

병렬유전알고리즘을 이용한 발전기의 기동정지계획

*문경준, *김형수, *박준호, **박태완, **류광렬, **정상화
* 부산대학교 전기공학과, ** 부산대학교 컴퓨터공학과

A Parallel Genetic Algorithm for Unit Commitment Problem

*K. J. Mun, *H. S. Kim, *J. H. Park, **T. H. Park, **K. R. Ryu, **S. H. Chung

*Dept. of Electrical Engineering, Pusan National University **Dept. of Computer Engineering, Pusan National University

Abstract

This paper proposes a unit commitment scheduling method based on Parallel Genetic Algorithm(PGA). Due to a variety of constraints to be satisfied, such as the minimum up and down time constraints, the search space of the UC problem is highly nonconvex.

So, we used transputer which is one of the practical parallel processors. It can give us fastness and effectiveness features of the proposed method for solving the problem. To show the effectivness of the PGA based unit commitment scheduling, we tested results for system of 5 units and we can get desirable results.

1. 서론

기동정지계획(unit commitment:UC) 문제는 각 발전기가 갖는 다양한 제약조건과 운용비용 등을 고려하여 단기 운용계획을 세우기 위해서 발전기의 조합을 결정하는 문제이다. 계통의 규모가 커짐에 따라 발전기의 조합수가 증가하기 때문에 제약조건을 만족하면서 모든 해를 탐색하는 것은 어렵다. 이러한 문제를 해결하기 위해 종래에는 dynamic programming, Lagrangian Relaxation, simulated annealing 등이 제안[1-3]되었으나 계산시간이 많이 소요되거나 극부최소값에 수렴하는 문제점을 가지고 있다.

본 논문에서는 이런 문제점을 해결하기 위해서 자연의 유전학과 진화이론에 바탕을 둔 확률적인 최적점 탐색방법인 유전알고리즘을 사용하였다.[4-6] 유전알고리즘은 복잡성 병렬화에 적합한 알고리즘이므로 여러 종류의 최적화 문제에 대해 최적해를 구하는 과정을 병렬로 처리함으로써 수행시간을 더욱 감소시키는 잇점을 가진다. 본 논문에서는 기동정지계획 문제에 병렬 트랜스퓨터(Transputer) 시스템을 이용한 병렬유전알고리즘을 사용하며, 제약조건들을 스트링으로 구성하는 방법을 제안하였다. 사례연구를 통하여 제안한 방법의 유용성을 입증하였다.

2. 기동정지계획 문제

기동정지계획 문제는 일간 혹은 수일간의 발전계획을 수립하기 위해서 발전기의 조합을 결정하여 총 발전비용을 최소화하는 것이다. 다시 말하면 전력계통에서의 기동정지계획 문제는 미래의 단기간에 대해 예측된 부하수요에 대해 발전소의 여러 제약조건들을 만족시키면서 총 발전비용을 최소화하도록 발전소의 기동, 정지계획을 결정하는 문제이다.

단, 총 발전비용은 연료 비용, 기동 비용, 정지 비용으로 구성된다. 여기서 연료비용은 열 소비율과 연료 가격에 의해 계산되며, 기동비용은 발전기가 정지한 시간에 대한 함수로 표현된다. 그리고 정지비용은 각 발전기에 대해 일정한 값으로 주어진다. 위의 발전비용과 더불어 기동정지계획문제에서 고려해야 할 제약조건은 다음과 같다.

- (1) 계통의 부하 평형(부하수요+손실)
- (2) 운전 예비력
- (3) 발전기의 초기 조건
- (4) 발전기의 최대, 최소 발전량
- (5) 최소 운전 시간, 최소 정지 시간
- (6) 발전기의 상태에 대한 제약조건
(반드시 기동되어야 하는 발전소, 발전량 MW가 고정된 발전소, 사용가능하거나 사용불가능한 발전소 등)
- (7) 발전소에서 사용가능한 연료
- (9) 기타 발전기의 제약조건

단, 제약조건 (1)은 각 발전소들의 상호 연관성을 고려하기 때문에 시스템 제약조건(system constraint)이라 하며 제약조건 (2)-(8)은 각각의 발전소가 갖는 제약조건으로 local constraint라 한다.

3. 유전 알고리즘 병렬화

유전 알고리즘 자체가 그 속성상 병렬화에 적합하다는 점을 이용하여 기동정지계획문제의 최적해를 구하는 과정을 병렬로 처리함으로써 그 수행 시간을 더욱 줄일 수 있다.[7-9] 아래에서는 우선, 본 문제를 병렬로 처리하기 위한 prototype 병렬 machine으로서 다중 트랜스퓨터(Transputer) 시스템에 대해 간략히 설명한다. 그리고 본 문제에 적합한 병렬 유전 알고리즘의 모델과 그에 대한 실제적인 구현 원리를 제시하고 있다. 끝으로 병렬 유전 알고리즘 수행 후 최종 결과를 수집 결정하는 방법에 대해 잠시 살펴본다.[9]

3.1 병렬 트랜스퓨터 시스템

본 연구에서 병렬 유전 알고리즘을 구현하게 될 prototype 병렬 machine은 프로세서 17개를 사용하고 각 프로세서마다 4 Mb RAM을 보유한 다중 트랜스퓨터 시스템으로 그림 1과 같은 구조를 가진다. 프로세서 중 1개는 root 프로세서(T₀)로서 host와의 인터페이스를 담당하며, 나머지 16개(T_c)는 실제 각종 작업의 병렬 처리에 사용되는 프로세서이다. 각 프로세서는 IMS T805 32bit Transputer이며, host 컴퓨터는 IBM호환 PC이다. 각 트랜스퓨터는 4개의 양방향 고속

interprocessor communication links를 통해 인접 프로세서와 20Mbits/sec로 background 통신이 가능하다. Interconnection network은 다양한 형태가 가능하지만 본 연구에는 2-dimensional mesh with wrap-around를 사용한다.

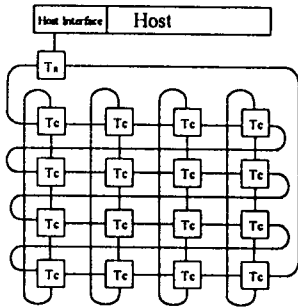


그림 1. 병렬 Transputer 시스템의 구성

3.2 병렬 유전 알고리즘의 모델과 구현

유전 알고리즘의 병렬화는 집단의 형태로 존재하는 후보해의 적합도 계산을 병렬로 처리하는데 그 주안점을 두고 있다. 이 때 적합도 계산은 각 후보해에 대해 독립적으로 이루어지는 것이므로 병렬처리를 위해 후보해 집단을 소집단으로 나누어 각 프로세서에 고르게 분배하여 병렬로 처리하면 된다.

그러나 이는 전역적으로 적용되어야 할 복제와 교차 연산까지 각 소집단 내에서만 국지적으로 이루어지도록 함으로써 유전적 탐색 성능의 저하를 초래할 소지가 커진다. 이런 문제를 해결하기 위해 소집단 간에 각각의 최우수 후보해들이 이식될 수 있도록 하는 것을 기본으로 하는 여러 병렬 모델들이 제안되어 왔다.

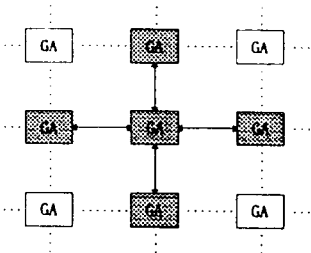


그림 2. 병렬 유전 알고리즘 모델

본 연구에서는 이들 여러 모델들이 지향하는 장점을 최대한 살리면서 본 연구에서 사용한 트랜스퓨터 기반의 병렬 컴퓨터 구조에서 최상의 성능을 보이는 새로운 모델인 병렬 SIMG 모델(Semi-Independent Medium-Grained subpopulation model)을 개발하였다. 이 모델에서는 그림 2에서 보는 바와 같이 각 트랜스퓨터 프로세서에 중규모의 후보해 소집단을 유지시키고 복제와 교차가 소집단 내에서만 국지적으로 수행되도록 하되, 각 트랜스퓨터가 네 개의 양방향 채널(channel)을 통해 이웃과 연결되어 있으므로 소집단 내의 몇몇 우수 후보해들을 이들 인접 트랜스퓨터들과 교환

하게 하여 탐색의 극지화를 막을 수 있다. 그리고 후보해의 교환 과정이 비동기적(asynchronous)으로 이루어지도록 구현하기 위해 각 트랜스퓨터에는 내 방향의 채널 각각에 대해 입력과 출력을 담당하는 프로세스(process)를 따로 마련해 두고 실제 유전 알고리즘을 수행하는 프로세스와 병렬적으로 동작하도록 하였다. 그 외에 각 트랜스퓨터 내에 후보해의 전송과 수신을 위한 버퍼를 만들어 둬으로써, 유전 알고리즘을 수행하는 프로세스는 송수신 동작에 따른 수행 지연없이 계속적으로 작업을 해 나갈 수 있도록 하였다.

결국 이런 교환 과정이 유전 알고리즘 수행시 몇 세대에 걸쳐 계속 반복됨에 따라 멀리 떨어져 있는 소집단 간에도 후보해 교환이 이루어지는 효과가 발생하므로 본 병렬 모델의 모든 소집단들이 서로 완전히 독립적으로 격리되지는 않는다. 그 외에 인접 프로세서 간에만 후보해 교환이 이루어지므로 그에 따른 통신 부담이 크지 않아 타 병렬 모델에 비해 많은 수의 후보해 교환이 가능하다. 그러므로 이 모델은 병렬 프로세서의 수가 많지 않아 각 프로세서 별 소집단의 규모가 비교적 커져야 하는 경우에도 좋은 성능을 유지할 수 있다.

이상과 같은 모델을 바탕으로 실제 구현한 병렬 유전 알고리즘의 프로그램은 병렬 트랜스퓨터 시스템을 가장 효율적으로 활용할 수 있는 Occam2 병렬 프로그래밍 언어로 작성하였다. 그리고 본 프로그램에서는 다음의 구체적인 구현 방법들을 사용하였다. Fitness scaling은 sigma scaling 방법을 사용하여 확률적 분산값을 벗어나는 worst fitness를 갖는 후보해들은 제거하였다. 선택과정은 sigma scaling에 바탕을 두고 generation gap의 정도에 따라 선택하는 Baker's selection 기법을 사용하고, 교차과정은 two-point crossover를 사용하였다.

각 프로세서 별로 하나의 최종해가 결정되면 미리 정해진 순서에 따라 토너먼트(tournament) 방식으로 두 프로세서 씩 결과를 서로 비교하여 상대적으로 우수한 해를 선택함으로써 결국 전체 프로세서군에서 최적의 해를 찾을 수 있게 된다. 이는 비교 과정에 host 컴퓨터가 전혀 개입하지 않도록 하고 비교 과정이 병렬적으로 수행될 수 있도록 하기 위한 것이다.

4. 병렬유전알고리즘을 이용한 기동정지계획 문제

본 논문에서는 병렬유전알고리즘으로 기동정지계획문제를 해결하기 위해서 그림 3과 같이 2가지 방법으로 스트림을 구성하였다.

그림 3(a)에서는 제약조건을 고려하지 않고 매 시간대에서 발전기의 on/off상태로 스트림을 구성하였고, 그림 3(b)에서는 최소운전시간 및 최소정지시간을 포함하도록 스트림을 구성하였다. 그림 3(b)에서 첫번째 비트는 발전기의 기동/정지상태를 나타내고 나머지 3비트는 발전기의 상대지속시간을 나타낸다. 각 스트림을 평가하기 위해서 식(1)과 같이 발전비용과 기동비 및 위반지수(penalty factor)를 이용하여 적합도를 선정하였다.

$$\text{fitness} = FC_T + ST_T + \sum_{j=1}^N PF_j \quad (1)$$

여기서, $PF_j = \mu_j \cdot |V_j|$,
 PF_j : 제약조건 j의 위반값, FC_T : 총 연료비용
 μ_j : 제약조건 j의 위반 계수, ST_T : 총 기동비용
 V_j : 제약조건 j의 위반 여부, N: 제약조건 수

State 1 12 24
 Unit 1 111110100011111111111010011111

Unit M 111110100111111111111010011111

(a) 제약조건을 고려하지 않은 경우

Unit 1 00010111001011111111111110

Unit M 10010111001011001101011111

String 1 : Unit 1 Unit 2 . . . Unit N

String M : Unit 1 Unit 2 . . . Unit N

(b) 제약조건을 고려한 경우

여기서, N : unit의 개수, M : 스트링의 수

그림 3 스트링의 구성 방법

5. 사례 연구

제안한 방법의 유용성을 검토하기 위해서 참고 문헌[6]에 있는 예제 데이터를 이용하였다. 표 1에는 참고문헌[6]의 각 발전기에 대한 최대, 최소 전력, 최소운전시간, 최소정지시간, 기동비 및 연료비용함수의 계수를 나타내었고, 표 2에는 시간별 부하수요를 나타내었다. 병렬유전알고리즘의 실험 계수를 표 3에 나타내었다.

병렬유전알고리즘으로 기동정지계획 문제를 해결하기 위해서 제안한 두가지 부호화방법을 이용하여 병렬 트랜스퓨터 상에서 실험하였다.

경우 1) 제약조건을 고려하지 않은 경우

제약조건을 고려하지 않은 스트링을 이용하여 매 세대마다 구한 총비용은 그림 4와 같다. 최종적으로 구해진 매 시간 부하 수요에 대한 각 발전기의 발전량 및 총비용을 표 4에 나타내었다.

표 1 기동정지문제에 사용된 실험 데이터

	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5
P_{max} (MW)	455	130	130	80	55
P_{min} (MW)	150	20	20	20	55
a (\$/h)	1000	700	680	370	660
b (\$/MWh)	16.19	16.60	16.50	22.26	25.92
c (\$/MWh)	0.00048	0.002	0.00211	0.00712	0.00413
최소운전시간(h)	8	5	5	3	1
최소정지시간(h)	8	5	5	3	1
기동비(h)	9000	1100	1120	340	60
초기상태 (h)	8	-5	-5	-3	-1

여기서, a, b, c : 연료비용함수의 계수, 운전 예비력 : 10 %

표 2 시간별 부하 수요

시간[h]	1	2	3	4	5	6	7	8	9	10	11	12
부하수요 [MW]	400	450	480	500	530	550	580	600	620	650	680	700
시간[h]	13	14	15	16	17	18	19	20	21	22	23	24
부하수요 [MW]	650	620	600	550	500	550	600	650	600	550	500	450

표 3 병렬유전알고리즘의 특성 데이터

계수	방법	GA
Transputer의 수		16
해집단 개수		960
스트링의 길이		120 비트
교차 확률		0.6
돌연변이 확률		0.05
세대수		300

표 4 최종적으로 구해진 매 시간별 부하 수요에 대한 각 발전기의 발전량

발전량 시간별부하	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	총발전량 (MW)
1	400	400.0				400.00
2	450	430.0	20.0			450.00
3	480	455.0	25.0			480
4	500	455.0	45.05			500.05
5	530	455.0	75.05			530.05
6	550	455.0	75.05	20.00		550.05
7	580	455.0	105.08	20.00		580.08
8	600	455.0	125.07	20.00		600.07
9	620	455.0	62.28	82.72	20.00	620.00
10	650	455.0	77.73	97.37	20.00	650.1
11	680	455.0	93.08	111.93	20.00	680.01
12	700	455.0	103.35	121.66	20.00	700.01
13	650	455.0	87.99	107.1		650.09
14	620	455.0	72.54	92.45		619.99
15	600	455.0	62.28	82.73		600.01
16	550	455.0	36.58	58.37		549.95
17	500	450.8	20	29.11		499.91
18	550	455.0	36.59	58.37		549.96
19	600	455.0	62.28	82.73		600.01
20	650	455.0	87.99	107.10		650.09
21	600	455.0	62.28	82.73		600.01
22	550	455.0	36.59	58.37		549.96
23	500	455.0		45.08		500.08
24	450	426.4		23.54		450.04

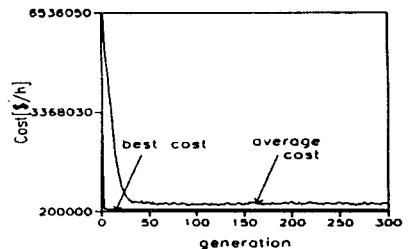


그림 4 매 세대에서 총비용

경우 2) 제약조건을 고려한 경우

제약조건을 고려한 스트림을 이용하여 매 세대마다 구한 총비용은 그림 5와 같다. 최종적으로 구해진 매 시간 부하 수요에 대한 각 발전기의 발전량 및 총비용을 표 5에 나타내었다.

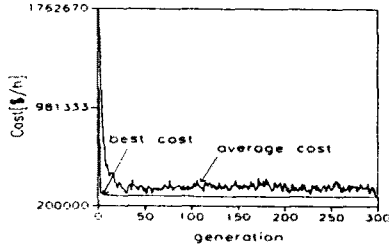


그림 5 매 세대에서의 총비용

표 5 최종적으로 구해진 매 시간별 부하 수요에 대한 각 unit의 발전량

발전량 시간별부하	Unit 1	Unit 2	Unit 3	Unit 4	Unit 5	총발전량 (MW)
1	400	399.99				399.99
2	450	430.0	20.0			450.00
3	480	455.0	25.04			480.04
4	500	455.0	45.05			500.05
5	530	455.0	75.05			530.05
6	550	455.0	75.05	20.00		550.05
7	580	455.0	105.08	20.00		580.08
8	600	455.0	125.07	20.00		600.07
9	620	455.0	62.28	82.72	20.00	620.00
10	650	455.0	77.73	97.37	20.00	650.1
11	680	455.0	93.08	111.93	20.00	680.01
12	700	455.0	103.35	121.66	20.00	700.01
13	650	455.0	87.99	107.1		650.09
14	620	455.0	72.54	92.45		619.99
15	600	455.0	62.28	82.73		600.01
16	550	455.0	36.58	58.37		549.95
17	500	450.8	20	29.11		499.91
18	550	455.0	36.59	58.37		549.96
19	600	455.0	62.28	82.73		600.01
20	650	455.0	87.99	107.10		650.09
21	600	455.0	62.28	82.73		600.01
22	550	455.0	36.59	58.37		549.96
23	500	455.0		45.08		500.8
24	450	426.4		23.54		450.04

이상의 결과를 요약하면 표 6과 같다. 표 6에서 보는 바와 같이 실행시간면에서 경우 2의 부호화방법을 적용한 병렬유전알고리즘에 비해 Sequential GA가 약 10배 정도 걸린다.

수렴한 세대수 관점에서 제약조건을 고려하지 않은 스트림 구성방법이 제약조건을 고려한 경우보다 더 좋은 특성을 나타내었다.

표 6 병렬유전알고리즘 실행 결과

방법	사항	실행시간[sec]	수렴한 세대수	총비용[\$/h]
경우 1		381	8	278874
경우 2		454	100	278874

6. 결론

본 논문에서는 기동정지계획문제를 해결하기 위해 병렬유전알고리즘을 적용하는 방법을 제안하였다. 이 때, 기동정지계획문제의 여러 제약조건을 만족하도록 병렬유전알고리즘의 스트림을 구성하는 방법과 각 스트림을 평가하기 위한 적합도 함수를 설정하는 방법을 제안하였다.

제안한 방법을 이용하여 병렬트랜스퓨터상에서 실험해본 결과, 기존의 Sequential GA보다 실행시간면에서 많이 향상되었고, 또한 제약조건을 고려하지 않은 스트림의 구성방법이 제약조건을 고려한 부호화 방법보다 실행시간 및 수렴한 세대수 면에서 더 나은 특성을 나타내었다.

7. 참고 문헌

- [1] Pang, C.K. and Chen, H.C. "Optimal short-term thermal unit commitment, IEEE Trans., 1976, PAS-95, pp. 1336-1346
- [2] Zhuang, F. and Galiana, F.D. "Towards a more rigorous and practical unit commitment by Lagrangian relaxation", IEEE Trans., 1988, PWRS-3, no. 2, pp. 763-773
- [3] Zhuang, F. and Galiana, F.D. "Unit commitment by simulated annealing", IEEE Trans., 1990, PWRS-5, no. 1, pp. 311-317
- [4] David E. Goldberg, "Genetic Algorithms in Search, Optimization, and Machine Learning", ADDISON-WESLEY PUBLISHING COMPANY, INC., 1989
- [5] Hiroumi Saitoh, Kazushige Inoue, Junichi Toyoda, "Genetic algorithm approach to unit commitment problem", ISAP, 1994, vol 2, pp. 583-589
- [6] S.A. Kazariis, A.G. Bakirtzis, V. Petridis, "A genetic algorithm solution to the unit commitment problem", IEEE Trans., 1996, PWRS-11, no. 1, pp. 83-92
- [7] M.E.C. Hull, D. Crookes, P.J. Sweeney, Parallel Processing : The Transputer and its Applications, Addison-Wesley, England, 1994.
- [8] R. Schonkwiler, "Parallel Genetic Algorithms", Proceedings of International Conference on Genetic Algorithms, pp. 199-205, 1993.
- [9] 정상화, 류광렬, 오수철, 박태완, 박수희, "초고속 경보검색용 병렬처리 시스템", 병렬처리 시스템 연구회지 제 7 권 2호, 1996