

CADPAD 프로그램의 알고리즘 분석 성과 및 국산화 개발 방향

이재봉, 박창호, 김준오
한전 전력연구원

An Analysis on Algorithm of CADPAD Program and Development of KEPCO Version

J. B. Lee, C. H. Park, J. O. Kim
Korea Electric Power Research Institute

Abstract - CADPAD program has been used as a distribution planning tool in KEPCO since 1989. In recent we've been upgraded the I/O modules. Now we analyze the key algorithm of FEEDERSITE module. The objective function is represented by the sum of the multiples of power flow and cost. To minimize the objective function, it is used to Linear Programming algorithm. We will show this algorithm in this paper.

1. 서 론

한국전력공사에서는 1989년 배전계획 수립 업무의 효율화와 정확성을 기하기 위하여 CADPAD 전산 패키지를 도입하였다. 그 동안 전산환경의 급속한 변화로 인하여 현재까지 지속적으로 I/O 모듈의 개선작업을 진행해 왔다. 그러나 CADPAD 시스템의 핵심 프로그램을 분석하여 개선시키지 않고는 현장업무에 밀착하여 배전계획을 수립하는 것이 곤란하다고 판단되었으며 따라서 핵심 모듈인 FEEDERSITE의 알고리즘을 분석하였다.

FEEDERSITE의 프로그램은 크게 데이터 입력부분, 데이터 점검부분, Linear Program 형태로 구성하는 부분, Linear Programing에 의한 최적화 부분, 결과 점검부분, 결과 출력부분으로 구성되어 있다.

선택할 수 있는 최적화 방법은 3가지로 주어진다. 첫째, 신뢰도 즉 부하율을 최소화하는 방법으로 비용은 선로의 길이에 비례하며, 둘째, 전압강하를 최소화하는 방법으로 선로의 임피던스에 비례하며, 셋째, 자본 및 손실을 포함한 모든 비용을 최소화하는 방법으로 여기에는 선형근사화에 의하여 총 비용이 계산되며, 배전계획 목적에 따라 최소화방법을 선택함으로써 목적함수가 달라지게 된다. 이러한 내용을 본 논문에서 소개하고 한전의 실무에 맞도록 재개발하는 방향을 모색한다.

2. 본 론

2.1 Feeder Network 최적화모델

feeder 네트워크는 배전선로에 사용되는 배전선의 선종에 관한 데이터와 각 feeder segment의 길이 그리고 그 segment에 연결된 양쪽의 부하 데이터로 이루어진다. 이때 각 feeder가 현재 운전중인지 신설예정인지에 대한 정보는 feeder 데이터에 명시된다. 변전소 데이터는 기설 및 신설 변전소의 용량과 위치로 구성되며 변전소의 용량은 기설 혹은 신설되는 주 변압기뱅크의 총 용량합계로 나타낸다.

2.1.1 Network 조류 모델

변전소로부터 각 부하 지점에 최소의 비용으로 전력을 공급하기 위한 문제를 다음과 같은 형태로 표현할 수 있다.

$$\text{Minimize } \sum_{k \in A} X_k C_k \tag{1}$$

$$\text{단, } \sum_{k \in IN_j} X_k - \sum_{k \in OUT_j} X_k = a_j - b_j; \text{ for all } j \in N \tag{2}$$

$$0 \leq X_k \leq U_k \text{ for all } j \in A \tag{3}$$

여기에서,
A : node를 연결하는 모든 feeder segment의 집합
N : 모든 부하점 또는 bus의 집합
X_k : feeder segment *k*의 부하 (kVA)
C_k : segment *k*를 따라 전력을 전송하는 비용
U_k : segment *k*에 전송할 수 있는 최대 전력(kVA)
IN_j : bus *j*로 전력을 보내는 feeder segment의 집합
OUT_j : bus *j*로부터 전력을 내보내는 feeder segment의 집합
a_j : bus *j*의 공급용량
b_j : bus *j*에서의 부하량

2.1.2 최적화 모델의 선택

Network 조류모델에 의하여 최적화모델은 해석자의 의도에 따라 다음과 같은 세 가지의 함수로 표현이 가능하다. 첫째, 부하율(load moment)을 최소화하고자 한다면 feeder segment의 길이에 비례한다. 두 번째, 전압강하의 합을 최소화하고자 한다면 feeder segment의 저항에 비례한다. 세 번째로 총 비용을 최소화하기 위해서는 다음 식(4)과 같이 근사화된다.

$$CP = FK + 10^{-5} DRP^2 \tag{4}$$

여기에서,
C : feeder 종류에 따른 단위km당 투자비용
P : 선로조류
F : 선로건설 또는 보강의 자본비용
K : 자본비용을 연간 고정비용으로 환산하기 위한 계수
D : 연간 손실비용
R : feeder 종류에 따른 저항

여기에 변전소의 신설 및 용량변경 비용이 추가되면 배전계통의 전체 비용이 계산된다. 따라서 세 가지 목적함수는 각각 다음과 같이 표현된다.

$$\text{Minimize } \sum_{k \in A} X_k (\text{Distance})_k \text{ for 부하율} \tag{5}$$

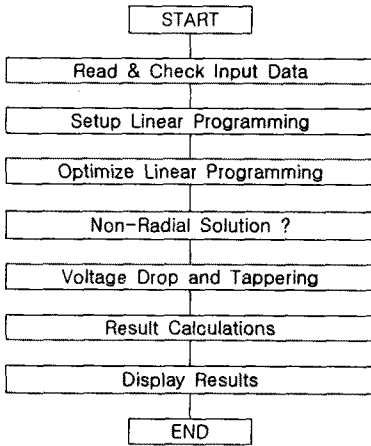
$$\text{Minimize } \sum_{k \in A} X_k (\text{Resistance})_k \text{ for 전압강하} \tag{6}$$

$$\text{Minimize } \sum_{k \in A} X_k (\text{Cost})_k \text{ for 총비용} \tag{7}$$

CADPAD의 배전계획 process에서는 최적조류계산이 수행된 다음 최적의 feeder segment의 종류와 회선수가 결정되고 그에 따라 투자비용이 계산된다.

2.2 배전계획모듈의 플로우

CADPAD 프로그램의 배전계획 모듈인 FEEDER-SITE의 개략적인 플로우에는 다음과 같다.



2.2.1 Read & Check Input Data

FEEDERSITE모듈의 첫 번째 프로세스에서는 I/O 프로그램으로부터 다음과 같은 데이터를 읽어들이고 최적화 루틴에 적합하지 않은 것들을 점검한다.

표. 1 Input Data

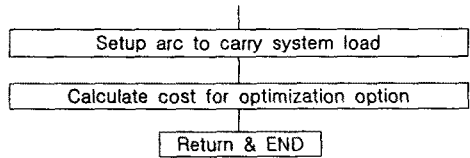
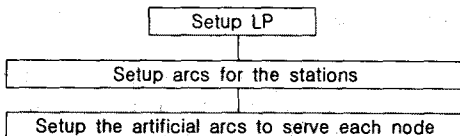
Data Type	Data
Control Parameter	- Power Factor - Optimization Option - Percent Voltage Drop Limit - Percent Load Ratio
Feeder Type	- Capacity - Resistance & Reactance - Capital Cost
Bus	- Coordinate & Demand Year - Loads(kW, kVAR)
Feeder Segment	- Type Number - Phase - Length
Substation	- Coordinates & Built Year - Voltage - MVA Rating - Capital Cost
Regulator Type	- Capital Cost - Voltage Rise

표. 2 처리가능 용량

Data	처리 용량
Feeder Types	100 types
Bus Locations & Loads	7,000 buses
Feeder Segments	14,000 segments
Substations	100 substations
Regulator Types	100 Types

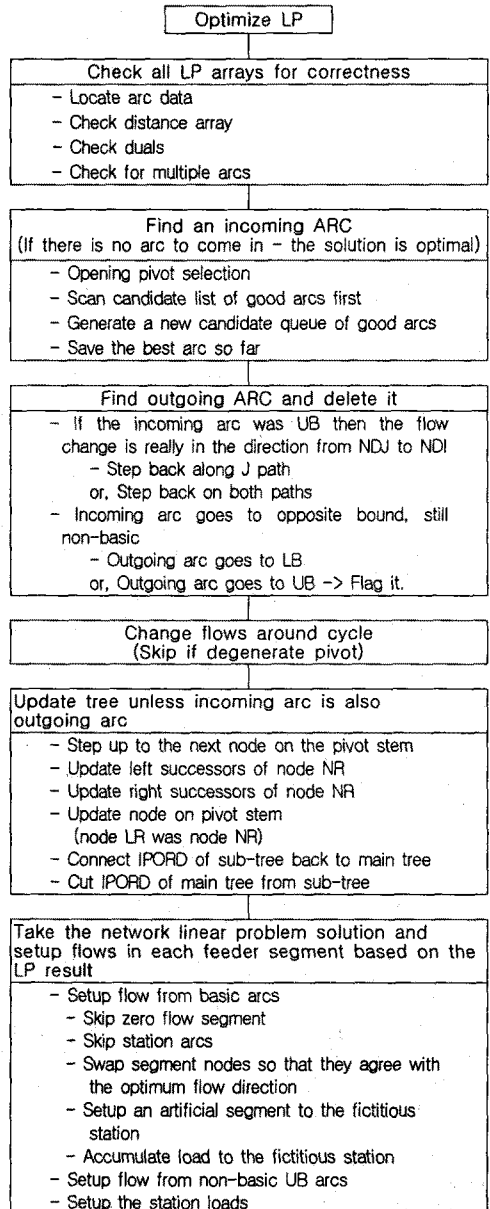
2.2.2 Setup LP

Setup LP 루틴에서는 입력된 데이터와 최적화 조건에 따라서 다음과 같이 데이터를 가공한다.



2.2.3 Optimize LP

Setup LP 루틴에서 최적화에 알맞은 데이터 구조로 구성된 다음 LP 프로그램의 해법인 Revised Simplex Algorithm에 의하여 최적화 과정이 수행된다. Simplex Algorithm은 Linear Programming의 해법으로 간단한 Pivot 연산을 통하여 목적함수의 최적값을 찾아가는 알고리즘이다. 이 알고리즘은 OR 이론을 설명한 짧은 저서에 설명되어 있다.



Check all LP arrays for correctness

- Locate arc data
- Check distance array
- Check duals
- Check for multiple arcs

Find an incoming ARC
(If there is no arc to come in - the solution is optimal)

- Opening pivot selection
- Scan candidate list of good arcs first
- Generate a new candidate queue of good arcs
- Save the best arc so far

Find outgoing ARC and delete it

- If the incoming arc was UB then the flow change is really in the direction from NDJ to NDI
 - Step back along J path
 - or, Step back on both paths
- Incoming arc goes to opposite bound, still non-basic
 - Outgoing arc goes to LB
 - or, Outgoing arc goes to UB -> Flag it.

Change flows around cycle
(Skip if degenerate pivot)

Update tree unless incoming arc is also outgoing arc

- Step up to the next node on the pivot stem
- Update left successors of node NR
- Update right successors of node NR
- Update node on pivot stem (node LR was node NR)
- Connect IPORD of sub-tree back to main tree
- Cut IPORD of main tree from sub-tree

Take the network linear problem solution and setup flows in each feeder segment based on the LP result

- Setup flow from basic arcs
- Skip zero flow segment
- Skip station arcs
- Swap segment nodes so that they agree with the optimum flow direction
- Setup an artificial segment to the fictitious station
- Accumulate load to the fictitious station
- Setup flow from non-basic UB arcs
- Setup the station loads

Develops the sequence of segments as if one was walking away from the substations and down each feeder

- Flag segment that carries the bulk of load to each node
- Flag primary feed segments by negating LORDR but skip over any segments feeding a substation node
- Skip over any segments that do not have any feeder load
- Save the number of segments feed from this node
- Reorder LORDR into groups of segments pointing to successors
- first save non-primary feed segments
- (now save primary feed segments)
- now save segments in preorder sequence
- Skip over station that did not pick up load
- will return to this node (ND) later save in stack
- Continue down the feeder since this is a primary feed segment
- Back-track to last node in stack no more successors to ND

Return & END

2.2.4 Non-radial Solution

이 루틴은 non-radial solution일 경우 limiting segment를 제거함으로써 radial solution으로 조작한다.

2.2.5 Voltage Drop & Tapering

이 루틴에서는 각 node에서 전압강하를 계산하고 필요에 따라 저전압문제를 해결하기 위하여 선로를 용량에 따라 보강한다.

Compute the voltage drops at each node

Take the LP solution and determine necessary re-conductoring to resolve low-voltage problems (Uses a dynamic programming approach)

- Determine if there is any low voltage problem
- End a tree, go through voltage tapering if there is any low voltage
- Define the feeder for the voltage tapering algorithm
- Perform VT algorithm.
- Calculate F(S) and A
- Calculate necessary re-conductoring and new VDs
- Reconductoring not adequate. Display warning message
- Recalculate voltage drop of the buses omitted at feeder end for the VT algorithm
- Start a new tree
- Do the last segment
- Recalculate the voltage drop of the substation

2.2.6 Result Calculations & Display

이 루틴에서는 최적화 과정의 결과에 따라 보강된 선로의 feeder type에 따라 손실과 길이, 그리고 선로의 개폐상태 등을 계산하고 그 결과를 출력한다.

사용자는 FEEDERSITE 모듈의 계산결과를 확인하고 실제의 상황과 비교한 후 선로의 개폐상태를 조정할 수 있다. 이렇게 하여 사용자의 목적에 따라 최적의 배전계획 결과를 얻을 수 있다.

3. 결 론

본론에서 제시한 배전계획 알고리즘의 핵심 최적화 기법은 Linear Programming 문제를 해결하기 위한 Revised Simplex Method 이다.

지금까지 한전에서는 CADPAD의 FEEDERSITE 모듈의 계산을 통하여 배전사업소의 배전계획을 수립해 왔다. 그러나 발전하는 전산환경의 변화에 대응하기 위해서는 CADPAD 프로그램의 핵심 algorithm을 자체의 기술로 확보해야 한다. 이러한 배전계획 최적화 알고리즘의 분석을 통하여 배전계획 알고리즘에 대한 기술을 확보할 수 있었으며 새로운 알고리즘의 개발에 대한 가능성을 제시하였다.

앞으로 한전에서는 본 알고리즘을 더욱 면밀히 분석하고 국내의 배전선로 환경에 부합하는 새로운 배전계획 알고리즘을 개발해 나갈 것이다.

(참 고 문 헌)

- [1] 한전 전력연구원, "전산시스템을 이용한 배전계획 최적화 연구" 최종보고서, 1995
- [2] ABB Power Inc., "FEEDERSITE Program User's Manual", 1987
- [3] FEEDERSITE Program Source Code File
- [4] 박순달, "OR 프로그래밍", 1982
- [5] 송길영, "제통해석이론의 기초와 응용", 1984