

# 전력거래에서 최적조류계산과 혼잡비용 계산을 수행하는 인터넷 웹 프로그램 개발

論 文  
50A-6-1

## Development of Internet Web Program for the Calculation of OPF and Congestion Cost

李光浩\* · 鄭在玉\*\*  
(Kwang-Ho Lee · Jae-Ok Jung)

**Abstract** - The electric utility industry around the world is undergoing a revolutionary transition from vertically integrated monopoly structures to a competitive structures. Competition in an open transmission access(OTA) makes the market participants access easily the transmission system in a non-discriminatory and equitable manner. The competitions of suppliers and customers make the electric market price change every hour. This paper presents a web program in the internet environment with a function of optimal power flow(OPF) calculation. The Web program gives the nodal marginal cost and the congestion charge using the shadow prices resulted from OPF. This web program is realized by a Perl and JAVA languages, and using the common gateway interface(CGI).

**Key Words** : Internet Web Program, CGI, Optimal Power Flow, Congestion Cost

### 1. 서 론

현재 전 세계의 전력산업은 큰 변혁기에 직면하고 있다. 규모의 경제에 기초한 수직 통합 독점형 체제에서 기능분할에 기초한 시장경쟁 체제로 변화하고 있는 것이다.[1] 경쟁 체제에서는 공급자와 소비자가 동등하게 시장에 참여하므로 전력가격은 시간대별로 시장원리에 의해 결정된다. 이러한 시간대별 가격을 인터넷 기반에서 온라인으로 분석하게 되면 전력거래의 전략적인 측면에서 매우 효과적이다.

인터넷을 이용하면 시간이나 공간의 제약 없이 원하는 정보를 거의 실시간으로 제공받을 수 있는 장점이 있다. 인터넷 서비스 중 월드 와이드 웹(World Wide Web, Web)이라 불리는 서비스는 최근에 가장 주목받는 서비스로서 그 활용 영역이 점차 확대되고 있다.[2] 이런 추세에 맞추어 월드 와이드 웹은 사용자의 웹 브라우저에서만 이루어지는 단방향 통신이 아니라 HTML (Hyper Text Makeup Language)를 이용하여 서버와 사용자간의 양방향 통신을 할 수 있는 CGI(Common Gateway Interface)라는 하나의 규약을 탄생시켰다. CGI는 웹 페이지를 동적으로 구현할 수 있도록 해 주는 프로토콜로써 외부 프로그램과 웹 서버(HTTP Server)간의 연결 역할을 하기 위한 규약이다. 또한 월드 와이드 웹에서 객체 지향 프로그램으로 각광을 받고 있는 자바(JAVA)라는 언어의 발전으로 웹페이지에서 구현할 수 있는 적용 영역이 급격히 확산되고 있다.

본 논문에서는 각 발전사업자들의 연속적인 연료비 함수

를 인터넷을 통해 온라인으로 입력받아 비선형 최적화 기법인 최적조류계산(OPF: Optimal Power Flow)을 수행하는 프로그램을 개발했다. 발전 비용함수가 이차함수 형태로 주어지는 경우 제약 조건을 1차 근사화하여 DC OPF의 계산에 SQP(Successive Quadratic Programming)기법을 적용할 수 있다.[3] 본 연구에서 개발한 프로그램은 웹(Web) 프로그램 언어 중 하나인 펄(Perl :Practical Extraction and Report Language)을 CGI에 적용시켜 구현했으며 발전기, 선로 및 부하 데이터 입력은 HTML의 폼(FORM) 태그와 자바 스크립트를 이용하여 구성하였다. 또한 OPF의 결과로 도출되는 잠재가격(Shadow Price) 정보를 이용하여 모선별 한계비용(Nodal Marginal Cost)을 제시하며 송전선 혼잡이 발생했을 경우 모선별, 선로별 혼잡비용(Congestion Cost)을 계산하여 제공한다.

### 2. 최적조류계산과 혼잡비용

#### 2.1 모션한계비용

발전력 최적조류계산은 전력계통의 여러 가지 제약조건하에서의 경제급전계획을 의미하며 개념적으로는 경제급전계획과 전력조류계산을 동시에 수행하는 것이다. 비용최소화를 목적함수로 하고 선로의 손실을 고려하지 않을 경우 제약조건에는 다음 식(1)과 같이 계통의 수급조건과 선로에서의 최대 허용 용량, 그리고 각 발전기의 최소 및 최대 출력 조건을 포함한다.

$$\text{Minimize } \sum_{i=1}^n C_i(P_{Gi})$$

\* 正 會 員 : 檀國大 工大 電氣工學科 助教授 · 工博

\*\* 正 會 員 : OjuSnet SunSalt R&D Center

接受日字 : 2000年 11月 2日

最終完了 : 2001年 5月 22日

$$\begin{aligned} \text{Subject to } & \sum_{i=1}^m P_{Gi} - \sum_{k=1}^m P_{Dk} = 0 \\ & P_i \leq P_i^{\max} \\ & P_{Gi}^{\min} \leq P_{Gi} \leq P_{Gi}^{\max} \end{aligned} \quad (1)$$

여기서  $C_i$ 는 발전비용함수이며,  $P_{Gi}$ 는 발전력,  $P_{Dk}$ 는 부하를 나타낸다.  $P_{Gi}^{\min}, P_{Gi}^{\max}$ 는 각각 발전력의 하한과 상한값이고  $|P_i|$ ,  $P_i^{\max}$ 는  $l$  선로의 선로조류와 최대 허용용량이다. OPF를 수행하여 도출되는 잠재가격 정보를 이용하여  $i$ 번째 모선의 모션한계비용( $MC_i$ )을 계산하는 식은 한계비용 이론에 따라 다음 식(2)와 같다.[4]

$$MC_i = \lambda - \sum_{l=1}^{N_l} u_l \frac{\partial P_l}{\partial P_i} \quad (2)$$

여기서  $\lambda$ 는 전력수급조건에 관련된 라그랑지안 승수이며,  $u_l$ 은  $l$ 선로의 선로조류 한계용량에 해당하는 라그랑지안 승수이다. 또한  $N_l$ 은 전체 선로의 개수이고,  $P_l$ 은  $i$ 번째 모선에서의 주입전력을 의미한다. 혼잡비용은 모션한계비용을 이용해서 두 가지 방법으로 구할 수 있다. 첫 번째는 수용가가 지불해야 할 비용과 발전소가 수령해야 하는 금액의 차이로 계산되고, 두 번째는 다음 식(3)과 같이 각 선로에 대해서 구하는 것으로 선로 양단 모션에서의 모션한계비용의 차이와 선로조류의 곱으로 계산된다.[5]

$$CC_{ij} = (MC_j - MC_i) \times P_{ij} \quad (3)$$

여기서  $CC_{ij}$ 는  $ij$ 선로의 혼잡비용이고,  $P_{ij}$ 는  $ij$ 선로의 선로조류,  $MC_j$ 는  $j$ 번째 모션에서의 모션 한계비용을  $MC_i$ 는  $i$ 번째 모션에서의 모션한계비용을 의미한다.

## 2.2 SQP(Successive Quadratic Programming)

SQP 기법은 목적함수를 2차 형태로, 제약조건을 1차 형태로 근사화하여 최적조건식을 계산하는 방식이다. 다음은 일반적 최적화 문제에 대한 SQP 해법을 소개한다.

$$\begin{aligned} \text{Min } & f(x) \\ \text{s.t. } & h(x) = 0 \\ & g_i(x) \leq 0, \quad i = 1, \dots, m \end{aligned} \quad (4)$$

식(4)의 KKT 최적조건식에 대해 임의의 상태( $x_k, u_k, v_k$ )에서 테일러 전개를 적용하여 QP(Quadratic Programming)형태로 나타내면 다음 식(5)와 같다.[3]

$$\begin{aligned} \nabla f(x_k) + \nabla^2 L(x_k)d + \nabla g(x_k)u + \nabla h(x_k)v &= 0 \\ u_i [g_i(x_k) + \nabla g_i(x_k)'d] &= 0, \quad i = 1, \dots, m \\ h(x_k) + \nabla h(x_k)d &= 0 \end{aligned} \quad (5)$$

여기서

$$\nabla^2 L(x_k) = \nabla^2 f(x_k) + \sum_{i=1}^m u_{ki} \nabla^2 g_i(x_k) + \sum_{i=1}^m v_{ki} \nabla^2 h_i(x_k)$$

이며  $L$ 은 라그랑지안이고  $d = x - x_k$ 이다. 최적화 과정에서 뉴턴 기법으로 식 (5)의 해를 계산하며 2차 문제임이 가정되었기 때문에 식 (5)의 해는 해석적으로 유도가 된다.[3]

## 3. 웹 프로그램의 구현

### 3.1 인터넷(Internet)

인터넷은 다음 그림 1과 같이 네트워크와 네트워크들이 모인 하나의 거대한 네트워크(Network of Networks) 집합체이다. 인터넷을 이용하면 시간이나 공간의 제약없이 원하는 정보를 거의 실시간으로 제공받을 수 있다.

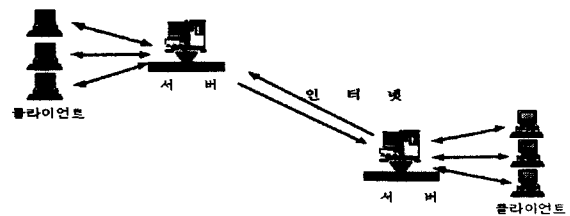


그림 1 인터넷의 개념도

Fig. 1 Diagram of Internet

첫 미국 국방성의 ARPA(Advanced Research Projects Agency)는 일정지역에 대한 긴급 사태시에도 장애를 받지 않고 제 기능을 발휘할 수 있는 통신망 구축을 연구하여 네트워크를 개발했는데 이것이 바로 연구를 목적으로 한 최초의 네트워크인 ARPANET(1969)이다. 이후 미국 과학재단의 교육 전산망인 NSFNET이 연결(1986)되었고 1990년 이후에는 일반 상업적인 목적의 네트워크이 연결되면서 현재의 인터넷으로 발전을 이루었다.[2]

인터넷 서비스에는 전자 우편(E-mail) 서비스, FTP(File Transfer Protocol) 서비스, Telnet(Telecommunication Network) 서비스, 뉴스그룹(NewsGroup) 서비스, 인터넷 채팅(IRC : Internet Relay Chatting) 서비스, Gopher 서비스가 있으며 가장 최근에 개발되어 가장 많이 사용하고 있는 서비스로 멀티미디어 형태의 인터넷 정보를 편리하게 사용할 수 있는 월드 와이드 웹(WWW: World Wide Web) 서비스가 있다. 규제완화된 전력계통하에서는 다수의 공급자와 소비자가 전력시장에 동등하게 참여하므로 시장 참여자들의 경쟁과 선택(Competition and Choice)으로 인해 전력산업의 효율성 증진을 위해서는 시간적, 공간적 제약을 받지 않는 인터넷을 이용하는 것이 바람직하다. 따라서 인터넷을 기반으로 다수의 사용자가 웹 브라우저를 이용하여 전력거래를 할 수 있는 프로그램이 필요하다.

### 3.2 웹 프로그램(Web-Program)

인터넷이 널리 사용되고 웹사이트를 통해 수행되는 일들이 늘어나면서 정적인 HTML만으로는 부족함을 느끼게 되어 탄생한 것이 바로 CGI다. CGI는 동적으로 웹 페이지를 만들어주는 프로그램이라 할 수 있는데 넓은 의미로 CGI를 수행하는 외부 프로그램을 포함하여 말하기도 한다. 그래서 흔히 CGI 프로그램이라 하고 이러한 CGI 프로그램을 웹 프로그램이라 정의한다.[6] CGI는 다음 그림 2와 같이 웹 브라우저로부터 받은 사용자 입력을 바탕으로 웹 페이지를 동적으로 생성시킨다.

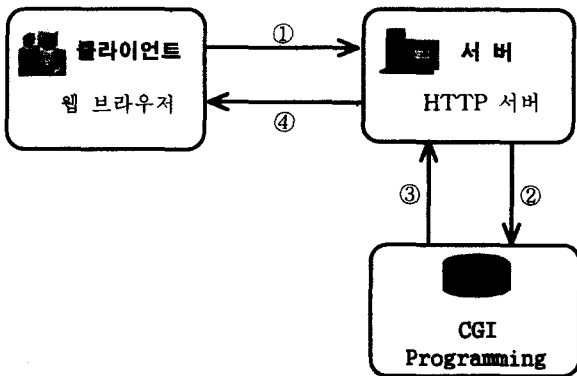


그림 2 CGI의 블록도

Fig. 2 Block Diagram of CGI

그림 2에서의 데이터 흐름을 살펴보면 다음과 같다.

- ① : 클라이언트(웹 브라우저)에서 서버로 TCP/IP 연결을 통해 데이터 전송.
- ② : 서버가 CGI 프로그램에게 실행을 하도록 요청.
- ③ : CGI 프로그램에서 서버로 출력이 전달.
- ④ : 적절한 MIME 헤더를 갖고 서버가 클라이언트로 응답한 후, 연결이 해제된다.

즉, 클라이언트가 HTML의 폼 태그를 이용해서 입력한 데이터를 TCP/IP를 통해 서버로 전송시키면 서버는 CGI 프로토콜을 이용해서 클라이언트의 요청을 수행한 후 다시 클라이언트로 출력을 전달하고 연결이 해제된다. 본 연구에서 사용하는 서버는 자신의 고유 IP 주소를 갖는 웹 서버로서 펄(Perl) 인터프리터(Active Perl), 자바 인터프리터(java) 및 자바 컴파일러(javac)가 탑재된 유닉스(UNIX) 서버를 사용하였다. 이와 같이 웹 프로그래밍이란 바로 CGI 프로그램의 개발을 의미하고, 이런 CGI 프로그램은 웹서버에서 지정한 형식을 맞출 수만 있다면 C나 C++등 어떤 언어로든지 작성할 수 있다.

### 3.3 펄(PERL)과 자바(JAVA)

펄(Perl)은 Larry Wall이 만든 Practical Extraction and Report Language라는 언어로 말 그대로 텍스트 파일을 검색하여 실용적인 정보를 추출(Extract)하고 그 정보를 가공하여 보고(Report)하는데 아주 적절한 언어이다. 유닉스나 리눅스에서는 셸 스크립트보다 더 정교한 처리가 가능한 스

크립트 언어이므로 시스템의 관리에도 많이 쓰인다. 펄은 C의 형태를 가지면서도 여러 가지 텍스트의 처리 기능이나 문자열의 일치여부 검색, 치환 등의 기능이 뛰어날 뿐만 아니라 C와 같은 까다로운 변수 타입의 엄격한 구분이 없어서 코딩이 무척 수월하다.[7] 인터프리터 언어인 펄은 C 처럼 컴파일 할 필요가 없어 소스코딩한 대로 실행이 가능하다. 물론 컴파일 언어의 장점인 코딩의 최적화나 실행속도에서는 다소 처지지만 컴파일을 반복할 필요가 없으므로 프로그램의 구현과 디버깅(Debugging)이 훨씬 수월하다. 또한 Perl은 유닉스나 NT 등의 여러 가지 운영체계에 사용할 수 있으므로 소스 파일을 이식하기가 아주 쉬어서 서버를 변경할 경우나 운영체제를 바꿀 경우에도 부담이 되지 않으며, 인터넷 상에서 무료로 자신의 시스템에 맞는 펄 인터프리터를 구할 수 있으므로 비용면에서도 커다란 장점이 있다.

자바(JAVA)는 순수한 객체지향언어로서 한 마디로 자바를 정의하기는 어렵다. 본 논문에서는 자바 애플릿(JAVA Applet)과 자바 스크립트(JAVA Script)를 사용하였다. 자바 애플릿은 월드 와이드 웹에서 브라우저를 통해 실행할 수 있는 프로그램이고, 자바 스크립트는 HTML 문서속에 코드가 직접 삽입되어 사용자가 마우스를 클릭하거나 키보드를 입력하는 것과 같은 작업을 즉시 처리할 수가 있다.[8] 본 연구에서 자바 애플릿은 프로그램의 텍스트 효과와 그래픽 효과에 사용되었으며 자바 스크립트는 사용자의 데이터 입력에 관련된 부분에 사용되었다.

## 4. 개발 프로그램

### 4.1 초기화면

본 논문에서 개발한 웹 기반 OPF 프로그램은 사용자의 편의를 위해서 다음 그림 3과 같이 데이터 입력에서부터 초기 조류계산 수행, 선로의 한계용량 위반 여부 검사, OPF 수행, 혼합비용계산의 순서로 수행된다. 사례연구의 대상 계통은 선로의 손실을 고려하지 않은 3기 5모선 계통으로 계통도 및 프로그램의 주 화면은 다음 그림 4와 같다.

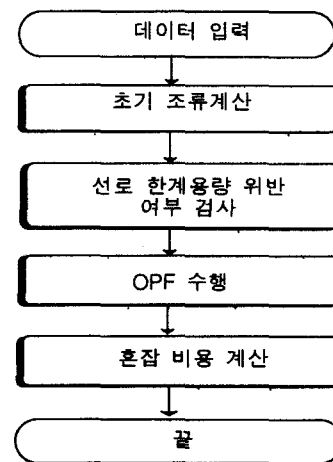


그림 3 프로그램 실행 순서도  
Fig. 3 Flowchart of Program

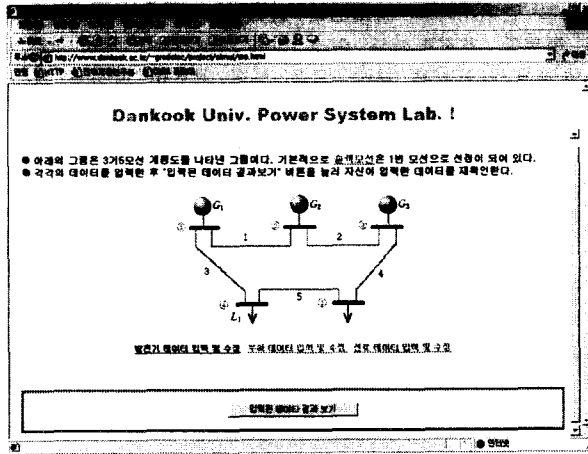


그림 4 프로그램 주 화면  
Fig. 4 Main Window of the Program

4.2 데이터 입력

본 연구에서 개발한 프로그램은 발전기 데이터와 선로 데이터, 그리고 부하 데이터를 자바 스크립트를 이용해서 새로운 창에 각각 독립적으로 입력을 받는다. 발전기 데이터 입력 화면은 그림 5.1과 같이 2차 함수 형태의 발전기 연료비 함수와 각 발전기의 상한 및 하한값을 입력 받는다.

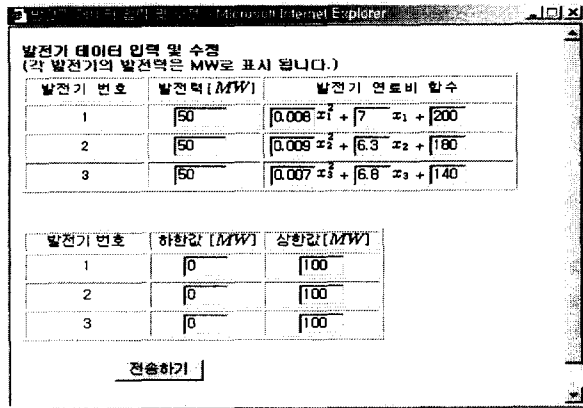


그림 5.1 발전기 데이터 입력 화면  
Fig. 5.1 Input Window for Generator Data

부하 및 선로 데이터 입력은 다음 그림 5.2와 같이 이루어지며, 이러한 입력 데이터는 웹 상에서 복수의 사용자가 접속할 경우 자신이 입력한 데이터를 구분하기 위해서 IP Address를 이용하여 사용자별로 저장된다. 자신이 입력한 데이터 값을 서버로 전송시키고 난 후 그 값을 확인 할 수 있으며 만약 자신이 입력한 데이터 값이 아닐 경우 자바 스크립트(Java Script)를 이용해서 수정이 가능하다.

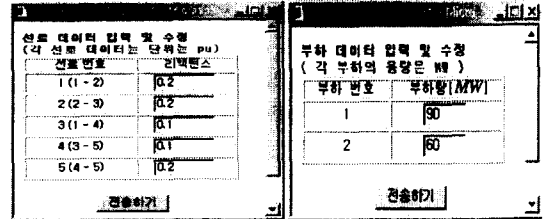


그림 5.2 부하 및 선로 데이터 입력 화면  
Fig. 5.2 Input Window for Load and Line Data

4.3 초기 조류계산 수행

그림 6은 입력받은 각각의 데이터 값을 가지고 선로 제약 없이 DC 조류계산을 수행한 화면이다. 조류계산에 사용하는 데이터는 현재 이 화면에 접속하고 있는 사용자의 IP Address를 식별하여 입력 데이터를 불러오기 때문에 자신이 입력했던 데이터로 조류계산을 수행 할 수 있다.

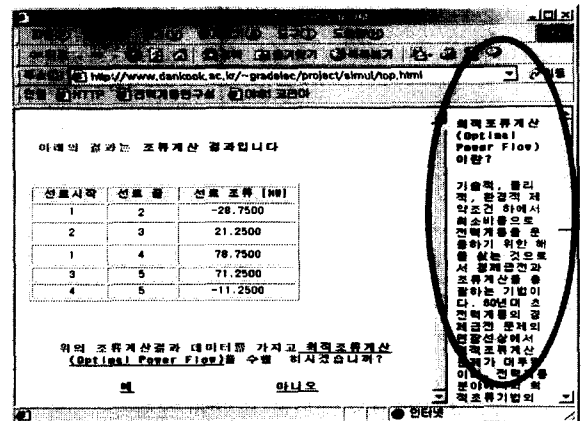


그림 6 조류계산 결과 화면  
Fig. 6 Result Window of Power Flow

또한 오른쪽 프레임 공간을 이용하여 해당화면의 핵심이 되는 용어는 HTML의 하이퍼링크 기법을 이용하여 설명하였다. 그림 6은 조류계산의 결과 값을 보여주는데 결과 값으로는 각 선로의 선로조류를 보여주고, 최적조류계산 수행의 가부를 묻는다. 만약 최적조류계산을 수행하게 되면 선로 한계용량 위반 여부 검사를 하며, 그렇지 않을 경우 프로그램의 초기화면으로 돌아간다.

4.4 최적조류계산 수행

최적조류계산에는 선로에서의 최대 허용 용량이 제약조건으로 포함되는데 본 논문에서 개발한 프로그램은 정해진 선로조류 제약량 뿐만 아니라 그림 7과 같이 사용자가 여러 상황에서 모의실험을 할 수 있도록 선로조류 제약량을 직접 입력할 수도 있다. 본 사례연구에서는 3번 5번 모선 사이의 선로(4번 선로)에 혼잡이 발생했을 경우를 고려했다. 4번 선로의 선로조류는 그림 7에서와 같이 71.25MW지만 한계용량

을 60MW로 지정했기 때문에 4번 선로에 혼잡이 발생하게 된다.

선로시작	선로종	선로조류 [MW]
1	2	-29.7500
2	3	21.2500
1	4	78.7500
3	5	71.2500
4	5	-11.2500

선로시작	선로종	선로조류제한 [MW]
1	2	100
2	3	100
1	4	100
3	5	60
4	5	60

그림 7 선로 한계용량 위반 여부 검사 화면  
Fig. 7 Window for Line Capacity check

이와 같이 4번 선로에 혼잡이 일어났을 경우 최적조류계산을 수행하면 각 선로의 선로조류와 최적 발전력은 그림 8과 같다. 7000으로 혼잡을 발생시키고 있음을 알 수 있다.

선로시작	선로종	선로조류 [MW]
1	2	-26.7847
2	3	41.7847
1	4	90.0000
3	5	60.0000
4	5	0.0000

발전기 번호	발전력
1	63.2353
2	68.5294
3	18.2353

그림 8 최적조류계산 수행 결과 화면  
Fig. 8 Result Window of OPF

그림 8에서는 혼잡이 일어났던 선로에서 혼잡이 해결된 것을 확인 할 수 있다. 이렇게 선로조류 제약조건을 만족시키기 위해서 재경제급전(Redispatch) 과정을 거치게 되며, 재경제급전에 의해 추가로 소요되는 비용은 각 모션별로 다르게 나타나며 이를 모션한계비용(Nodal Marginal Cost)이라 하고 혼잡비용(Congestion Cost)은 모션한계비용을 이용하여 계산된다.

4.5 혼잡비용 계산

각 모션의 한계비용은 식(2)에서와 같이 혼잡비용이 계통 한계비용( $\lambda$ )에 추가되어 나타나고 발전소가 수령해야 할 금

액과 수용가가 지불해야 할 모션별 비용은 다음과 같이 계산된다.

- 1번 모션 = 한계비용 × 발전량 =  $8.729 \times 63.24 = 551.99$
- 2번 모션 = 한계비용 × 발전량 =  $8.521 \times 68.53 = 565.43$
- 3번 모션 = 한계비용 × 발전량 =  $7.773 \times 18.24 = 141.74$
- 4번 모션 = 한계비용 × 부하량 =  $8.968 \times 90 = 807.14$
- 5번 모션 = 한계비용 × 부하량 =  $9.447 \times 60 = 566.79$

따라서 수용가(4,5번 모션)가 지불해야할 전체비용은  $1373.9(=807.14+566.79)$ 이며 발전소(1,2,3번 모션)가 수령해야 하는 금액은  $1259.2(=551.99+565.43+141.74)$ 이다. 결과적으로 전체 혼잡비용은 수용가가 지불해야 할 비용에서 발전소가 수령해야 하는 금액을 뺀  $114.7(=1373.9-1259.2)$ 이다.

그림 9는 각 모션의 한계비용과 혼잡비용을 계산한 결과이다.

선로	한계비용 (CC)	선로조류	혼잡비용 (CC)
1	8.729	63.24	12.80
2	8.521	68.53	-19.97
3	7.773	18.24	21.52
4	8.968	90	100.43
5	9.447	60	0.00
합계			114.7965

발전소	한계비용 (CC)	발전량	수령액
1	8.729	63.24	551.99
2	8.521	68.53	565.43
3	7.773	18.24	141.74
합계			1259.16

그림 9 모션한계비용 및 혼잡비용 계산 결과  
Fig. 9 Result of Nodal Marginal Cost and Congestion Cost

손실이 없는 계통에서는 모션 한계비용에 혼잡관련항만 이 있기 때문에 선로양단의 한계비용은 혼잡비용의 정도를 나타낸다. 따라서 선로에서의 혼잡비용은 식 (3)에서와 같이 선로 양단 모션의 모션한계비용의 차이와 선로조류의 곱으로 계산된다.

$$CC_1 = (8.251 - 8.729) \times (-26.76) = 12.80$$

$$CC_2 = (7.773 - 8.251) \times (41.76) = -19.97$$

$$CC_3 = (8.968 - 8.729) \times (90) = 21.52$$

$$CC_4 = (9.447 - 7.773) \times (60) = 100.43$$

$$CC_5 = (9.447 - 8.968) \times (0) = 0$$

따라서 전체혼잡비용은 전체 선로에서의 혼잡비용을 합한

114.7(=  $\sum_{i=1}^n CC_i$ )이 되어 발전비용과 부하비용으로부터 계산된 값과 일치함을 알 수 있다.

### 5. 결 론

본 논문에서는 경쟁적 전력시장에서 시간대 별로 변하는 모션한계비용을 인터넷 기반에서 온라인으로 처리하는 최적 조류계산 프로그램을 개발하였다. 이 프로그램은 펄(Perl)과 자바(JAVA)를 이용하여 유닉스(UNIX) 서버에서 실행되는 CGI 프로그램으로서 전력시장 참여자의 입찰 전략측면에서 보다 효과적으로 이용될 수 있다. 또한 각 모션과 선로에서의 혼잡비용을 잠재가격과 모션한계비용을 이용하여 계산하는 방법과 의미를 분석하였다. 혼잡비용의 정보는 전력계통 참여자에게 송전시스템의 새로운 투자를 위한 신호로 이용될 수 있다. 향후 연구과제로 본 논문에서 개발한 프로그램을 기반으로 실제 전력거래 시스템의 다양한 조건과 환경을 반영할 수 있도록 개선해 나갈 것이다.

### 참 고 문 헌

[1] S. Hunt, G. Shuttlesworth, *Competition and choice in Electricity*, John Wiley & Sons, 1996.

[2] Erickson, Fritz J, *Internet Primer : Getting Started on the internet* , Richard d Irwin, 1996.

[3] M. S. Bazaraa, H. D. Sherali, C. M. Shetty, *Nonlinear Programming*, John Wiley & Sons, 1993.

[4] A. A. El-Keib, X. Ma, "Calculating Short-Run Marginal Cost of Active and Reactive Power Production," *IEEE Trans. on Power Systems*, Vol.12, No.2, pp.559-565, May 1997.

[5] T. W. Gedra, "On Transmission Congestion and Pricing," *IEEE Trans. on Power Systems*, Vol.14, No.1, pp.241-248, February 1999.

[6] Elizabeth Castro, *Perl and CGI for the World Wide Web*, Addison Wesley Longman, Inc. 1998.

[7] Larry Wall, Tom christiansen, Randal L. schwartz, *Programming Perl*, O'Reilly & Associates, 1997.

[8] 이현우, 김현국, 홍성민, *JAVA Programming Bible Ver 2*, 영진출판사, 1999.

## 저 자 소 개



이 광 호 (李 光 浩)  
1965년 12월 22일 생. 1988년 서울대 공대 전기공학과 졸업. 1990년 동 대학원 전기공학과 졸업(석사). 1995년 동 대학원 전기공학과 졸업(공학). 현재 단국대 공대 전기공학과 조교수

Tel : 02-709-2868  
E-Mail : mania49d@ns.dankook.ac.kr



정 재 옥 (鄭 在 玉)  
1976년 6월 10일 생. 1999년 안양대 공대 전기공학과 졸업. 2001년 단국대 대학원 전기공학과 졸업(석사). 현재 (주)오주에스넷 부설 선술기술연구소 연구원  
Tel : 02-2203-1661  
E-Mail : jaeok@oju.co.kr