위칭 허브로 연결하였고, 데이터 전송은 TCP/IP를 이용한 message-passing방식으로 MPI library를 이용하였다. 원격 관리프로그램으로 symantec PC-anywhere, 병렬프로그래밍을 위해 MS Visual C++ 6.0을 사용하였다. 표 1에 시스템의 구체적인 내역을 제시하였다.

표 1 6-node PC 클러스터 시스템의 사양

Table 1 Specification of 6-node PC cluster system

CPU	Intel 2.0 GHz
M/B	Leotec P4XFA
Chipset	VIA P4X266A
RAM	DDR-SD RAM 256 MB
HDD	Samsung 40GB 5600rpm
NIC	3Com 3CSOHO 100-TX
Network Switch	3Com 3C16465C Switch
OS	Windows2000server, Windows2000pro
MPI Lib.	MPICH 1.2.5
Compiler	Visual C++6.0

3.2 병렬 PSO 알고리즘

3.2.1 PSO 알고리즘 개요

Particle Swarm Optimization(PSO)[8]라 불리는 새로운 진화연산 기법의 일종으로 1995년에 J. Kennedy 와 R. Eberhart 에 의해 제안되었다. PSO는 물고기의 떼 (schooling)와 조류의 무리(flocking)와 같은 조직체의 행위에 관한 논문에서 동기가 유발되었다. PSO는 군생 모델 (social model)과 인식 모델(cognitive model)을 조합으로 이루어졌다. 군생 요소는 개인이 자신의 경험을 무시하고 주위 개인들의 신조에 따라 그들의 행동을 조절한다는 것을 암시하고, 인식 요소는 개인들의 고립된 개체로서 취급한다는 것이다. 즉, 이러한 무리가 먹이를 찾아가는 과정에서 무리 전체의 정보를 서로 공유한다는 가설과 무리내부의 각 개체(agent)가 자기 자신의 경험과 무리 전체에 공유되어있는 정보를 사용하여 각 개체의 의견을 결정한다는 개념을 최적화 과정에 도입한 방법이라 할 수 있다. PSO의 특징은 알고리즘이 간단하고, 구현하기도 쉬우며, 계산시간이 짧으며 대용량의 메모리가 필요 없고, 다른 휴리스틱 알고리즘과는 달리, PSO 탐색공간에서 전역해와 국부해 사이를 균형 있게 탐색함으로써 조기 수렴하는 것을 극복 할 수 있다. 이러한 PSO 알고리즘의 특징은 다음과 같다.

- PSO는 GA등과 같이 복수의 탐색점을 가지며 각 탐색점의 Pbest (personal best)와 집단의 Gbest (group of global best)를 이용하여 각 탐색점을 확률적으로 변경시켜가는 것에 의해 전역적인 최적해를 탐색하는 최적화 방법이다.
- PSO는 전역 및 국부 탐색의 결합으로 기존 방법들보다 전역해를 찾는데 더 유연하고 강인하게 만들 수 있다.
- PSO는 기존의 다른 휴리스틱 방법과는 다르게, 초기 해 집단에 의해서 좌우되지 않는다. 그래서 탐색공간 어디에서 출발하여도 각 개체들간의 공유된 정보에 의해서 조기 수렴을 극복할 수 있고 전역 최적해에 수렴할 수 있다.

○ PSO의 개념은 원래 2차원 공간에서 고안되었으나 n차원 공간으로도 확장가능하다.

그림 2는 Pbest와 Gbest에 대한 개념을 나타내었다. 그림 2에서 보는 것 처럼, 탐색과정 동안 각 탐색점은 목적함수를 사용하여 평가되고 각 개체는 경로를 따라 현재까지 최적의 목적함수 값을 가지는 탐색점, Pbest를 기억하고 있다. 또한, 모든 개체의 Pbest 중 최적의 목적함수 값을 가지는 탐색점을 Gbest라 하고 이 정보를 공유한다. 각 개체는 현재의 위치 벡터와 속도 벡터, 그리고 Pbest, Gbest를 이용해서 식 (17)에 의해 이동을 하게 된다. 여기서 속도는 각 탐색점의 방향을 변화시킬 수 있고, 식(17)의 전역 및 국부탐색의 결합을 볼 수 있다. 각 개체의 위치벡터 수정은 현재의 위치와 수정된 속도를 이용해서 식(18)과 같이 갱신된다.

$$V_i^{k+1} = w \cdot V_i^k + c_1 \cdot rand_1(P_{best} - S_i^k) + c_2 \cdot rand_2(G_{best} - S_i^k) \quad (17)$$

$$S_i^{k+1} = S_i^k + V_i^{k+1} \quad (18)$$

- 여기서, V_i^k : i 번째 개체의 k 번째 세대의 속도 벡터
- S_i^k : i 번째 개체의 k 번째 세대의 위치 벡터
- V_i^{k+1} : i 번째 개체의 k+1 번째 세대의 속도 벡터
- S_i^{k+1} : i 번째 개체의 k+1 번째 세대의 위치 벡터
- N : 개체의 총 수
- w : 관성 가중치(inertia weight)
- c_1, c_2 : 가속 상수(acceleration constant)
- P_{best_i} : i번째 개체가 지금까지 탐색 중 발견한 최적 위치벡터
- G_{best} : 전체 개체가 지금까지 탐색 중 발견한 최적 위치벡터
- k : 1,2,...,n

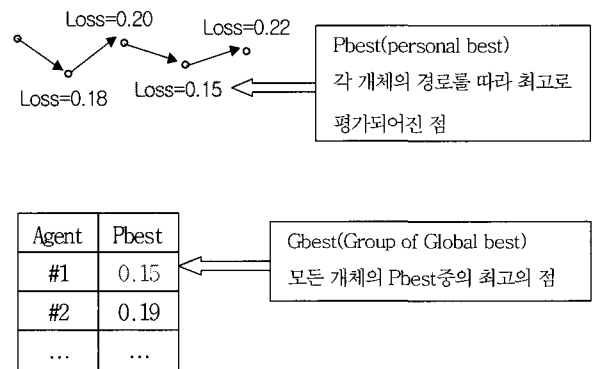


그림 2 Pbest와 Gbest의 개념

Fig. 2 Concept of Pbest and Gbest

그림 3은 각 개체의 속도와 위치의 변경에 관한 개념을 나타내었다. 그림에서 현재 위치와 그 속도가 주어졌으므로 한다. 예를 들어 이 개체가 Pbest, Gbest 위치를 기억하

고 있을 때 다음과 같은 속도로 변경한다. 이 개체는 현재 향하고 있는 방향으로부터 이제까지 탐색한 해 중에 제일 우수한 해 Pbest와 무리 전체가 발견한 해 중에서 제일 우수한 해인 Gbest가 있는 방향으로 자신의 방향을 변경하고자 한다. 이 변경하고자 하는 행동은 V^{mod} 벡터로 표현된다. 즉, 이 개체의 속도 변경은 현재의 속도 V^k 와 V^{mod} 를 더하는 것으로 이루어지면, 속도를 변경한 후의 개체는 V^{k+1} 로 표현된다. 따라서 식(18)에 의해서 현재 위치 S^k 는 V^{k+1} 를 더하여 새로운 위치 S^{k+1} 로 자신의 위치를 변경한다.

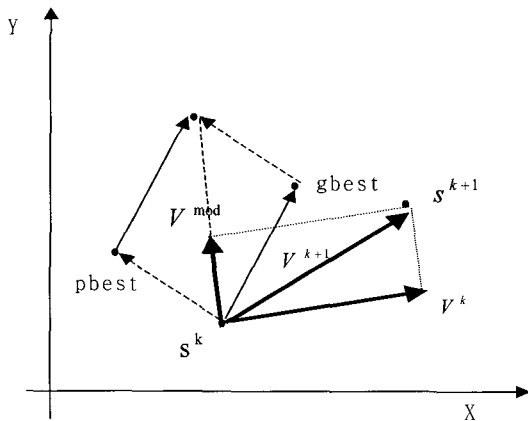


그림 3 각 개체의 속도와 위치 변경에 관한 개념
Fig. 3 Concept of updating the position and velocity of particle

각 개체는 해 공간안에서 무리로서 이동하고, 그리고 각 개체들간의 정보 교환에 의해서 최적해를 찾는다. 그림 4는 이러한 과정을 나타내고 있다.

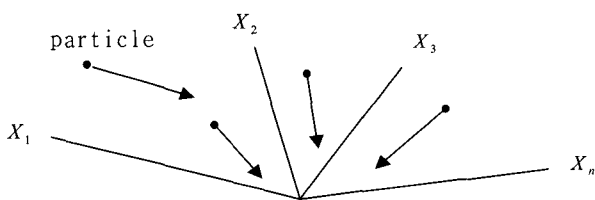


그림 4 각 개체들간의 정보교환 과정
Fig. 4 Information exchange process between particles

3.2.2 병렬 PSO 알고리즘

해집단 탐색을 토대로 하는 진화연산기법에서 해집단의 크기는 탐색성능에 직접적인 영향을 미치는 중요 파라미터 중의 하나이다. 해집단이 클 경우 좋은 탐색성능을 기대할 수 있지만, 상대적으로 많은 연산시간이 소요되는 단점이 있다. 또한 해집단의 크기가 너무 작을 경우 탐색성능이 현저히 저하되는 결과를 초래하기도 한다. 따라서, 본 논문에서는 적은 해집단 크기로 탐색성능의 저하를 초래하지 않으면서 연산시간을 단축할 수 있는 PC 클러스터 시스템 기반 병렬 PSO 알고리즘을 제안하였다.

일반적인 병렬 PSO 알고리즘(SPPSO)의 구조는 그림 5와 같다. 하나의 부모 해집단을 여러 개의 서브 해집단으로 나누고, 서브 해집단 간에 서로 최적해 탐색정보를 교환할 수 있는 구조로 설계하였다.

앞서 식(17)에서 언급한 바와 같이, PSO 알고리즘에서 각 개체는 전역 해집단 중 최고 적합도 개체의 위치정보와 자기 자신이 찾은 최고 적합도 위치정보를 이용하여 다음 이동지점을 결정하게 된다. 따라서, 전체 해집단의 최고 적합도 개체의 위치정보는 각 서브 해집단간의 통신에 의해 결정되어 진다. 이러한 형태의 병렬 구조는 일반적인 PSO 알고리즘의 메커니즘을 그대로 반영할 수 있어 병렬계산으로 인한 최적해 탐색능력 저하 가능성을 제거할 수 있지만, 서브 해집단의 수, 즉 프로세서의 수가 증가 할수록 통신량이 증가하게 되는 단점을 가지고 있다.

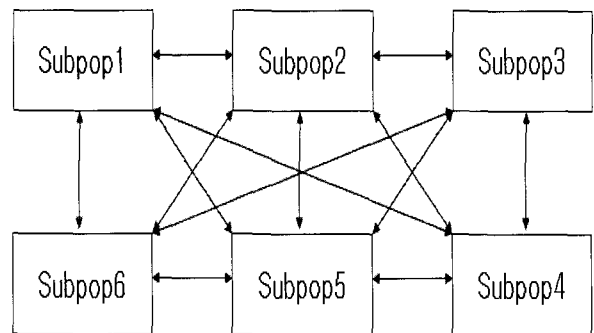


그림 5 일반적인 병렬 PSO 알고리즘 (SPPSO)
Fig. 5 General parallel PSO algorithm (SPPSO)

이에 반해 본 논문에서 제시하는 링 구조의 병렬 PSO 알고리즘(RPPSO)은 아래 그림 6과 같이 하나의 부모 해집단을 여러 개의 서브 해집단으로 나누고, 링 구조로 연결하여 서로 인접한 서브 해집단 간에만 최적해 탐색정보를 교환하는 구조이다. 이러한 연결구조에서는 각 서브 해집단이 항상 자기와 바로 인접한 2개의 인근 서브 해집단과 정보를 교환하기 때문에, 서브 해집단 수가 증가하더라도 각 서브 해집단의 통신량은 항상 일정하게 된다. 따라서, 서브 해집단 수, 즉 프로세서 수가 증가하더라도 통신에 소요되는 시간이 동일하게 되어 병렬처리 효율을 향상시킬 수 있다.

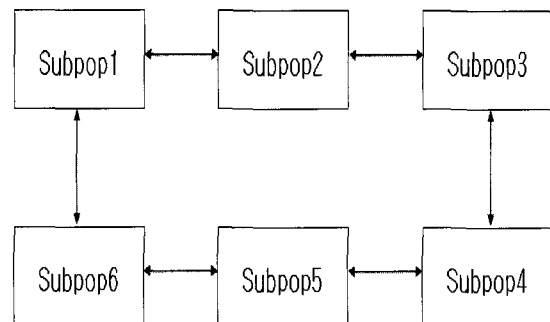


그림 6 링 구조를 갖는 병렬 PSO 알고리즘(RPPSO)
Fig. 6 Ring type parallel PSO algorithm (RPPSO)

제안한 알고리즘은 먼저 하나의 임의의 해집단을 만들고 이를 N개의 서브 해집단으로 나눈 후 PC 클러스터의 각 프로세서들에 하나의 서브 해집단을 배당한다. 이후 각 프로세서는 독립적으로 PSO 알고리즘 연산을 수행한 후, 이주 (migration) 연산을 통해 인접 프로세서와 현재까지 발견한 우수한 해 정보를 상호 공유함으로써 최적해 탐색능력을 향상시키도록 하였다.

제안한 병렬 PSO 알고리즘은 초기해 생성과정, 적합도 평가과정, 각 프로세서간의 정보교환 과정, 각 개체 위치 갱신 과정으로 구성되어 있으며, 세부내용과 흐름도는 그림 7에 나타내었다.

- (단계1) Initialize population : 하나의 부모 해집단을 랜덤하게 생성하고, 이를 N개의 서브 해집단으로 나누어 각 프로세서에 하나의 서브 해집단을 배당한다.
- (단계2) Evaluate fitness function : 각 프로세서에서 서브 해집단의 적합도를 평가한다. 적합도 평가 후 각 서브 해집단의 최고 적합도를 가진 개체의 정보를 기록한다.
- (단계3) Information exchange : 자신의 서브 해집단이 찾은 최고 적합도 개체의 정보를 인접한 2개 프로세서에 상호 전송함으로써 최적해 탐색정보를 공유한다.
- (단계4) Update position of each particle : 인접 프로세서와의 정보교환 수행후, 자신의 최적해 정보와 인근 서브 해집단에서 제공받은 최적해 정보를 종합하여 서브 해집단내의 각 개체의 위치를 갱신한다.

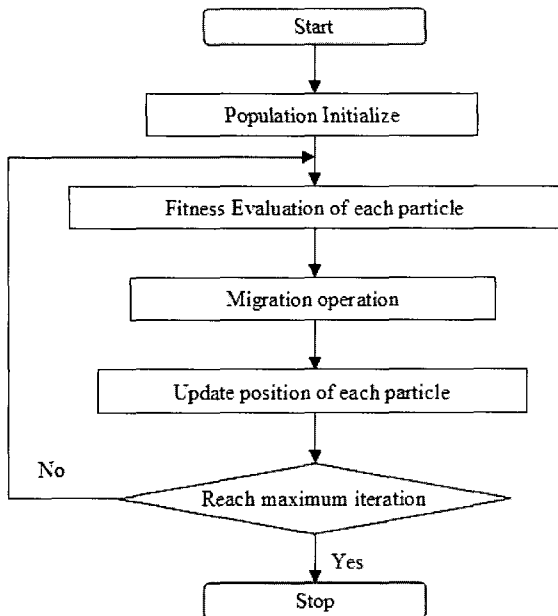


그림 7 제안한 병렬 PSO 알고리즘 연산 흐름도
Fig. 7 Flowchart of proposed parallel PSO algorithm

3.3 병렬 PSO 알고리즘을 이용한 최적조류계산

실 계통에서 최적조류계산을 행하여 실시간으로 계통을 제어하고 모니터링하기 위해서는 최적해를 가능한 한 빠른 시간에 구해야 하며 또한 비상시에도 안전하게 계통을 운용할 수 있는 고가용성의 시스템이 필요하다. 따라서 본 논문에서는 최근에 소개된 새로운 진화연산기법인 PSO 알고리즘을 이용한 최적조류계산 기법을 제안하였고, 이를 PC 클러스터 시스템으로 병렬화하여 전역 최적해 탐색능력을 저하시키지 않으면서 탐색시간을 단축하였다. 최적조류계산 문제에 제안한 알고리즘을 적용하는 방법은 다음과 같다.

PSO 알고리즘에서 제어변수인 발전기의 유효전력 출력, 발전기의 전압 크기, 변압기의 변압비, 전력용콘덴서 투입량을 각각 실수형태의 변수로 표현하여, 하나의 위치벡터를 구성하도록 하였고, 이를 그림 8에 나타내었다. 또한 제어변수의 최대 및 최소 한계 제약조건은 각 개체의 초기 생성과정에서 자동적으로 만족시켰다.

$$\begin{aligned}
 S_1 &= (P_{G1,1}, \dots, P_{GN,1}, V_{G1,1}, \dots, V_{GN,1}, t_{1,1}, \dots, t_{N,1}) \\
 &\vdots \\
 &\vdots \\
 S_n &= (P_{G1,n}, \dots, P_{GN,n}, V_{G1,n}, \dots, V_{GN,n}, t_{1,n}, \dots, t_{N,n})
 \end{aligned}$$

그림 8 PSO 알고리즘의 위치벡터 구성
Fig. 8 Configuration of position vector of PSO algorithm

또한 본 논문에서 사용한 적합도 함수를 식 (19)에 나타내었다. 식 (19)에 나타낸 바와 같이 각 발전기의 연료비용 합이 작을수록 적합도 값도 적어짐을 알 수 있다. 그리고 최적조류계산에서 운용상의 제약조건들은 적합도 함수에 페널티 항을 두어 제약조건 위반 정도에 따라 벌점을 부과하였다.

$$FF = \sum_{i=1}^{N_G} F_i(P_{Gi}) + \sum_{j=1}^{N_C} w_j \cdot Pen_j \quad (19)$$

- 여기서, FF : 적합도 함수
- w_j : j번째 계통 운용과 관련된 제약조건 가중치
- $F_i(P_{Gi})$: i번째 발전기의 연료비용 함수
- Pen_j : j번째 계통 운용과 관련된 제약조건 페널티 함수
- N_G : 발전기의 대수
- N_C : 계통 운용과 관련된 제약조건 개수

식 (19)의 적합도 함수를 사용한 후보해의 평가 과정은 다음의 단계 1) ~ 단계 4) 와 같다.

- (단계 1) 제어변수 u 를 생성한다. 생성된 제어변수는 최대 및 최소 한계의 제약 조건을 만족한다.

- (단계 2) 조류계산 과정을 수행하여 상태변수 x 를 구한다.
- (단계 3) 계통 운용상의 제약조건을 위반하였는지의 여부에 따라, 벌점을 부과한다.
- (단계 4) 식 (19)를 사용하여 적합도를 계산한다.

4. 사례 연구

본 논문에서 제시한 기법의 효과를 검증하기 위해서 테스트 계통[2]에 적용하여 그 결과를 비교검토 하였다. 검토대상은 IEEE 30모선 계통으로, 6개의 발전기, 21개의 부하 모선, 41개의 분기로 구성되어 있다. 최적조류계산의 목적함수는 총 유효발전 비용을 최소화하도록 설정하였으며, 제어변수는 5대 발전기의 유효전력 출력량, 6개 발전기 모선의 전압 크기, 4개 변압기 탭 비, 9개 전력용콘덴서 투입량으로 총 24개이다.

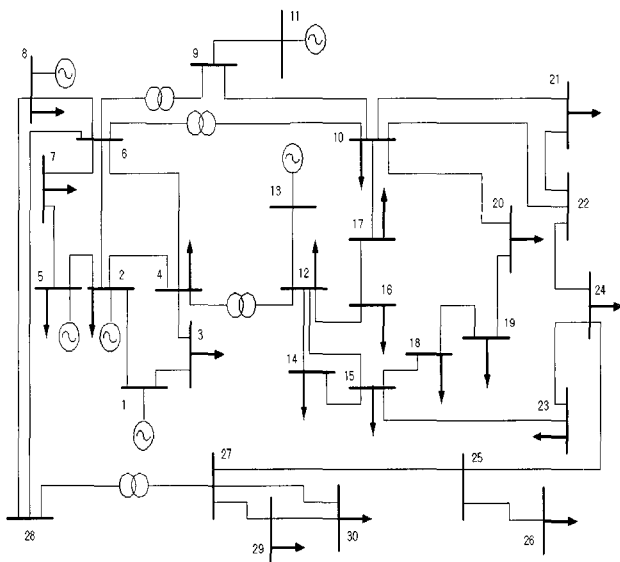


그림 9 IEEE 30 모선 계통
Fig. 9 IEEE 30 bus system

표 2에서는 발전기 관련한 데이터를 나타내고 있다. 변압기 탭비는 0.9pu~1.1pu이며, 전력용콘덴서 투입량은 0pu~0.05pu, 모선전압 허용범위는 발전기 모선 0.95~1.1pu, 부하 모선 0.95pu~1.05pu로 설정하였다.

본 논문에서 적합도 평가를 하기위한 조류계산은 Fast Decoupled Newton-Raphson 기법을 이용하였으며, 조류계산의 수렴조건은 실 계통해석에서 일반적으로 사용하는 최대 유효 및 무효전력 mismatch 0.001pu 이하를 적용하였다.

그림 10에서는 본 논문의 병렬연산을 위해 구축한 PC 클러스터 시스템을 나타내고 있다. 본 시스템은 Fast Ethernet 방식의 6-node PC 클러스터 시스템이며, 각 node에 탑재된 프로세서는 Intel Pentium IV 2.0 Ghz이다.

표 2 발전기 관련 데이터

Table 2 Generation data

Bus No.	P_G^{\min} (MW)	P_G^{\max} (MW)	Q_G^{\min} (MW)	S_G^{\max} (MVA)	Cost coefficients		
					a	b	c
1	50	200	-20	250	0.0	2.00	0.00375
2	20	80	-20	100	0.0	1.75	0.01750
5	15	50	-15	80	0.0	1.00	0.06250
8	10	35	-15	60	0.0	3.25	0.00834
11	10	30	-10	50	0.0	3.00	0.02500
13	12	40	-15	60	0.0	3.00	0.02500

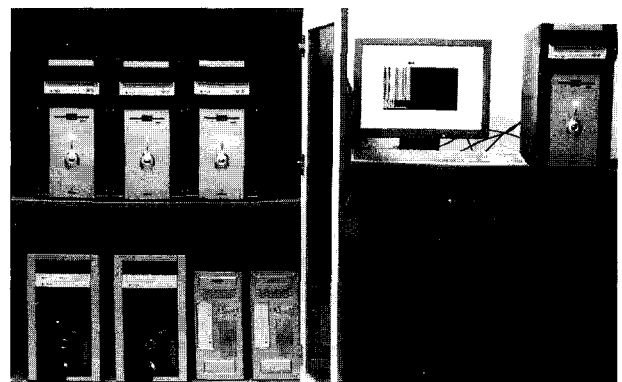


그림 10 PC 클러스터 시스템
Fig. 10 PC cluster system

표 3에서는 최적조류계산시 PSO 알고리즘에서 사용한 시뮬레이션 계수를 나타내고 있으며, w , c_1 , c_2 는 일반적으로 알려진 typical 값을 사용하였다.

표 3 PSO 알고리즘 파라미터

Table 3 Parameters of PSO algorithm

파라미터	설정치
세대수	50
해집단 수	60
c_1, c_2	2.0
w	0.9~0.4

아래 표에서는 PSO 알고리즘을 이용한 최적조류계산 결과를 나타내고 있다. 먼저, 표 4 및 표 5에서는 최적조류계산에 의해 도출된 변압기 탭 설정 및 전력용콘덴서 투입량을 나타내고 있으며, 표 6은 발전기 모선의 전압크기, 유효 전력 출력량 및 발전 비용을 나타내고 있다. 총 발전비용을 참고문헌[19] 결과와 비교한 결과, 본 논문의 발전비용이 \$800.81/hr로써 참고문헌에서 제시하는 804.8\$/hr 보다 더 우수함을 확인 할 수 있었다.

표 4 변압기 탭

Table 4 Results of tap ratio of transformer

변압기 번호	1(4-12)	2(6-9)	3(6-10)	4(28-27)
변압기 탭 비	0.9933	1.0241	0.9526	0.9815

표 5 전력용콘덴서 투입량

Table 5 Results of shunt capacitor

Bus No.	전력용콘덴서 투입량(pu)	Bus No.	전력용콘덴서 투입량(pu)
10	0.009681	21	0.035668
12	0.022531	23	0.042323
15	0.023112	24	0.025306
17	0.023322	29	0.046592
20	0.034603	-	

표 6 발전기 모선의 전압, 유효전력 출력, 발전비용

Table 6 Result of generators

Bus No.	Voltage (p.u.)	P _G (MW)	Cost (\$/hr)
1	1.083	176.95	471.33
2	1.059	48.63	126.50
5	1.030	21.36	49.89
8	1.036	21.67	74.37
11	1.074	11.83	38.99
13	1.067	12.03	39.73
Total	-	292.49	800.81

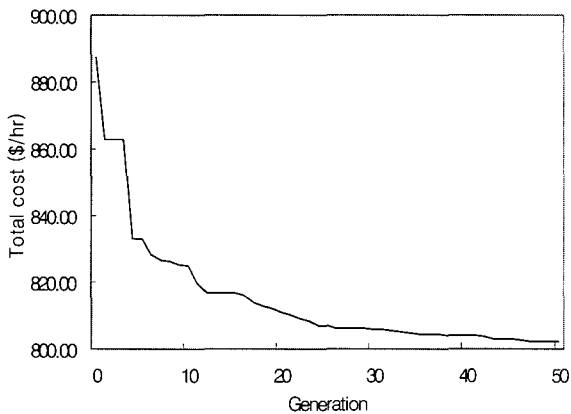


그림 11 세대수에 따른 발전비용 변화추이

Fig. 11 Evolution process of parallel PSO algorithm

그림 11에서는 세대수 증가에 따른 발전비용의 변화추이를 나타내고 있다. 초기 \$880/hr 정도인 발전비용이 세대수가 증가함에 따라 점차 감소하여 50세대에서는 \$800.81/hr로 수렴함을 알 수 있으며, 50세대까지 소요되는 전체 연산시간은 47.98sec로 나타났다.

본 논문의 목적이 탐색성능의 저하없이 최적해 탐색 시간을 단축하는데 있으므로, 제안한 RPPSO 알고리즘의 최적해 탐색능력과 연산시간을 살펴보았으며, 이를 일반적인 병렬 PSO 알고리즘인 SPPSO 알고리즘과 비교하여 그 우수성을 입증하였다. 표 7 및 표 8에서는 기존 PSO 및 병렬 PSO의 발전비용 및 연산시간 결과를 나타내고 있다. 표에서 알 수 있듯이 발전비용은 병렬화 이전과 거의 동일한 수준의 값을 나타내었으며, 연산시간은 프로세서 수가 증가함에 따라 비례하여 감소하는 결과를 보였다. 프로세서 수에 따른 연산시간을 살펴보면 1대의 프로세서를 사용하는 경우 약 48초가 소요되었으며, 6대의 프로세서를 사용한 경우 약 8초 정도가 소요되어 본 논문에서 목적인 탐색성능의 저하없는 연산시간 단축을 확인할 수 있었다. 또한, SPPSO와 RPPSO 알고리즘 간의 비교결과, 발전비용은 두 알고리즘이 거의 유사한 수준으로 나타났으나, 연산시간에서는 프로세서 수가 증가함에 따라 RPPSO가 더 유리한 것으로 나타났다. 그림 11에서 보여지듯이 프로세서 수가 4대에서 점차 증가함에 따라 SPPSO와 RPPSO 사이의 연산시간 비가 점점 더 커지고 있다. 이는, SPPSO가 프로세서 수가 증가 할 수록 통신량이 비례하여 증가하는 반면, RPPSO는 항상 동일한 통신량을 가지게 되어 연산시간 차이가 점점 커짐을 알 수 있다. 이러한 현상은 프로세서 수가 더욱 증가하면 보다 더 분명하게 나타날 것으로 사료된다.

표 7 발전비용 비교

Table 7 Comparison of generation cost

프로세서 수 (대)	발전비용 (\$/hr)					
	SPPSO			RPPSO		
	Min	Max	Ave	Min	Max	Ave
1	800.70	801.10	800.84	800.70	801.10	800.84
4	800.72	801.06	800.75	800.66	800.97	800.92
5	800.76	800.99	800.87	800.78	801.11	800.95
6	800.72	800.98	800.88	800.75	801.17	800.96

표 8 연산시간 비교

Table 8 Comparison of execution time

프로세서 수 (대)	연산시간 (sec)	
	SPPSO	RPPSO
1	47.98	47.98
4	12.26	12.17
5	9.94	9.74
6	8.59	8.24

참고 문헌

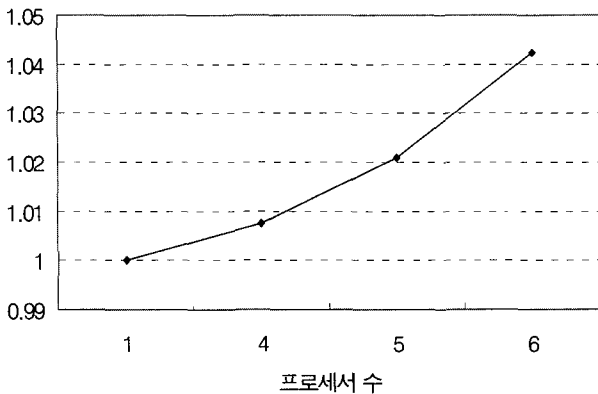


그림 12 SPPSO와 RPPSO간의 연산시간 비율
 Fig. 12 Ratio of execution time between SPPSO and RPPSO

5. 결 론

보다 신뢰성 높고 효율적인 전력계통의 운영 및 계획을 위해서 최적조류계산에 대한 요구가 점점 커져가고 있다. 특히, 전력IT 기술의 발달과 더불어 온라인 최적조류계산에 대한 필요성도 점점 부각되고 있다. 종래의 진화연산 기법을 이용한 최적조류계산 기법의 경우 탐색성능측면에서는 우수한 결과를 입증하였으나, 여러 개의 점을 동시에 탐색하는 근본적인 특성으로 인해 연산시간이 상대적으로 많이 소요되는 단점을 가지고 있었다. 이러한 연산시간측면의 불리함은 특히 온라인 최적조류계산 적용과 같은 문제에 있어서는 적용상의 중요한 장애요인으로 볼 수 있다. 따라서, 본 논문에서는 진화연산 기법의 일종인 PSO 알고리즘을 이용한 최적조류계산 문제의 시간단축을 위하여 PC 클러스터 시스템 기반 병렬 PSO 알고리즘을 제안하였다. 제안한 PC 클러스터 기반 병렬 PSO 알고리즘은 하나의 모 해집단을 여러 개의 서브 해집단으로 나누고 각 서브 해집단을 하나의 프로세서에 각각 할당함으로써 병렬연산이 가능하였다. 또한, 병렬구조를 링 구조로 하여 인접한 두 개 프로세서와 탐색 정보를 교환하도록 하여 각각의 서브 해집단 크기가 적어짐으로 인해 발생할 수 있는 최적해 탐색능력 저하를 방지하였다. 제안한 방법의 유용성을 입증하기 위하여 IEEE 30 모선 최적조류계산 문제에 적용하여, 프로세서 수에 따른 최적해 탐색능력과 연산시간을 비교검토 하였다. 검토결과, 본 논문에서 제안한 방법을 적용할 경우 탐색성능 저하없는 연산시간 단축효과를 확인할 수 있었다.

감사의 글

본 연구는 산업자원의 지원에 의하여 기초전력공학공동연구소 주관으로 수행된 과제(R-2005-7-064)임.

[1] Hermann W. Dommel and William F. Tinney, "Optimal power flow solutions," IEEE Trans. on Power Apparatus and Systems, Vol. PAS-87, pp. 1886-1876, October 1968.

[2] O. Alsac and B. Stott, "Optimal load flow with steady state security," IEEE Trans. on Power Apparatus and Systems, Vol. PAS-93, pp. 745-754, 1974.

[3] O. Alsac, J. Bright, M. Prais, and B. Stott, "Further developments in lp-based optimal power flow," IEEE Trans. on Power Systems, Vol. 5, pp. 697-711, August 1990.

[4] G. F. Reid and L. Hasdorf, "Economic dispatch using quadratic programming," IEEE Trans. on Power Apparatus and Systems, Vol. PAS-92, pp. 2015-2023, 1973.

[5] L. L. Lai, J. T. Ma, R. Yokoyama, and M. Zhao, "Improved genetic algorithms for optimal power flow under both normal and contingent operation states," Elec. Power Energy Syst., Vol. 19, No. 5, pp. 287-292, 1997.

[6] T. Numnonda and U. D. Annakkage, "Optimal power dispatch in multinode electricity market using genetic algorithm," Elec. Power Syst. Res., Vol. 49, pp. 211-220, 1999.

[7] Q. H. Wu and J. T. Ma, "Power system optimal reactive power dispatch using evolutionary programming," IEEE Trans on Power Syst., vol.10, pp.1243-1249, 1995.

[8] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," IEEE international Conference on Neural Networks, vol.4, pp.1942-1948, 1995.

[9] J.-Y. Choi, J.-R. Shin, and M.-H. Kim, "Parallel computing environment based on windows operating system," 한국항공우주학회지, May 2002.

[10] R. Romano, Victor H. Quintana, R. Lopez, and V. Valadez "Constrained economic dispatch of multi-area system using Dantzig-Wolfe decomposition principle", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-100, No. 4, pp. 2127-2137, 1981.

[11] J. Q. Wu, and A. Bose "Parallel solution of large-sparse matrix and parallel power flow", IEEE Power Engineering Society 1994 Summer Meeting, SanFrancisco, CA, July, 15-19, 1994.

[12] R. P. Sundarraj, S. Kingsley Gnanendran, and J. K. Ho "Distributed price-directive decomposition applications in power systems", IEEE Power Engineering Society 1994 Summer Meeting, SanFrancisco, CA, July, 15-19, 1994.

[13] Balho H. Kim and Ross Baldick "Coarse-grained distributed optimal power flow", IEEE Trans. on

Power Systems, Vol. 12, No. 2, pp. 932-939, 1997.

- [14] R. Tanese, "Parallel genetic algorithm for a hypercube," Proc. of 2th ICGA'87, pp. 177-183, 1987.
- [15] R. Schonkwiler, "Parallel genetic Algorithms," Proc. of International Conference on Genetic Algorithms, pp. 199-205, 1993.
- [16] 김형수, 병렬 프로세서와 타부 탐색법을 이용한 수화력 단기 발전계획, 박사논문, 부산대학교, 2002.
- [17] 김태균, 김규호, 유석규 "병렬유전알고리즘을 응용한 대규모 전력계통의 최적 부하배분", 대한전기학회 논문지, Vol. 48A, No. 4, pp. 388-394, 1999.
- [18] C. H. Lol, C. Y. Chung, D. H. M. Nguyen, and K. P. Wong "Parallel Evolutionary Programming for optimal power flow", IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies(DPRT2004), Hong Kong, April, 2004.
- [19] K. Y. Lee, Y. M. Park, and J. L. Ortiz "A United approach to optimal real and reactive power dispatch", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-104, No. 5, pp. 1147-1153, 1985.



이 화 석 (李 和 錫)

1966년 7월 10일생. 1991년 부산대 공대 전기공학과 졸업. 1993년 동 대학원 전기공학과 졸업(석사). 1997년 동 대학원 전기공학과 졸업(공학박). 현재 거제대학 전기과 부교수.

Tel : 055) 680-1604
Fax : 055) 681-3993
E-mail : hslee@koje.ac.kr



박 준 호 (朴 俊 灝)

1955년 9월 17일생. 1978년 서울대 공대 전기공학과 졸업. 1980년 동 대학원 전기공학과 졸업(석사). 1987년 동 대학원 전기공학과 졸업(공학박). 1978~1981년 충남대 공대 전기공학과 전임강사 및 조교수. 현재 부산대 공대 전자전기정보컴퓨터공학부 교수.

Tel : 051) 510-2370
Fax : 051) 513-0212
E-mail : parkjh@pusan.ac.kr

저 자 소 개



김 종 울 (金 鍾 律)

1974년 07월 06일생, 1997년 부산대학교 전기공학과 졸업, 1999년 동 대학원 전기공학과 졸업(공학석사), 현재 한국전기연구원 전력연구단 신재생에너지그룹 선임연구원

Tel : 055-280-1336
Fax : 055-280-1390
E-mail : jykim@keri.re.kr



문 경 준 (文 景 俊)

1972년 10월 25일생. 1994년 부산대 공대 전기공학과 졸업. 1996년 동 대학원 졸업(석사). 2005년 동 대학원 전기공학과 졸업(공학박). 현재 한국원자력연구소 양성자 기반공학기술개발사업단 post-doc.

Tel : 042) 868-4663
Fax : 042) 868-8131
E-mail : kjmun@kaeri.ac.kr