

유전알고리즘을 이용한 보수계획수립에 관한 연구

정정원 김정의
경성대학교

Maintenance Scheduling using A Genetic Algorithms with a new crossover operator

Jung-Won Jung Jung-Ik Kim
Dept. of Electrical Engineering, Kyungsung University

Abstract - Maintenance scheduling is one of mid-term scheduling problems of power systems. There have been many methods for this problem, but there is no effective way to treat all the generators simultaneously except evolutionary algorithms. In this paper, we apply GA to the maintenance scheduling problem. And we proposed new crossover operator (BOX type crossover) to improve searching ability of GA. Satisfactory results are obtained by GA with proposed crossover operator.

$$F_i(x) = p_i \cdot F_{i-1}(x) + q_i \cdot F_{i-1}(x - C_i) \quad (2)$$

여기서,
 p_i = i 번째 발전기의 가동율 ($1 - q_i$)
 q_i = i 번째 발전기의 고장정지율
 C_i = i 번째 발전기의 발전용량
 F_i = i 번째 발전기까지의 고장정지율 고려한
 등가부하지속곡선

1. 서 론

발전기의 보수계획은 전력계통의 경제적 운용과 신뢰도 측면에서 매우 중요한 문제이다. 특히, 최근의 전력계통은 많은 수의 발전기를 보유함으로써 보수계획 수립이 더욱 복잡해지고 있다. 이를 해결하기 위한 적절한 최적화 기법이 필요하다. 그래서, 본 논문에서는 유전알고리즘을 이용하여 보수계획 수립하고자 한다.

보수계획 문제는 보수비용, 발전비용등의 비용 또는 공급예비율, 공급지장화률(LOLP)등의 신뢰도를 목적함수로 하는 최적화 문제로 취급되어 연구되어 왔다. 비용을 목적함수로 두는 경우 대부분 발전비용을 그 대상으로 하는데, 이는 많은 근사화 과정이 필요하거나 계산량이 많은 시뮬레이션을 필요로 한다. 그래서, 통상적으로 보수계획에서는 신뢰도를 목적함수로 두고, LOLP나 공급지장에너지를 평활화하거나 최소화한다. 따라서, 본 연구에서는 LOLP의 최소화를 목적함수로 선택하였다.

보수계획 문제를 최적화하는 기법으로서 동적 계획법, 정수 계획법, 분지한정법 등이 적용되어왔다. 그러나 이러한 기법들은 전 발전기를 동시에 고려할 수 없어 전역 최적해를 구할 수 없는 단점이 있다. 본 논문에서는 전역적 최적해를 찾을 수 있는 유전알고리즘을 적용하되 새로운 교배 방법을 적용하여 최적 보수계획을 수립할 수 있는 방안을 제시하고자 한다.

2. 유전알고리즘의 적용

2.1 목적함수

본 논문에서는 목적함수를 연간 LOLP로 사용하였다. 적합도 함수는 목적함수를 최소화하고 최대화되도록 주어진다.

$$Max \left[\frac{\beta}{(\alpha + \text{연간 LOLP})} \right] \quad (1)$$

여기서 α, β 는 적합도 함수를 조절하기 위한 파라메타이고, 연간 LOLP 계산은 다음과 같이 주어진다.

i 번째 발전기까지의 고장정지를 고려한 등가부하지속곡선은 i-1 발전기까지의 고장정지영향을 고려한 등가부하지속곡선에 i 번째 발전기의 운전상태를 상승적분하면 구해진다.

모든 발전기의 고장정지를 고려한 등가부하지속곡선의 시설용량점에 대응하는 값은 등가부하가 시설용량을 초과할 확률, LOLP를 나타낸다.

$$LOLP^j = F^j \left(\sum_{i=1}^m C_i (1 - u_i') \right) \quad (3)$$

여기서, $LOLP^j$ 는 j주의 LOLP이고, F^j 는 j주의 보수정지중인 발전기를 제외한 모든 발전기의 고장정지를 고려한 등가부하지속곡선이다. 그리고 u_i' 는 j주m이 i 번째 발전기의 운전여부를 나타낸다. 따라서, 목적함수인 연간 LOLP는 주별 LOLP를 합함으로서 구할 수 있다.

$$\text{연간 LOLP} = \sum_{j=1}^{52} LOLP^j = \sum_{j=1}^{52} F^j \left(\sum_{i=1}^m C_i (1 - u_i') \right) \quad (4)$$

2.2 유전알고리즘

2.2.1 개체

본 논문에 사용된 유전알고리즘의 모든 개체를 실수로 발생시켰다. 생성된 개체는 각 발전기의 보수가능기간 안에서 보수시작 시작점을 나타낸다.

2.2.2 재생산

재생산은 적합도 함수에 의한 룰렛과 엘리티즘을 적용하였다.

2.2.3 교배

본 논문에서는 산술 교배 연산자와 BOX형 교배 연산자를 사용하였다. 유전알고리즘의 효율은 교배 연산자에 크게 영향을 받는다. 그래서, 본 논문에서는 유전알고리즘의 효율을 높이기 위하여 BOX형 교배 연산자를 제안하였다. 기존의 산술 교배 연산자는 두 벡터의 선형 조합(Linear Combination)으로 정의된다. 즉, u_1 과 u_2 가 교배될 때 그 자손세대는 식(5)와 같이 된다.

$$\begin{aligned} u_1' &= (1 - \lambda) \cdot u_1 + \lambda \cdot u_2 \\ u_2' &= \lambda \cdot u_1 + (1 - \lambda) \cdot u_2 \end{aligned} \quad (5)$$

여기서 λ 는 0과 1사이의 난수이다.

제안된 BOX형 교배는 다음과 같이 정의된다. 두 개체가 $u=(u_1, \dots, u_k, \dots, u_n)$, $v=(v_1, \dots, v_k, \dots, v_n)$ 이면, 식 (6)과 같이 표현할 수 있다.

$$\begin{aligned} u_k' &= (v_k - u_k) \cdot \lambda_k + u_k \\ v_k' &= (u_k - v_k) \cdot \gamma_k + v_k \end{aligned} \quad (6)$$

여기서, λ_k 와 γ_k 는 0과 1사이의 난수이다.

2.2.4 돌연변이

돌연변이는 불균등 돌연변이(non-uniform mutation)를 사용하였다. 이것은 시스템의 지역적 미세 조정 작용을 한다. 주어진 부모 u 에 대하여, x_k 가 선택되어지면 자손은 $v' = [x_1, \dots, x_k', \dots, x_n]$ 으로 주어진다. x_k' 는 식(7)에 의해서 결정되어진다.

$$x_k' = \begin{cases} x_k + \Delta(t, x_k^U - x_k) \\ x_k - \Delta(t, x_k - x_k^L) \end{cases} \quad (7)$$

여기서, x_k^U 와 x_k^L 는 x_k 에 대한 상한치와 하한치이다. x_k' 의 선택은 임의로 할 수 있다. 함수 $\Delta(t, y)$ 는 다음의 수(8)과 같이 $[0, y]$ 의 값을 가지고, t (세대수)가 할수록 0에 접근한다.

$$\Delta(t, y) = y \cdot r \cdot \left(1 - \frac{t}{T}\right)^b \quad (8)$$

여기서, r 은 0과 1사이의 난수이고, T 는 총세대수이다. 그리고, b 는 불균등 정도를 결정하는 파라미터이다.

2.2.5 순서도

본 논문에서 제시된 보수계획 순서도를 그림 1.에 나타내었다.

3. 사례연구

사례연구를 위해 사용된 계통은 최대부하가 20,438 MW이고, 시설용량은 23,417MW인 계통에 본 논문의 알고리즘을 적용시켰다.

교배 방법에 따른 GA의 효율을 비교하기 위하여 두 연산자를 각각 GA에 적용시켜 비교하였다. 이때 개체수를 100, 세대수는 100, 교배확률은 0.8, 돌연변이 확률은 0.1, 상수 b 는 5를 사용하였다. 이 결과를 그림 2에 나타냈다. 그림 2로부터 제안된 BOX형 교배 연산자를 적용시킨 GA가 더 우수한 해를 구한다는 알 수 있다. 각 결과는 기존의 분지한정법[4]의 결과 [LOLP 77.20]보다는 훨씬 우수하다.

초기개체에 따른 영향을 분석하기 위해 초기개체를 변화시켜가면서 BOX형 교배를 적용한 GA에서 LOLP를 구하였다. 이때 개체 수는 50으로 하였다. 이 결과를 그림 3에 나타내었다. 그림 3으로부터 GA가 초기개체에 관계없이 잘 수렴한다는 것을 알 수 있다.

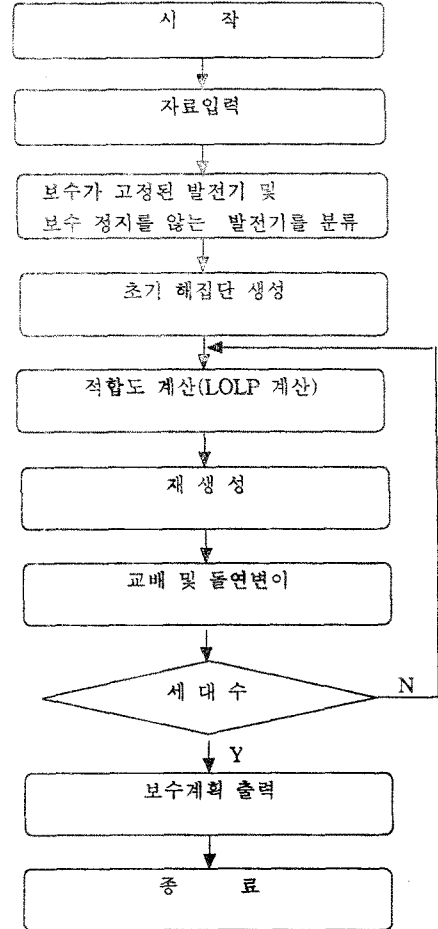


그림 1. 유전알고리즘을 이용한 보수계획 순서도
FIG 1. Flowchart of maintenance scheduling using real-coded GA

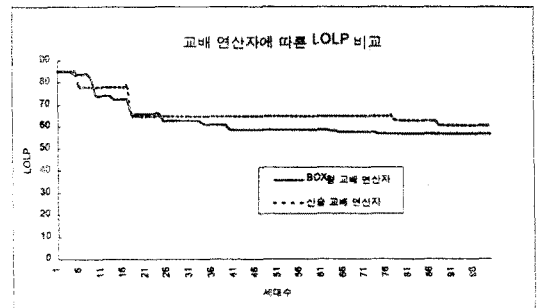


그림 2. 두 교배 연산자에 대한 LOLP 비교
FIG 2. Comparison of LOLP for two Crossover

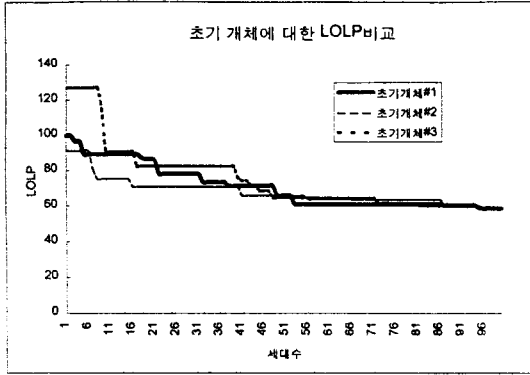


그림 2. 초기개체에 대한 LOLP 비교
 FIG 2. Comparison of LOLP for initial Population

4. 결 론

본 논문에서는 보수계획 수립문제에 LOLP를 목적함수로 두고 유전알고리즘을 적용하였다. 여기서 얻은 LOLP는 기존의 분지한정법[5]으로부터 얻는 LOLP보다 우수하였다. 본 논문에서는 새로운 교배 연산자 (BOX형 교배 연산자)를 제안하였고 기존의 산술 교배 연산자와 제안된 BOX형 교배 연산자 각각 유전알고리즘에 적용시켜 그 최적해를 비교하였다. 그 결과는 유전알고리즘의 효율을 향상 시키기 위하여 제안된 Box형 교배 연산자가 적용된 GA에서 보다 우수한 해를 얻었다.

[참 고 문 헌]

- [1] Mitsuo Gen & Runwei Cheng, "Genetic Algorithms & Engineering Design", WILEY - INTERSCIENCE
- [2] Zbigniew Michalewicz, "Genetic Algorithms + Data Structures = Evolution Program", Springer-Verlag, 1992
- [3] Aijaz Ahmad & D.P. Kothari, "A review of advances in generator maintenance scheduling", Electric Machine & Power System, Vol 6, pp373~387, 1998.
- [4] 정 정 원, "분지한정법을 이용한 발전계획의 보수계획수립에 관한 연구", Trans. KIEE, Vol 44, PP829~835, 1995

	10	20	30	40	50
HYD	-----	-----	-----	-----	-----
PMP	-----	-----	-----	-----	-----
nuc #1	-----	-----	-----	-XXXXXXXX	X-----
nuc #2	-----	XX	XXXXXXXX	-----	-----
nuc #3	-----	-----	-----	XXX	XXXXXXXX
nuc #4	-----	XXXXXXXX	XXX	-----	-----
nuc #5	-----	-----	-----	-XXXXXXXX	X-----
nuc #6	-----	XXXXXXXX	XXX	-----	-----
nuc #7	-----	XXXXX	XXXXX	-----	-----
nuc #8	-----	-----	-----	-----	-XXXXXXXX
nuc #9	XXXXXXXX	-----	-----	-----	-----
ther#1	-----	XXXX	X	-----	-----
ther#2	-----	-----	-----	-----	XXX
ther#3	-----	XXXXX	-----	-----	-----
ther#4	-----	-----	-----	XXXXX	-----
ther#5	-----	-----	XXXXX	-----	-----
ther#6	-----	-----	XXXXX	-----	-----
ther#7	-----	-----	-----	XXXX	X-----
ther#8	-----	-----	-----	-----	XXXX
ther#9	XXXXX	XXX	-----	-----	-----
ther#10	-----	-----	XXXXX	XXX	-----
ther#11	XXXXXXXX	X	-----	-----	-----
ther#12	-----	XXXXXXXX	-----	-----	-----
ther#13	XXXXX	-----	-----	-----	-----
ther#14	X	XXXX	-----	-----	-----
ther#15	-----	-----	-----	-----	XXXXX
ther#16	-----	-----	-----	XXXXX	-----
ther#17	-----	-----	XX	XXX	-----
ther#18	XXXXX	-----	-----	-----	-----
ther#19	XXXXXXXX	-----	-----	-----	-----
ther#20	XXXXXXXX	-----	-----	-----	-----
ther#21	XXXXXX	X	-----	-----	-----
ther#22	XXXXXXXX	-----	-----	-----	-----
ther#23	-----	XXXXX	-----	-----	-----
ther#24	XXX	XX	-----	-----	-----
ther#25	-----	-----	XXXXXXXX	-----	-----
ther#26	-----	XXXXXX	X	-----	-----
ther#27	XXXXX	-----	-----	-----	-----
ther#28	-----	XXXXX	-----	-----	-----
ther#29	-----	XXXXX	-----	-----	-----
ther#30	XXX	XX	-----	-----	-----
ther#31	XXXXX	-----	-----	-----	-----
ther#32	-----	-----	-----	XX	XXX
ther#33	XXXXX	-----	-----	-----	-----
ther#34	XXXXX	-----	-----	-----	-----
ther#35	-----	XXX	XX	-----	-----
ther#36	-----	-----	-----	XXXXXX	-----
ther#37	-----	XXXXX	X	-----	-----
ther#38	XXXXX	-----	-----	-----	-----
ther#39	XXXXX	-----	-----	-----	-----
ther#40	-----	-----	-----	-----	XXXXXX
ther#41	-----	-----	-----	-----	XXXXXX
ther#42	XX	XXXX	-----	-----	-----
ther#43	-----	XX	XXX	-----	-----
ther#44	-----	-----	-----	-----	-----
ther#45	-----	-----	-----	-----	-----
ther#46	-----	-----	-----	-----	-----
ther#47	-----	-----	-----	-----	-----
ther#48	-----	-----	-----	-----	XXXXXX

그림 4. BOX형 교배를 사용한 보수계획
 FIG 4. Maintenance Scheduling using BOX crossover