

유전 알고리즘을 이용한 발전기 기동정지계획수립에 관한 연구

정정원 김정익
경성대학교 전기전자컴퓨터공학부

Unit Commitment Using a Genetic Algorithm with New Crossover Operator

Jung-won Jung Jung-ik Kim
Department of Electrical Engineering, Kyungsung University

Abstract - The unit commitment is an important problem of production scheduling which determines the generating unit to in service(on/off) during scheduling period, to meet system demand and reserve requirement at minimum cost.

This paper presents an box type crossover to improve searching ability of GA, to solve unit commitment problem. Satisfactory results are obtained by GA with the proposed crossover operator.

1. 서 론

UC 문제는 최소 비용으로 부하 수요와 예비력을 만족시키면서 주어진 기간에 대하여 발전기의 기동정지계획을 결정하는 중요한 문제이다.

UC 문제의 정확한 해는 발전기들의 적절한 조합의 열거로써 구할 수 있다. 그러나, 이러한 열거는 매우 큰 수이므로 실제 발전 계통에 적용하기에는 어려움이 있다. 해결하기 위한 방법으로서 DP, Larangian Relaxation, SA, GA 등 여러 가지 방법이 적용되었으나, 계산시간이 많이 소요되거나 국부최소값에 수렴하는 문제점을 가지고 있다.

본 논문에서 UC 문제에 유전알고리즘을 적용하였으며 이 유전 알고리즘을 효율성을 향상을 위해서 새로운 박스형 교배연산자를 제시하였다. 효율성을 알아보기 위해서 convex 연산자와 선형연산자, 우선순위법과 비교하였다.

2. 본 론

2.1 발전기 기동정지계획(UC) 문제(2)

UC문제의 목적함수는 각 제약조건을 만족시키면서 총 발전비용을 최소화하는 것이다. 문제의 정의를 다음과 같이 할 수 있다.

2.1.1 목적함수

$$\text{MIN} \sum_{t=1}^T \sum_{i=1}^N [U_{it}F_i(P_{it}) + U_{it}(1 - U_{it})S_i(X_{it})] \quad (1)$$

여기서, T 총 시간
N 전체 발전기 수
U_{it} 시간 t에서 발전기 i의 상태
F_i(P_{it}) 연료비 함수
F_i(P_{it}) = a_i + b_i × P_{it} + c_i × P_{it}²
X_{it} 시간 t에서 발전기 i의 과거 상태
X_{it}=3이면 발전기 i가 3시간동안 기동을 표시한다.
S_i(X_{it}) 기동비용함수

2.1.2 제약조건

· 부하 제약조건

$$\sum_{i=1}^N U_{it} P_{it} = P'_D \quad t=1, 2, \dots, T \quad (2)$$

여기서, P'_D 시간 t에서 부하수요

· 예비력 제약조건

$$\sum_{i=1}^N U_{it} \cdot P_i^{\max} - P'_D \geq P'_R \quad t=1, 2, \dots, T \quad (3)$$

여기서, P'_R 시간 t에서 예비력

· 최소 기동, 정지 제약조건

$$\begin{cases} |X_{it}| \geq \text{Minup} & \text{if } U_{it-1}(1 - U_{it}) = 1 \\ |X_{it}| \geq \text{Mindn} & \text{if } U_{it}(1 - U_{it-1}) = 1 \end{cases} \quad (4)$$

· 발전기 용량 제약조건

$$P_i^{\min} \leq P_i \leq P_i^{\max} \quad i=1, 2, \dots, N \quad (5)$$

여기서, P_i^{min} 발전기 i의 최소 출력

P_i^{max} 발전기 i의 최대 출력

2.2 유전알고리즘을 이용한 UC 문제

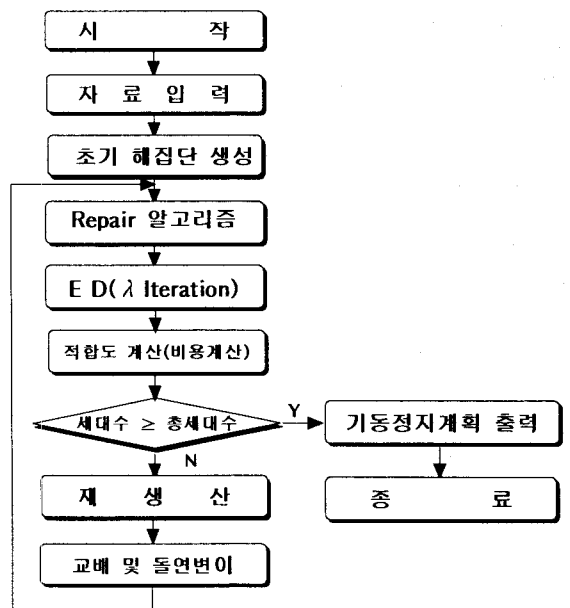


그림 1. 유전 알고리즘을 이용한 기동정지계획 순서도
Fig 1. Flowchat of UC using GA

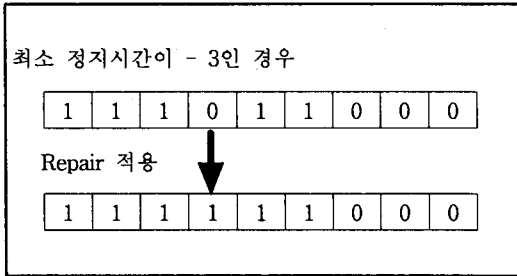
2.2.1 개체

본 논문에서는 각 개체를 $0 \sim 2^{24}$ 사이의 실수로써 발생시켰다. 각 발전기의 기동, 정지는 2진수의 0, 1로 변환해서 나타내었다.

2.2.2 Repair 알고리즘

최소 기동 및 정지 제약조건에 대해서 Repair 알고리즘을 적용시켰다. 이것은 매우 큰 탐색영역으로 인해서 많은 수행시간을 갖는다. 따라서 과도한 탐색시간을 줄이기 위해서 Repair 알고리즘을 적용시켰다.

여기서 적용된 Repair 알고리즘은 최소기동 및 정지 제약조건을 지키지 못한 시간에 대해서 발전기를 기동상태로 변환하였다.



2.2.3 적합도 계산과 재생산

적합도 함수는 목적함수를 최소화하면서 최대화되도록 설정하였다. 즉, 식 (5)로 표현할 수 있다. 그리고, 여비력 제약조건에 대하여 penalty를 적용시켰다.

$$Fitness = \left[\frac{\beta}{\alpha + Objective + Penalty} \right] \quad (5)$$

여기서, α 와 β 는 적합도 함수를 조절하기 위한 매개변수이다.

재생산은 주어진 한 세대의 개체집단에서 다음 세대의 개체집단을 만들기 위한 중간단계의 집단을 선정하는 작업이다. 이 중간단계의 집단의 선정에는 여러 가지 방법들이 존재할 수 있는데 기본적으로 적합도가 큰 개체가 발생될 가능성이 크다는 가정에 의한다. 본 연구에서는 룰렛 휠(roulette wheel)을 사용하였다.

2.2.4 교배연산자

본 논문에서는 유전알고리즘의 기능을 향상시키기 위해 새로운 박스형 교배연산자를 제시하였다. 기존의 교배연산자인 산술연산자(컨벡스 교배연산자, 선형 교배연산자)[3][4]와 비교하였다.

2.2.4.1 산술 교배연산자[3][4]

산술 교배연산자는 두 벡터의 일차결합으로 정의된다. 먼저 컨벡스 교배연산자를 살펴보면, 두 개체 S_u^t 와 S_v^t 를 교배시키면 자손 S_u^{t+1} 및 S_v^{t+1} 은

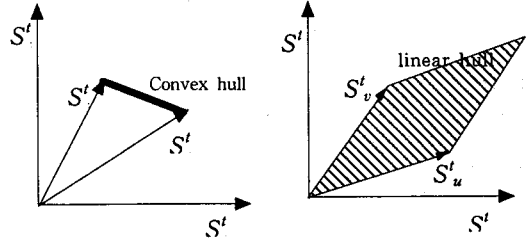
$$\begin{aligned} S_u^{t+1} &= a \cdot S_v^t + (1-a) \cdot S_u^t \\ S_v^{t+1} &= a \cdot S_u^t + (1-a) \cdot S_v^t \end{aligned} \quad (6)$$

로 나타낸다. 난수 a 는 $a \in [0, 1]$ 이다. 이를 그림으로 도식하면 새로운 개체 S_u^{t+1} 와 S_v^{t+1} 는 부모인 S_u^t 와 S_v^t 사이의 직선 영역에서 발생된다. 그림 2. (a)에 나타내었다.

선형 교배연산자[4]는 식 (7)로 표현된다.

$$\begin{aligned} S_u^{t+1} &= a_1 \cdot S_v^t + a_2 \cdot S_u^t \\ S_v^{t+1} &= a_1 \cdot S_u^t + a_2 \cdot S_v^t \end{aligned} \quad (7)$$

여기서, 난수 a_1, a_2 는 $0 \leq a_1 + a_2 \leq 2$ 이고 $a_1, a_2 \in [0, 1]$ 이다. 선형교배연산자의 해 탐색영역을 그림 2. (b)에 나타내었다.



(a) 컨벡스 교배연산자 (b) 선형 교배연산자
(a) Convex crossover (b) Linear crossover

그림 2. 산술 교배 연산자의 탐색 영역

Fig 2. Illustration showing convex hull and linear hull

2.2.4.2 박스형 교배연산자

본 논문에서 제시된 교배연산자는 박스 교배연산자, 이동 박스 교배연산자, 확장 박스 교배연산자이다. 먼저 박스 교배연산자는 식 (8)로 정의할 수 있다. 개체 $S_u^t = \langle u_1^t, \dots, u_m^t \rangle$ 와 $S_v^t = \langle v_1^t, \dots, v_m^t \rangle$ 을 교배시키면 아래와 같이 자손 $S_u^{t+1} = \langle u_1^{t+1}, \dots, u_m^{t+1} \rangle$ 및 $S_v^{t+1} = \langle v_1^{t+1}, \dots, v_m^{t+1} \rangle$ 을 구한다.

$$u_i^{t+1} = (v_i^t - u_i^t) \cdot \lambda_i + u_i^t \quad (8)$$

$$v_i^{t+1} = (u_i^t - v_i^t) \cdot \gamma_i + v_i^t$$

여기서, $\lambda_i, \gamma_i \in [0, 1]$ 이다.

따라서, 박스 교배연산자에 의해 생성된 자손의 범위는 그림 3의 (a)에 나타내었다.

또한 본 논문에서는 박스 교배연산자의 탐색 영역을 이동하거나 확장한 이동 박스 교배연산자와 확장 박스 교배연산자도 제안한다. 두 개체의 적합도 함수를 비교해서 적합도가 큰 쪽으로 탐색 영역을 이동시킨 것이다. 이는 선택된 두 개체 중 적합도가 낮은 개체에서 보다 적합도가 높은 개체 근처에서 우수한 자손이 나올 수 있는 개연성이 높다는 것을 반영한다.

이동 박스 교배는 다음과 같이 정의할 수 있다. $Fitness(S_u^t) < Fitness(S_v^t)$ 라고 하면 S_v^t 방향으로의 이동 벡터를 다음으로 둔다.

$$d = \tau \cdot (S_v^t - S_u^t) \quad (9)$$

여기서, τ 는 탐색영역의 이동 정도를 나타내는 매개변수이다.

$$\begin{aligned} S_u^t &= S_u^t + d \\ S_v^t &= S_v^t + d \end{aligned} \quad (10)$$

S'_u 와 S'_v 를 d 만큼 이동시킨 S''_u 와 S''_v 에서 박스교배연산에 의해 자손 S''_u^{t+1} 및 S''_v^{t+1} 를 구한다.

확장 박스 교배연산자는 선택된 두 개체의 적합도 함수를 비교해서 적합도 함수가 큰 쪽으로 탐색영역을 확장한 것이다. $Fitness(S'_u) < Fitness(S'_v)$ 라고 하면 S'_v 방향으로의 이동 벡터 d 를 이동박스 교배에서와 같이 구한다. 적합도가 큰 S'_v 만을 d 만큼 이동시킨 S''_v 를 구하고 S''_u 와 S''_v 를 대상으로 박스교배연산에 의해 자손 S''_u^{t+1} 및 S''_v^{t+1} 를 구한다.

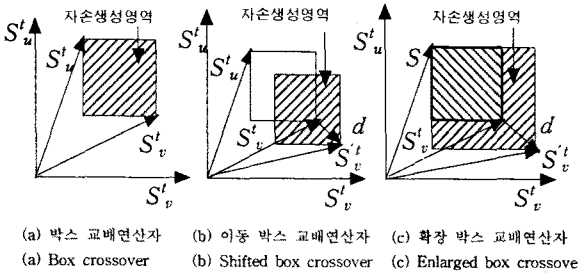


그림 3. 박스 교배연산자, 이동 박스 교배연산자, 확장 박스 교배연산자의 자손 생성 영역

Fig 3. Illustration showing offspring areas by box crossover, shifted box crossover, and enlarged box crossover

2.2.5 돌연변이 연산자

돌연변이는 불균등 돌연변이(non-uniform mutation)[3][4]를 사용하였다. 주어진 부모 S 에 대하여, u_k 가 선택되어지면 자손은 $S' = [u_1, \dots, u_k', \dots, u_n]$ 으로 주어진다. u_k' 는 식 (11)에 의해서 결정된다.

$$u_k' = \begin{cases} u_k + \Delta(t, u_k^U - u_k) \\ u_k - \Delta(t, u_k - u_k^L) \end{cases} \quad (11)$$

여기서, u_k^U 와 u_k^L 는 x_k 에 대한 상한치와 하한치이다. 어떤 개체에 대한 돌연변이 적용여부 및 u_k' 의 선택은 난수에 의한다.

$$\Delta(t, y) = y \cdot r \cdot \left(1 - \frac{t}{T}\right)^b \quad (12)$$

여기서, $r \in [0, 1]$ 이고, t 는 세대수, T 는 총 세대수이다. 그리고, b 는 불균등 정도를 결정하는 매개변수이다.

3. 사례연구

제안된 방법에 대한 사례연구에 대하여 참고 문헌의 자료를 사용하였다. 표1과 표2에 각 발전기 대한 자료와 부하 수요를 나타냈었다. 표3에 UC의 결과를 우선순위법과 각 교배연산자에 대해 비교하였다.

4. 결 론

본 논문에서는 발전기 기동정지계획 문제를 해결하기 위한 방법으로서 유전 알고리즘을 적용하였다. 적용된 유전 알고리즘의 효율성을 향상을 위해서 박스형 교배연

산자를 제안하였다. 제안된 방법(이동박스교배연산자, 확장박스교배연산자)을 이용해서 사례연구에 적용한 결과, 기존의 연산자나 우선순위법을 사용한 경우 우수한 결과를 얻을 수 있었다.

표 1 발전기 특성과 상수
Table 1 The unit characteristics and coefficients

| Unit | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|--|-------|------|-------|-------|-------|-------|-------|-------|-------|-------|
| P_{max} (MW) | 60 | 80 | 100 | 120 | 150 | 280 | 520 | 150 | 320 | 200 |
| P_{min} (MW) | 15.0 | 20.0 | 30.0 | 25.0 | 50.0 | 75.0 | 250.0 | 50.0 | 120.0 | 75.0 |
| MDT(hr) | 2 | 4 | 4 | 3 | 3 | 3 | 4 | 2 | 5 | 6 |
| MUT(hr) | 3 | 3 | 4 | 3 | 1 | 6 | 10 | 3 | 7 | 6 |
| IC(hr) | 3 | 3 | 4 | 3 | -3 | -3 | 10 | 3 | 7 | 6 |
| IP(MW) | 60 | 80 | 100 | 120 | 0 | 0 | 520 | 150 | 320 | 200 |
| a(\$) | 15.0 | 25.0 | 40.0 | 32.0 | 29.0 | 72.0 | 150.0 | 100.0 | 49.0 | 82.0 |
| b(\$/MW) | 1.400 | 1.50 | 1.350 | 1.400 | 1.540 | 1.350 | 1.395 | 1.329 | 1.264 | 1.214 |
| c(\$/MW ²) ($\times 10^{-4}$) | 51 | 40 | 39 | 38 | 21 | 26 | 13 | 14 | 29 | 15 |
| σ (\$) | 15 | 15 | 25 | 12 | 30 | 30 | 60 | 80 | 50 | 70 |
| δ (\$) | 123 | 123 | 110 | 100 | 130 | 146 | 207 | 202 | 137 | 157 |
| τ | 5 | 5 | 5 | 5 | 5 | 6 | 11 | 11 | 7 | 9 |
| SDC(\$) | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

표 2 부하 수요
Table 2 Load demand

| 시간 | 부하요수 (MW) | 예비운전력 (MW) | 시간 | 부하요수 (MW) | 운전예비력 (MW) |
|----|--------------|---------------|----|--------------|---------------|
| 1 | 1167.0 | 350.0 | 13 | 923.0 | 280.0 |
| 2 | 1097.0 | 329.0 | 14 | 910.0 | 270.0 |
| 3 | 1039.0 | 329.0 | 15 | 900.0 | 270.0 |
| 4 | 1028.0 | 300.0 | 16 | 876.0 | 260.0 |
| 5 | 1017.0 | 300.0 | 17 | 853.0 | 260.0 |
| 6 | 1051.0 | 300.0 | 18 | 829.0 | 250.0 |
| 7 | 1098.0 | 300.0 | 19 | 794.0 | 240.0 |
| 8 | 1051.0 | 300.0 | 20 | 782.0 | 240.0 |
| 9 | 1017.0 | 300.0 | 21 | 770.0 | 240.0 |
| 10 | 993.0 | 300.0 | 22 | 818.0 | 240.0 |
| 11 | 958.0 | 280.0 | 23 | 864.0 | 260.0 |
| 12 | 946.0 | 280.0 | 24 | 1167.0 | 350.0 |

표 3 결과 비교
Table 3. comparison of result

| 방법 | 비용(\$) | 개선비율(%) |
|-----------|--------------------|----------------------|
| 우선 순위법 | 48668.6100 | - |
| GA | convex 교배연산자 | 48631.4550 0.08 |
| | 선형 교배연산자 | 49033.2866 - 0.74 |
| | 박스교배연산자 | 49001.3705 - 0.27 |
| | 이동박스교배연산자 | 47956.3383 1.49 |
| 확장박스교배연산자 | 48201.0614 0.96 | |

[참고 문헌]

- [1] Subir Sen, "Optimal thermal generating unit commitment: review", Elsevier Science, Vol 20, 443~451, 1998S
- [2] Zhu Mingyu, "Using an Enhanced Genetic Algorithm to Solve the Unit Commitment Problem", IEEE, 611~614, 1997
- [3] Zbigniew Michalewicz, "Genetic Algorithms + Data Structures = Evolution Program", Springer-Verlag, 1992
- [4] Mitsuo Gen & Runwei Cheng, "Genetic Algorithms & Engineering Design", WILEY-INTERSCIENCE