

객체 지향 기법에 기반을 둔 보수 계획 수립에 관한 연구

박영문 김진호 박종배 원종률

서울대학교 전기공학부

A Study on the Maintenance Scheduling based on the Object-Oriented Programming

Young-Moon Park Jin-Ho Kim Jong-Bae Park Jong-Ryul Won

Department of Electrical Engineering, Seoul National University

ABSTRACT

This paper is concerning on a study on the object-oriented programming and its application to maintenance scheduling. The concept of object-oriented programming enables us to modify and reuse software with much ease. By introducing object-oriented programming to maintenance scheduling, we can develop a hierarchical and reusable software in maintenance scheduling. The maintenance scheduling problem becoming more and more large and complex can be dealt with the concept of object-oriented technique and we hope this concept will give a reasonable solution. And evolutionary computation will be developed as a optimization technique.

I. 서론

대상 시스템의 대형화/복잡화 추세에 따라, 이를 모델링하고 해석하는 프로그램도 점점 대형화되고 있다. 그런데 데이터와 그 데이터를 이용하는 함수가 복잡하게 얽혀있는 기존의 프로그래밍 방법으로는 이러한 대형 시스템에 적용했을 경우 한계에 부딪히게 된다. 프로그래밍하기가 어려우며, 이미 작성된 프로그램에 대해서는 수정이 어렵고, 재사용이 거의 불가능하다. 이러한 한계점을 극복하기 위해 새로운 개념인 객체-지향 프로그래밍(Object-Oriented Programming) 기법이 도입되었다. 객체 지향 프로그래밍 기법이란 다루고자 하는 대상 시스템을 이루고 있는 모든 개별 요소를 객체(Object)로 표현하고, 이러한 객체를 통해 전체 시스템을 모델링하는 방법을 의미한다. 기존의 프로그래밍 방법이 데이터를 처리하는 함수 중심의 사고방식이라면, 객체 지향 프로그래밍 방법은 데이터와 데이터를 처리하는 함수 두 가지를 동시에 하나의 패키지 개념으로 묶은 객체 중심의 사고방식이라고 볼 수 있다. 이러한 객체 지향 프로그래밍 기법은 전력 계통 분야에서도 활발히 연구되고 있으며, 본 논문에서는 진화 연산을 이용한 보수 계획 연구에 객체 지향 개념을 도입하였다.

II. 보수 계획

전력 계통 분야에서 신뢰도 문제는 고려해야 할 매우 중요한 사항이다. 신뢰도 문제는 전력 사업에서 전력 생산과 소비 문제에 모두 관련되어 있으며, 그 중요성

이 점점 더 높아지고 있다. 보수 계획은 이런 신뢰도에 상당한 관련성을 가지는 문제로 전력 계통의 운용에 커다란 영향을 미친다. 따라서 보수 계획은 시스템의 신뢰도 문제로 오랫동안 연구되어 왔으며, 여러 가지 방법에서의 접근이 시도되어 오고 있다. 보수 계획이란 주어진 기간 내에 각 발전기의 보수 시기를 결정하는 최적화 문제이다.

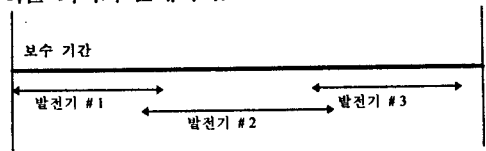


그림 1 보수 계획

보수 계획에 있어서 고려하는 목적 함수(Cost Function)로는 비용(Cost), 신뢰도(Reliability), 비용+신뢰도 등이 있으나, 본 논문에서는 신뢰도를 목적 함수로 채택하였다. 신뢰도 목적 함수에는 확률적(probabilistic)인 고려를 하지 않는 예비력(Reserve Capacity)/예비율(Reserve Rate) 최대화, 예비력/예비율 평활화(levelize) 등이 있으며, 발전기의 확률적 사고(Random Outage)를 고려하는 공급 지장 확률(LOLP, Loss of Load Probability) 최소화/평활화, 공급 지장 에너지(EENS, Expected Energy Not Served) 최소화 등이 있다. 본 논문에서는 신뢰도와 발전기의 확률적 사고를 고려할 수 있는 공급 지장 확률 최소화 (min LOLP) 를 목적 함수로 채택하였다.

III. LOLP

LOLP 란 수요(Demand)가 설비 용량(Installed Capacity)을 초과할 확률이다. 수요가 설비 용량을 초과할 경우 공급에 지장이 있기 때문에, 공급 지장 확률이라고도 한다. LOLP 는 신뢰도와 직접적인 관련이 있으므로, 이를 최소화함으로써 신뢰도를 향상시킬 수 있다.

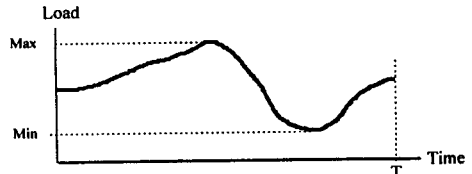


그림 2 시간대별 부하 곡선

위의 그림 2와 같이 시간대별 부하(수요) 곡선이 주어진 경우에 부하를 크기 순으로 나열하면 아래 그림 3과 같은 부하 지속 곡선(Load Duration Curve)을 얻을 수 있다. 부하 지속 곡선 아래 부분의 넓이는 공급해야 하는 에너지를 의미한다.

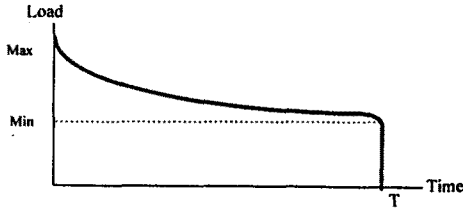


그림 3 부하 지속 곡선

그림 3 부하 지속 곡선의 가로축과 세로축을 서로 바꾸고, 세로축을 시간 T로 나누어 조정(Scaling)하면, 가로축은 수요를 나타내며 세로축은 수요가 가로축의 값을 초과할 확률을 의미하게 되는 그림 4와 같은 정형화되고 전도된 부하 지속 곡선(Normalized Inverted Load Duration Curve)을 얻을 수 있다. 일반적으로, 부하 지속 곡선은 그림 4와 같은 정형화되고 전도된 부하 지속 곡선을 의미한다.

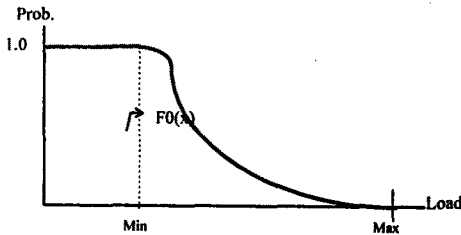


그림 4 정형화/전도된 부하 지속 곡선

지금까지는 모든 발전기의 고장 정지 확률(Forced Outage Rate)이 0(zero)이라는 가정하에 전개하였지만, 각 발전기의 고장 정지 확률을 고려하게 되면 부하 지속 곡선의 수직이 불가피해진다. 즉, 각 발전기마다 다른 부하 지속 곡선을 가지게 되고, 이 곡선을 등가 부하 지속 곡선(ELDC, Equivalent Load Duration Curve)이라고 한다.(그림 5)

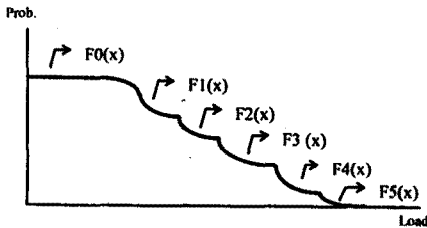


그림 5 등가 부하 지속 곡선

각 발전기의 고장 정지 확률을 q_i 라고 하면, i 번째 발전기의 등가 부하 지속 곡선 $F_i(x)$ 는,

$$F_i(x) = p_i * F_{i-1}(x) + q_i * F_{i-1}(x - C_i)$$

$$p_i = 1 - q_i,$$

$C_i = i$ 번째 발전기의 용량

따라서, 현재의 시설 용량(IC, Installed Capacity)을 알면 수요가 시설 용량을 초과할 확률인 LOLP를 계산할 수 있다.

IV. Evolutionary Programming

Simulated Annealing(SA), Genetic Algorithms(GAs)과 같은 확률적 탐색 알고리즘(stochastic search algorithm)으로서 Evolutionary Programming가 연구되고 있다.

SA의 경우, 관련된 제어 변수를 조정하기 어려우며, 계통에 적용하기가 적당하지 않다. 또한 GAs는 코딩(coding)과 디코딩(decoding)에 시간이 많이 할애된다. EP는 2진수로 코딩/디코딩하지 않고, 실제 변수를 사용하여 개발한다.

다음은 EP의 개념도이다.

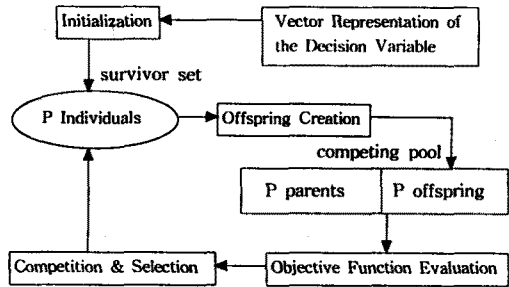


그림 6 schematic diagram of e.p.

Step 1. Vector Representation of Decision Variable

결정해야 하는 변수를 n 차원 벡터 p 로 표현해야 한다. 이 때, 각 벡터 p 는 진화된 인구(population)의 한 개인이다.



$x_j = j$ 번째 발전기의 보수 시작 주. $1 < x_j < 52$

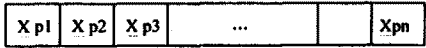
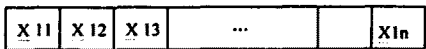
Step 2. Initialization

초기 인구의 설정은 무작위 추출을 통해 이루어지게 되고, 각 변수가 취할 수 있는 전체 값을 일정한 분포로 가지게 된다.



Step 3. Creation of Offspring

초기화에 의해 설정된 부모 세대의 인구로부터 다음 세대인 자손이 생성된다. 자손 세대는 부모 세대의 인구에 가우시안 값(Gaussian random variable)을 더해줌으로써 생성된다. 그 결과 부모 세대의 인구와 자손 세대의 인구가 함께 다음 세대를 생성하기 위한 경쟁에 들어간다.



Step 4. Competition & Selection

새로운 다음 세대는 부모 세대와 자손 세대의 인구 가운데서 확률적으로 선택되어진 개인들의 모임으로 생성된다.

부모 세대 N개의 인구와 자손 세대 N개의 인구를 합한 총 2N개의 인구에서, 하나를 선택하고 나머지 2N-1개의 인구 가운데서 N개를 임의로 선택한 후, 처음 선택한 것과 목적 함수를 비교하여 N개보다 몇 번 나은 목적 함수를 가지는지 그 횟수를 기억한다. 이 같은 작업을 2N개의 모든 인구에 대해 시행한다. 그리고 난후, 기억된 횟수가 높은 순서대로 N개의 인구를 선택하여 새로운 세대의 부모가 된다.

Step 5. Stopping Rule

목적 함수 값이 현저히 좋아지지 않거나, 이미 정해진 횟수를 지나게 되면, 새로운 세대를 생성하는 작업은 멈추게 된다.

V. Object-oriented Programming

본 논문에서 사용할 객체 지향 프로그래밍 기법은 새로운 클래스를 정의함으로써 그 의미를 지닌다. 새롭게 정의된 클래스는 각 발전기의 데이터와 LOLP 등의 함수로 이루어진 Generator 클래스와 각 세대의 인구에 대한 데이터와 초기화, 자손 생성, 경쟁, 선택 등의 함수로 이루어진 Population 클래스이다.

VI. 사례 연구

다음과 같이 발전기 4대로 구성된 계통에 적용하였다.

발전기 번호	용량	고장 정지율	보수 기간
1	25	0.01	1
2	50	0.02	1
3	75	0.01	1
4	100	0.02	1

표 1. 발전기 자료

주	1주	2주	3주	4주	5주
Load	133	144	117	129	101

표 2. 수요 자료

발전기 4대는 용량이 25[MW]씩 증가하도록 하였고, 보수 기간은 모두 1주씩 하도록 하였으며, 고려하는 총 보수 기간은 5주로 하였다.

본 논문에서 사용한 E.P.에서는, 4차원인 초기 부모(Parent) 세대의 인구(Population)수를 5개로 하였고, 목적 함수로는 LOLP를 최소화하는 것으로 하여 새로운 세대의 인구를 확률적인 방법으로 경쟁, 선택하였으며, 새로운 부모(Parent) 인구의 생성 횟수는 5회로 제한하였다.

각 주의 LOLP을 Lolp(j)라 하고, 첫째 주부터 마지막 주까지의 LOLP(j)을 모두 더한 값을 LOLP라고 하면 본 논문에서 고려된 목적 함수는 min LOLP이다. 다음은 새로운 부모 세대를 생성하는 횟수와 인구의 개수에 따른 LOLP의 변화를 보여주는 그림이다.

생성 횟수	1	2	3	4	5
LOLP1	0.548	0.574	0.431	0.475	0.423
LOLP2	0.582	0.496	0.412	0.423	0.396

표 3 LOLP

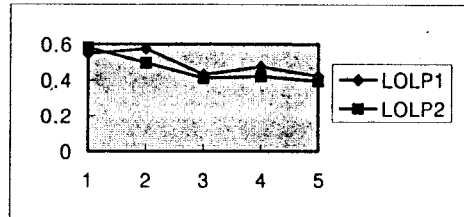


그림 7 LOLP

VII. 결론

객체 지향 프로그래밍 개념을 도입함으로써, 시스템을 실제 상태와 비슷한 모습으로 모델링할 수 있었고, 각 부분간의 독립성이 유지되는 장점을 갖게 되었다. 또한 Evolutionary Programming 기법은 기존의 Branch/Bound, Dynamic Programming 기법에 비해 계산 시간과 계산량을 줄이는 장점을 지니며, 표 3, 그림 7에서 보는 바와 같이 인구의 개수를 증가시키고, 생성 횟수를 증가시키기에 따라 LOLP의 값이 줄어드는 것을 알 수 있다.

VIII. Reference

- [1] X.Wang, J.R.McDonald, "Modern Power System Planning", McGRAW-HILL BOOK COMPANY
- [2] R.L.Sullivan, "Power System Planning", McGRAW-HILL BOOK COMPANY
- [3] James Rumbaugh, Michael Blaha, "Object-Oriented Modeling and Design", PRENTICE HALL
- [4] Timothy Budd, "Object-Oriented Programming", ADDISON-WESELY PUBLISHING COMPANY
- [5] E.Z.Zhou, "Object-oriented Programming, C++ and Power System Simulation", IEEE Trans. on Power System, Vol. 11, No. 1, February 1996
- [6] D.B. Fogel, "An introduction to simulated evolutionary optimization", IEEE Trans.on Neural Network, Vol.5, No.1, Jan. 1994, pp. 3-14