

<http://dx.doi.org/10.7236/IIBC.2015.15.2.255>

IIBC 2015-2-34

# 균형-교환방법을 적용한 경제급전문제 최적화 알고리즘

## Optimization Algorithm for Economic Load Dispatch Problem Using Balance and Swap Method

이상운\*

Sang-Un Lee\*

**요약** 경제급전 최적화 문제를 해결하는 결정론적인 알고리즘에 존재하지 않아 지금까지는 비결정론적인 휴리스틱 알고리즘들이 제안되고 있다. 본 논문은 균형과 교환 방법을 도입하여 경제급전의 최적화 문제를 풀 수 있는 알고리즘을 제안하였다. 제안된 알고리즘은 초기치에 대해 성인걸음수와 아기걸음 수별로 발전량을 감소시켜  $\Sigma P_i = P_d$ 로 균형을 맞추고, 이 때 최소 발전비용을 가진 방법을 선택한다. 다음으로 선택된 방법에 대해 성인걸음-아기걸음 교환과 거인걸음 교환 방법으로 최적화한 값을 구하여 최소값 방법을 선택한다. 마지막으로 선택된 방법에 대해  $P_i \pm \beta$ , ( $\beta = 0.1, 0.01, 0.001, 0.0001$ )의 교환을 수행하였다. 경제급전 문제의 시험사례로 빈번히 활용되고 있는 3개 데이터에 대해 제안된 알고리즘을 적용한 결과 2개 데이터에서는 성능을 향상시켰으며, 1개 데이터는 기존의 최적해와 동일한 결과를 얻었다. 제안된 알고리즘은 항상 동일한 결과를 얻을 수 있고, 모든 데이터에 적합하므로 경제급전 최적화 알고리즘으로 실제 적용이 가능하다.

**Abstract** In the absence of a deterministic algorithm for economic load dispatch optimization problem (ELDOP), existing algorithms proposed as solutions are inevitably non-deterministic heuristic algorithms. This paper, therefore, proposes a balance-and-swap algorithm to solve an ELDOP. Firstly, it balances the initial value to  $\Sigma P_i = P_d$  by subsequently reducing power generation for each adult-step and baby-step and selects the minimum cost-generating method. Subsequently, it selects afresh the minimum cost-generating method after an optimization of the previously selected value with adult-step baby-step swap and giant-step swap methods. Finally, we perform the  $P_i \pm \beta$ , ( $\beta = 0.1, 0.01, 0.001, 0.0001$ ) swap. When applied to the 3 most prevalently used economic load dispatch problem data, the proposed algorithm has obtained improved results for two and a result identical to the existing one for the rest. This algorithm thus could be applied to ELDOP for it has proven to consistently yield identical results and to be applicable to all types of data.

**Key Words** : Economic Load Dispatch, Optimization, Balance, Swap, Heuristic algorithm, Deterministic algorithm

### 1. 서론

발전비용함수  $F(P_i)$ 가 각기 다른  $n$ 개의 발전기로 된

하는 전력 수요량  $P_d$ 를 충족시키면서 총 발전비용  $\Sigma F(P_i)$ 을 최소화시키도록 각 발전기별로 발전량  $P_i$  ( $i = 1, 2, \dots, n$ )를 어떻게 할당하는 것이 최적인지의 해를 구

\*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과  
접수일자 : 2014년 10월 28일, 수정완료 : 2015년 2월 25일  
게재확정일자 : 2015년 4월 10일

Received: 28 October, 2014 / Revised: 25 February, 2015 /  
Accepted: 10 April, 2015

\*Corresponding Author: [sulee@gwnu.ac.kr](mailto:sulee@gwnu.ac.kr)  
Dept. of Multimedia Eng., Gangneung-Wonju National University,  
Korea

하는 문제를 경제급전 (economic load dispatch, ELD)이라 한다.<sup>[1]</sup>

전력분야에서 발전량이 수요량 (부하량 또는 요구량)보다 적을 경우 주파수 난조와 정전 발생으로 전력계통에 큰 문제를 발생시키고, 산업 전반에 걸쳐 경제적 손실도 막대해진다. 전력은 저장이 불가능한 특성을 갖고 있어 시간대별로 수요량과 일치하도록 발전량을 맞추어야 한다.

현재의 설비용량으로 추가 소요 비용 없이 가장 경제적인 이득을 얻을 수 있는 방법이 경제급전이다. 왜냐하면, 동일한 총 발전량을 유지함에도 불구하고 발전비용 함수가 각기 다른 각 발전기별로 발전량을 어떻게 배정하느냐에 총 발전비용은 엄청난 차이가 발생하기 때문이다. 그러나 경제급전 문제는 발전비용함수의 복잡성으로 인해 최적화 문제들 중에서도 가장 난제에 속한다고 할 수 있으며, 전력분야에서 반드시 해결해야 경쟁력을 갖출 수 있는 필수불가결한 난제 중 하나이다.

경제급전 최적화와 관련하여 초기에는 2차 다항식의 발전비용함수만을 고려하여 대부분 수학적 최적화 기법인 라그랑주 완화법 (Lagrangian relaxation)이나 경사법 (Gradient method)을 적용하였으나 밸브효과를 적용한 발전비용함수에 대해서는 미분 불가능으로 수학적 기법은 더 이상 적용될 수 없게 되었으며, 최적화 난제로 인해 휴리스틱 방법인 인공지능 기법으로 연구를 수행하고 있다. 인공지능 기법으로는 유전자 알고리즘 (Genetic algorithm, GA), 신경망 (neural network, NN), 퍼지 (Fuzzy), 군집 알고리즘 (Particle Swarm Optimization, PSO) 등 비결정론적인 알고리즘 (non-deterministic algorithm)들이다.<sup>[1-19]</sup>

휴리스틱 기법은 초기치를 랜덤하게 설정하고 수많은 시행 결과 최소값을 최적해로 결정한다. 따라서 최소값에 대한 수많은 모수에 대한 초기치를 기억하지 않는 한 동일한 결과를 다시 얻는다는 보장을 하지 못한다. 또한, 다양한 휴리스틱 알고리즘을 연구하고 있음에도 불구하고, 여러 경우에 대한 실험 데이터에 모두 최적 해를 얻은 알고리즘은 아직 제안되지 않고 있다.

본 논문에서는 휴리스틱 알고리즘들에 비해 경제급전 문제의 해를 개선할 수 있으면서도 쉽고 빠르게 해를 구하며 항상 동일한 결과를 얻을 수 있는 결정론적 알고리즘 (deterministic algorithm)을 제안한다. 2장에서는 경제급전과 관련된 연구의 문제점을 고찰해 본다. 3장에서

는 균형과 교환 (balance and swap) 기법을 적용한 결정론적 알고리즘을 제안한다. 4장에서는 제안된 알고리즘을 사례들에 적용하여 휴리스틱 방법들과 성능을 비교 검증한다.

## II. 관련연구와 문제점

각기 다른 발전비용 함수  $F(P_i)$ 를 가진  $n$ 개의 발전기  $G_i$ , ( $i=1,2,\dots,n$ )가 있는 경우, 주어진 시간대의 총 발전량  $\Sigma P_i$ 는 식 (1)과 같이 수급 균형을 맞추기 위해 총 수요량 (또는 요구량)  $P_d$ 와 동일해야 한다. 여기서  $P_i$ 는 발전기  $i$ 의 발전량 (KW)이다.<sup>[1]</sup>

$$\sum_{i=1}^n P_i^{\min} < P_d < \sum_{i=1}^n P_i^{\max} \rightarrow \sum_{i=1}^n P_i = P_d \quad (1)$$

경제급전 문제의 목적함수는 경제급전 총 발전비용을 최소화시키는 것으로 식 (2)와 같다.<sup>[1]</sup>

$$\text{minimize } F(P) = \sum_{i=1}^n F(P_i) \quad (2)$$

여기서  $F(P_i)$ 는 발전기  $i$ 의 시간당 발전비용 (\$/ MWh)으로 식 (3)으로 표현되며, 그림 1에 제시되어 있다.<sup>[1-19]</sup>

$$F(P_i) = a_i P_i^2 + b_i P_i + c_i + |e_i \times \sin(f_i \times (P_i^{\min} - P_i))| \quad (3)$$

발전비용함수  $F(P_i)$ 는 발전량  $P_i$ 에 대해 기본적으로  $a_i P_i^2 + b_i P_i + c_i$ 의 2차함수로 변화되며, 출력을 높이기 위해 밸브를 열 때 진동 발생에 따른 밸브효과 (valve-effect)로 인해 발생하는 추가비용인  $|e_i \times \sin(f_i \times (P_i^{\min} - P_i))|$ 의 사인함수 절대값이 추가된다. 따라서 실제의 발전비용 함수는 식 (3)으로 2차함수 이상의 복잡한 비선형 형태를 갖고 있다. 이러한 복잡한 비선형 발전비용 함수를 미분하여 기울기를 구할 수 없어 수학적 방법은 차선책으로  $F'(P_i) = a_i' P_i^2 + b_i' P_i + c_i'$ 의 근사 평활함수 (smooth function)에 대한 기울기  $2a_i' P_i + b_i'$ 를 이용해 해를 구한다.

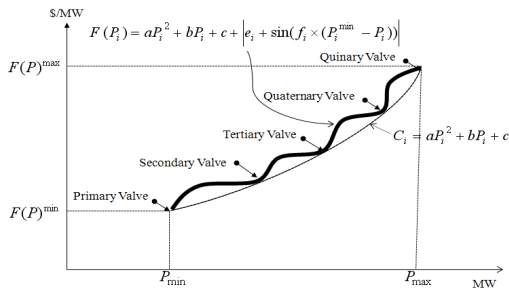


그림 1. 발전비용 함수  
 Fig. 1. Power cost function

그림 1에서 각 발전기의 발전량 조건은 식 (4)와 같다.

$$P_i^{\min} \leq P_i \leq P_i^{\max} \quad (4)$$

여기서  $P_i^{\min}$  은  $i$  번째 발전기의 최소 발전량을,  $P_i^{\max}$  는 발전기가 생산할 수 있는 최대 발전량이다. 이는 비록 고비용 발전기가 존재하더라도 그림 2[19]와 같이 1일 시간대별 전력 수요량 변동, 예측량을 초과하는 수요량 증가나 고장 발생시 즉시 이를 대체 충족시키기 위해 선정된 모든 발전기는 가동시킨다고 가정하고 있다.

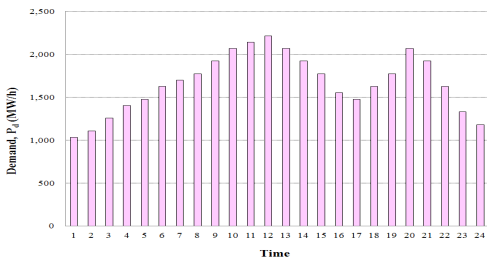


그림 2. 1일 수요량 변동 추이<sup>[19]</sup>  
 Fig. 2. Deviation of 1 day demand trend

또한, 최소 발전량을  $P_i^{\min} = 0$  (발전 또는 기동정지)로 설정하지 않고  $P_i^{\min} > 0$ 으로 하는 이유는 균일한 품질의 전력을 공급하기 위해, 일단 발전기가 가동된 이후 안전하게 연속적으로 운전이 가능한 최저수준의 출력에 도달한 시점 ( $P_i^{\min}$ )에서 발전기를 전력계통에 연결시켜 실제로 전력을 공급한다. 석탄 발전기의  $P_i^{\min}$  은 정격 출력의 30 ~ 40%, 중유 발전기는 25 ~ 35% 정도로 알려져 있다.

표 1은 경제급전 문제에서 가장 많이 활용되고 있는 실험 데이터로 Coelho와 Lee<sup>[2]</sup>가 제시한 13-발전기와 40-발전기의 사례이다. Coelho와 Mariani<sup>[3]</sup>는 13-발전기

데이터의 최소 발전총량  $\Sigma P_i^{\min} = 550 MW$ , 최대 발전 총량  $\Sigma P_i^{\max} = 2,960 MW$ , 요구량은  $P_d = 1,800 MW$ 와  $2,520 MW$ 의 2가지 경우이다. 40-발전기는  $\Sigma P_i^{\min} = 4,817 MW$ ,  $\Sigma P_i^{\max} = 12,722 MW$ ,  $P_d = 10,500 MW$ 이다.

표 1. 실험 데이터  
 Table 1. Case study data

Fuel Cost Function of 13 Generators							
$G_i$	$P_i^{\min}$	$P_i^{\max}$	$a$	$b$	$c$	$e$	$f$
1	0	680	0.00028	8.10	300	300	0.085
2	0	360	0.00056	8.10	309	200	0.042
3	0	360	0.00056	8.10	307	150	0.042
4	60	180	0.00324	7.74	240	150	0.063
5	60	180	0.00324	7.74	240	150	0.063
6	60	180	0.00324	7.74	240	150	0.063
7	60	180	0.00324	7.74	240	150	0.063
8	60	180	0.00324	7.74	240	150	0.063
9	60	180	0.00324	7.74	240	150	0.063
10	40	120	0.00284	8.60	126	100	0.084
11	40	120	0.00284	8.60	126	100	0.084
12	55	120	0.00284	8.60	126	100	0.084
13	55	120	0.00284	8.60	126	100	0.084
Sum:	550	2,960	$P_d = 1,800 MW$ and $P_d = 2,520 MW$				

Fuel Cost Function of 40 Generators							
$G_i$	$P_i^{\min}$	$P_i^{\max}$	$a$	$b$	$c$	$e$	$f$
1	36	114	0.00690	6.73	94.71	100	0.084
2	36	114	0.00690	6.73	94.71	100	0.084
3	60	120	0.00208	7.07	309.54	100	0.084
4	80	190	0.00942	8.18	369.03	150	0.063
5	47	97	0.01140	5.35	148.89	120	0.077
6	68	140	0.01142	8.05	222.33	100	0.084
7	110	300	0.00357	8.03	287.71	200	0.042
8	135	300	0.00492	6.99	391.98	200	0.042
9	135	300	0.00573	6.60	455.76	200	0.042
10	130	300	0.00605	12.90	722.82	200	0.042
11	94	375	0.00515	12.90	635.20	200	0.042
12	94	375	0.00569	12.80	654.69	200	0.042
13	125	500	0.00421	12.50	913.40	300	0.035
14	125	500	0.00752	8.84	1760.40	300	0.035
15	125	500	0.00708	9.15	1728.30	300	0.035
16	125	500	0.00708	9.15	1728.30	300	0.035
17	220	500	0.00313	7.97	647.85	300	0.035
18	220	500	0.00313	7.95	649.69	300	0.035
19	242	550	0.00313	7.97	647.83	300	0.035
20	242	550	0.00313	7.97	647.81	300	0.035
21	254	550	0.00298	6.63	785.96	300	0.035
22	254	550	0.00298	6.63	785.96	300	0.035
23	254	550	0.00284	6.66	794.53	300	0.035
24	254	550	0.00284	6.66	794.53	300	0.035
25	254	550	0.00277	7.10	801.32	300	0.035
26	254	550	0.00277	7.10	801.32	300	0.035
27	10	150	0.52124	3.33	1055.10	120	0.077
28	10	150	0.52124	3.33	1055.10	120	0.077
29	10	150	0.52124	3.33	1055.10	120	0.077
30	47	97	0.01140	5.35	148.89	120	0.077
31	60	190	0.00160	6.43	222.92	150	0.063
32	60	190	0.00160	6.43	222.92	150	0.063
33	60	190	0.00160	6.43	222.92	150	0.063
34	90	200	0.00010	8.95	107.87	200	0.042
35	90	200	0.00010	8.62	116.58	200	0.042
36	90	200	0.00010	8.62	116.58	200	0.042
37	25	110	0.01610	5.88	307.45	80	0.098
38	25	110	0.01610	5.88	307.45	80	0.098
39	25	110	0.01610	5.88	307.45	80	0.098
40	242	550	0.00313	7.97	647.83	300	0.035
Sum:	4,817	12,722	$P_d = 10,500 MW$				

표 2에서는 표 1의 실험 데이터에 대한 기존의 연구결과를 요약하여 제시하였다. 표에서 ( )는 실제 각 발전기별 발전량을 제시한 경우에 한해 결과 비교를 위해 MS-Office Excel 2007로 다시 계산된 값을 표기하였으며, 결과만을 제시한 경우는 (?)로 표기하였다.

표 2에 제시된 휴리스틱 기법의 문제점은 다음과 같다.

표 2. 시험사례의 최적화 수행 결과

Table 2. Results for optimization of test cases

Case with 13 Generators and Load Demand of 1,800 MW				
Method	Min Cost	Mean Cost	Max Cost	St Dev
Improved Genetic Algorithm (IGA) <sup>[4]</sup>	18,063.580 (?)	18,066.40	18,293.47	45.79
Particle Swarm Optimization (PSO) <sup>[6]</sup>	18,030.720 (?)	-	-	-
Evolutionary Programming (EP) <sup>[6]</sup>	17,994.070 (?)	-	-	-
Hybrid Evolutionary Programming with SQP (HEP-SQP) <sup>[5]</sup>	17,991.030 (?)	-	-	-
Modified Particle Swarm Optimization (MPSO) <sup>[7]</sup>	17,973.340 (?)	-	-	-
Artificial Immune System (IS) <sup>[1]</sup>	17,972.90 (?)	17,976.14	17,989.84	5.94
Hybrid Particle Swarm with SQP (HPS-SQP) <sup>[5]</sup>	17,969.930 (?)	-	-	-
Pattern Search (PS) <sup>[8]</sup>	17,969.170 (?)	-	-	-
Cultural IS (CIS) <sup>[1]</sup>	17,964.68 (?)	17,974.28	17,989.23	4.81
Fuzzy CIS (FCIS) <sup>[1]</sup>	17,964.37 (?)	17,974.50	17,989.81	3.19
Harmony Search (HS) <sup>[9]</sup>	17,960.3661 (17,960.3708)	-	-	-
Chaotic Differential Evolution and Quadratic Programming (DEC-SQP) <sup>[3]</sup>	<b>17,938.9521</b> (?)	17,943.1339	17,944.8105	1.9735

Case with 13 Generators and Load Demand of 2,520 MW				
Method	Min Cost	Mean Cost	Max Cost	St Dev
Cultural Immune System (CIS)[1]	24,171.13 (?)	24,190.34	24,289.44	31.68
Fuzzy CIS (FCIS)[1]	24,171.09 (?)	24,187.72	24,318.58	33.12
Artificial Immune System (IS)[1]	24,170.57 (?)	24,193.72	24,289.03	35.52
Distributed Tabu Search Algorithm (DTSA)[10]	24,169.950 (?)	-	-	-
Self-Adaptive Differential Evolution (SDE)[11]	<b>24,164.05</b> (?)	24,168.28	24,200.05	-

Case with 40 Generators and Load Demand of 10,500 MW				
Method	Min Cost	Mean Cost	Max Cost	St Dev
Genetic Algorithm (GA) <sup>[12]</sup>	119,732.25 (123,966.653)	-	-	-
Embedded Evolutionary Programming-PSO (CEP-PSO) <sup>[13]</sup>	123,670.000 (?)	124,145.60	124,900.00	-
Particle Swarm Optimization (PSO) <sup>[6]</sup>	122,930.450 (?)	-	-	-
Evolutionary Programming (EP) <sup>[6]</sup>	122,624.350 (?)	-	-	-
Hybrid Evolutionary Programming with SQP (HEP-SQP) <sup>[5]</sup>	122,379.630 (?)	-	-	-
Modified Particle Swarm Optimization (MPSO) <sup>[7]</sup>	122,252.265 (?)	-	-	-
Hybrid Particle Swarm with SQP (HPS-SQP) <sup>[5]</sup>	122,094.670 (?)	-	-	-
Chaotic Differential Evolution and Quadratic Programming (DEC-SQP) <sup>[3]</sup>	121,741.9793 (121,532.104)	122,295.12	122,839.29	386.18
New Particle Swarm Optimization Solution to Nonconvex (NPSO-LRS) <sup>[15]</sup>	121,664.430 (?)	122,209.31	122,981.59	-
Improved PSO (IPSO) <sup>[16]</sup>	121,495.7041 (121,504.702)	121,699.301	122,168.116	155.619
Artificial Immune System (IS) <sup>[1]</sup>	121,489.110 (?)	121,737.73	122,163.56	106.88
SA-CLONAL <sup>[17]</sup>	121,486.12 (?)	121,507.03	121,591.70	-
Fuzzy CIS (FCIS) <sup>[1]</sup>	121,446.710 (?)	121,662.67	121,863.39	138.04
Genetic Algorithm (GA) <sup>[18]</sup>	121,432.177 (121,532.104)	-	-	-
Cultural Immune System (CIS) <sup>[1]</sup>	121,423.680 (?)	121,717.97	122,147.93	144.81

Case with 13 & 40 Generators			
Method	13 Gen $P_d = 1,800$	13 Gen $P_d = 2,520$	40 Gen $P_d = 10,500$
<b>GA</b>			
Genetic Algorithm (GA) <sup>[12]</sup>	-	-	(123,966.653)
Genetic Algorithm (GA) <sup>[18]</sup>	-	-	(121,532.104)
Improved Genetic Algorithm (IGA) <sup>[4]</sup>	18,063.580 (?)	-	-
<b>PSO</b>			
Particle Swarm Optimization (PSO) <sup>[6]</sup>	18,030.720 (?)	-	122,930.450 (?)
Modified Particle Swarm Optimization (MPSO) <sup>[7]</sup>	17,973.340 (?)	-	-
Modified Particle Swarm Optimization (MPSO) <sup>[14]</sup>	-	-	122,252.265 (?)
Hybrid Particle Swarm with SQP (HPS-SQP) <sup>[5]</sup>	17,969.930 (?)	-	122,094.670 (?)
New Particle Swarm Optimization Solution to Nonconvex (NPSO-LRS) <sup>[15]</sup>	-	-	121,664.430 (?)
Improved PSO (IPSO) <sup>[16]</sup>	-	-	<b>(121,504.702)</b>
<b>IS</b>			
Artificial Immune System (AIS) <sup>[1]</sup>	17,972.90 (?)	24,170.57 (?)	121,489.11 (?)
Cultural Immune System (CIS) <sup>[1]</sup>	17,964.68 (?)	24,171.13 (?)	121,423.68 (?)
Fuzzy Cultural Immune System (FCIS) <sup>[1]</sup>	17,964.37 (?)	24,171.09 (?)	121,446.71 (?)
<b>EP</b>			
Evolutionary Programming (EP) <sup>[6]</sup>	17,994.070 (?)	-	122,624.350 (?)
Hybrid Evolutionary Programming with SQP (HEP-SQP) <sup>[5]</sup>	17,991.030 (?)	-	122,379.630 (?)
Chaotic Differential Evolution and Quadratic Programming (DEC-SQP) <sup>[3]</sup>	17,938.9521 (?)	-	(121,532.104)
Embedded Evolutionary Programming-PSO (CEP-PSO) <sup>[13]</sup>	-	-	123,670.000 (?)
Self-Adaptive Differential Evolution (SDE) <sup>[11]</sup>	-	<b>24,164.05 (?)</b>	-
<b>Extra</b>			
Pattern Search (PS) <sup>[8]</sup>	17,969.170 (?)	-	-
Harmony Search (HS) <sup>[9]</sup>	<b>(17,960.371)</b>	-	-
Distributed Tabu Search Algorithm (DTSA) <sup>[10]</sup>	-	24,169.950 (?)	-
SA-CLONAL <sup>[17]</sup>	-	-	121,486.12 (?)

(1) 수많은 가중치 (모수)에 대해 초기치를 랜덤하게 설정함으로 인해 각 시행의 결과가 모두 다르다.

이로 인해 표 2와 같이 최소 (최적)치, 평균치, 최대치와 표준편차를 결과로 제시한다. 따라서 평균적으로는 평균치의 결과를 얻을 뿐이다.

(2) 표 2에서의 최소 (최적)치는 수많은 시행횟수 중에서 임의의 랜덤한 초기치에서 요행히 하나의 최소치를 얻은 경우이다. 이로 인해, 이 결과를 다시 얻을 확률은 거의 없다. 만약, 현실적인 문제인 수요량이 변동되는 1일 (24시간)의 동적 경제급전 (dynamic ELD, DELD)문제에 적용하여 모든 시간대에서 최적치를 얻는다고 보장할 수도 없다.

(3) 표 2에서 비록 실험 데이터 (benchmark data)는 3개에 불과하지만 3개 데이터 모두에 적용된 알고리즘은 IS 뿐이며, 이 알고리즘조차도 모든 데이터에서 최적치를 얻었다고 볼 수 없다. 즉, 다양한 경우에 모두 적합한 하나의 일반화된 알고리즘이 존재하지 않고 있다.

### III. 균형-교환 알고리즘

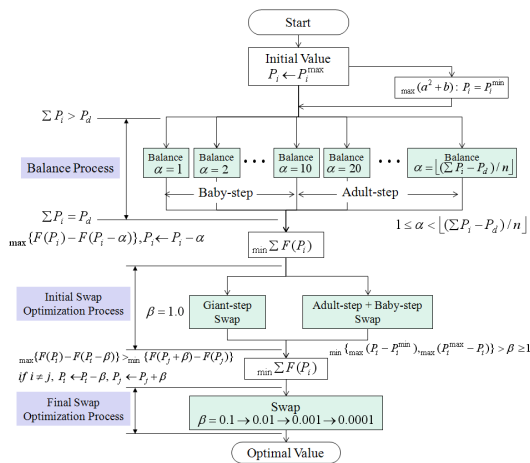


그림 3. 균형-교환 알고리즘  
 Fig. 3. Balance and swap algorithm (BSA)

본 장에서는 경제급전 문제의 최적 해를 구하기 위해 그림 3의 균형-교환 (balance & swap) 기법을 적용한 알고리즘을 제안한다. 제안된 알고리즘은 2단계 과정을 거친다. 초기치로  $\Sigma P_i > P_d$ 가 되도록 설정하고,  $\Sigma P_i = P_d$ 인 균형 과정 (balance process)을 수행한다. 균형과정에서의  $\min \Sigma F(P_i)$ 를 선택하여 최소 발전비용을 얻기 위한

최적화는 교환 과정 (swap process)을 통해 수행한다. 균형과 교환과정에서는 특정 범위에 대해 범위 전체를 보폭으로 하는 거인걸음 (giant-step), 보폭을 “10”으로 하는 성인걸음 (adult-step)과 보폭이 “1”인 아기걸음 (baby-step) 개념을 도입하였다.

초기치 (initial value)를 설정하는 방법은 다음의 2가지 경우를 고려한다.

- (1)  $P_i \leftarrow P_i^{\max}, \Sigma P_i > P_d$ .
- (2)  $P_i^{\min} = 0$ 를 포함하고 있는 경우에 한해  $P_i \leftarrow P_i^{\max}, \Sigma P_i > P_d$ 일 때까지  $\max(a^2 + b)$ 인 발전기  $i$ 에 대해  $P_i \leftarrow P_i^{\min}$ 으로 설정한다.

$\Sigma P_i > P_d$ 인 상태에서  $\Sigma P_i = P_d$ 로의 균형과정은 다음과 같이 수행한다.

- (1)  $1 \leq \alpha \leq \lfloor (\Sigma P_i - P_d) / n \rfloor$ 에 대해  $1 \leq \alpha \leq 10$ 에 대해서는 각 아기걸음 수에 대한 10개의  $\alpha$ 값을,  $10 < \alpha \leq \lfloor (\Sigma P_i - P_d) / n \rfloor$ 인 10의 배수 최대치에 대해서는 각 성인걸음 수  $kj$ 로  $\alpha = kj$  값을 결정한다. 여기서,  $k$ 는 걸음수,  $j$ 는 성인걸음 보폭 10을 의미한다.
- (2) 각  $\alpha$  걸음수에 대해  $\max\{F(P_i) - F(P_i - \alpha)\}$ 인 발전기  $i$ 의 발전량을  $P_i \leftarrow P_i - \alpha$ 로 감소시킨다. 이는  $\Sigma P_i > P_d$ 와  $P_i^{\min} \leq P_i \leq P_i^{\max}$ 를 만족하는 경우에 한정된다.
- (3) 만약, (2) 수행 후  $0 < \Sigma P_i - P_d < \alpha$ 이면  $\max\{F(P_i) - F(P_i - 1)\}$ 에 대해  $\Sigma P_i = P_d$ 일 때까지  $P_i \leftarrow P_i - 1$ 로 조절한다.
- (4) 균형을 맞추는 과정에서 수행된  $\alpha$  후보 방법들 중에서  $\min \Sigma F(P_i)$ 을 선택한다.

초기 교환 최적화 과정 (initial swap optimization process)은 다음과 같이 2가지 방법을 수행한다.

- (1)  $\min\{\max(P_i - P_i^{\min}), \max(P_i^{\max} - P_i)\} > \beta \geq 1$  범위에서  $\beta = \min\{\max(P_i - P_i^{\min}), \max(P_i^{\max} - P_i)\} \geq$ 의 10의 배수 최대치 -  $kj$ 로 10까지는 내림차순으로 성인걸음 교환을, 나머지 9부터 1까지는 아기걸음 교환을 수행한다.
- (2)  $\beta = 1$ 인 거인걸음 교환을 수행한다.

교환 방법은  $\max\{F(P_i) - F(P_i - \beta)\} > \min\{F(P_i + \beta) -$

$F(P_j)$ ,  $i \neq j$ 에 대해  $P_i \leftarrow P_i - \beta$ ,  $P_j \leftarrow P_j + \beta$ 로 발전량을 교환한다. 만약,  $i = j$ 이면 다음  $\min\{F(P_j + \beta) - F(P_j)\}$ 을 선택하여 교환이 가능한지 검증한다.

최종 교환 최적화 과정 (final swap optimization process)은 초기 교환 최적화 과정 방법들 중에서  $\min \Sigma F(P_i)$ 를 선택하여  $\beta = 0.1 \rightarrow 0.01 \rightarrow 0.001 \rightarrow 0.0001$ 에 대해 교환 과정을 수행하여 발전비용을 미세하게 감소시킨다.

#### IV. 적용 결과 및 분석

표 1의 13-발전기와 40-발전기에 대해 제안된 알고리즘을 수행한 결과는 표 3에 제시되어 있다.

13-발전기 ( $P_d = 1,800MW$ )의 경우  $\max(a^2 + b)$ 의 발전기에 대해 초기치가  $P_i \leftarrow P_i^{\min}$ 으로 설정되고,  $\alpha = 1$ 의 균형을 수행한 결과 최소 발전비용을 나타내었으며, 다시 성인걸음 교환을 수행한 결과가 최소 발전비용을 얻었다.

13-발전기 ( $P_d = 2,520MW$ )의 경우  $P_i \leftarrow P_i^{\max}$  초기치에 대해  $\alpha = 10$ 의 균형이 최소 발전비용을 나타내었으며, 다시 성인걸음 교환이 최소 발전비용을 나타내었다.

40-발전기 ( $P_d = 10,500MW$ )의 경우  $P_i \leftarrow P_i^{\max}$ 의 발전기에 대해 초기치가  $P_i \leftarrow P_i^{\min}$ 으로 설정되고,  $\alpha = 1$ 의 균형을 수행한 결과 최소 발전비용을 나타내었으며, 다시 거인걸음 교환을 수행한 결과가 최소 발전비용을 얻었다.

실험에서,  $\alpha$ 와  $\beta$ 의 범위는 표 4와 같이 설정되었다.

일반화된 모델이 되기 위해서는 가능한 모든 상황을 고려한 다양한 데이터에 적합해야 한다. 이러한 조건을 만족시키기 위해 3개 데이터에 모두 적용된 IS 계열 알고리즘과 제안된 BSA 알고리즘 결과를 기존에 알려진 최적해와 비교하여 표 5에 제시하였다. 표 5에서 13-발전기 ( $P_d = 2,520MW$ )의 경우 기존에 알려진 최적해에 대해 각 발전기별 발전량을 제시한 자료가 없어 부득이 기존 결과들에서 최소치를 선택하였다. 나머지 2개 데이터는 MS-Office Excel 2007로 검증된 값들 중에서 최소치로 제시하였다.

표 3. 균형과 교환 알고리즘 최적화 결과

Table 3. Optimization results of BSA for the test cases

Gen'	$P_i^{\min}$	$P_i^{\max}$	$P_d = 1,800MW$		$P_d = 2,520MW$	
			$P_i$	$F(P_i)$	$P_i$	$F(P_i)$
1	0	680	628.3185	5,749.920	628.3185	5,749.920
2	0	360	0.0000	0.000	299.1993	2,782.646
3	0	360	0.0000	0.000	294.4840	2,770.397
4	60	180	159.7331	1,559.002	159.7331	1,559.002
5	60	180	159.7331	1,559.002	159.7331	1,559.002
6	60	180	159.7331	1,559.002	159.7331	1,559.002
7	60	180	159.7331	1,559.002	159.7331	1,559.002
8	60	180	159.7331	1,559.002	159.7331	1,559.002
9	60	180	159.7331	1,559.002	159.7331	1,559.002
10	40	120	63.2829	774.287	77.3999	808.653
11	40	120	40.0000	474.544	77.3999	808.653
12	55	120	55.0000	607.591	92.3999	944.886
13	55	120	55.0000	607.591	92.3999	944.886
Sum	550	2,960	1,800.000	17,567.943	2,520.000	24,164.051

Gen'	$P_i^{\min}$	$P_i^{\max}$	$P_i$	$F(P_i)$
1	36	114	110.7998	925.096
2	36	114	110.7998	925.096
3	60	120	97.3999	1,190.548
4	80	190	179.7331	2,143.550
5	47	97	87.7999	706.500
6	68	140	140.0000	1,596.464
7	110	300	259.5996	2,612.884
8	135	300	284.5996	2,779.837
9	135	300	284.5996	2,798.230
10	130	300	130.0000	2,502.065
11	94	375	94.0000	1,893.305
12	94	375	94.0000	1,908.167
13	125	500	304.5195	5,110.297
14	125	500	394.2793	6,414.860
15	125	500	394.2793	6,436.586
16	125	500	394.2793	6,436.586
17	220	500	489.2794	5,296.711
18	220	500	489.2794	5,288.766
19	242	550	511.2794	5,540.930
20	242	550	511.2794	5,540.910
21	254	550	523.2794	5,071.290
22	254	550	523.2794	5,071.290
23	254	550	523.2794	5,057.224
24	254	550	523.2794	5,057.224
25	254	550	523.2794	5,275.089
26	254	550	523.2794	5,275.089
27	10	150	10.0000	1,140.524
28	10	150	10.0000	1,140.524
29	10	150	10.0000	1,140.524
30	47	97	87.7999	706.500
31	60	190	190.0000	1,643.991
32	60	190	190.0000	1,643.991
33	60	190	190.0000	1,643.991
34	90	200	164.7998	1,585.544
35	90	200	164.7998	1,539.870
36	90	200	164.7998	1,539.870
37	25	110	105.9245	1,190.693
38	25	110	110.0000	1,220.166
39	25	110	89.1141	959.296
40	242	550	511.2794	5,540.930
Sum	4,817	12,722	10,500.0000	121,491.013
$P_d$	10,500			

표 4.  $\alpha$ 와  $\beta$ 값 설정

Table 4. Range of  $\alpha$  and  $\beta$  value

구분	$\alpha$	$\beta$
13 Gen ( $P_d = 1,800$ )	$1 \leq \alpha < 89.231$ $\alpha = 1, 2, \dots, 10, 20, \dots, 80$	$(a^2 + b), \alpha = 1$ $1 \leq \beta \leq 360$ $\beta = 360, 350, \dots, 10$ $\beta = 9, 8, \dots, 1$
13 Gen ( $P_d = 2,520$ )	$1 \leq \alpha < 33.846$ $\alpha = 1, 2, \dots, 10, 20, 30$	$\alpha = 10$ $1 \leq \beta \leq 80$ $\beta = 80, 70, \dots, 10$ $\beta = 9, 8, \dots, 1$
40 Gen ( $P_d = 10,500$ )	$1 \leq \alpha < 55.55$ $\alpha = 1, 2, \dots, 10, 20, \dots, 50$	$\alpha = 1$ $1 \leq \beta \leq 269$ $\beta = 260, 250, \dots, 10$ $\beta = 9, 8, \dots, 1$

표 5. 알고리즘 결과 비교

Table 5. Comparison of algorithm results

구분		13 Gen ( $P_d = 1,800$ )	13 Gen ( $P_d = 2,520$ )	40 Gen ( $P_d = 10,500$ )
Known Optimal value		<b>(\$ 17,960.371)</b>	<b>\$ 24,164.05 (?)</b>	<b>(\$ 121,504.702)</b>
IS	AIS	\$ 17,972.900 (?) (+12,529 \$/h)	\$ 24,170.570 (?) (+6,520 \$/h)	\$ 121,489.110 (?) (-15,592 \$/h)
	CIS	\$ 17,964.680 (?) (+4,309 \$/h)	\$ 24,171.130 (?) (+7,080 \$/h)	\$ 121,423.680 (?) (-81,022 \$/h)
	FCIS	\$ 17,964.370 (?) (+3,999 \$/h)	\$ 24,171.090 (?) (+7,040 \$/h)	\$ 121,446.710 (?) (-57,992 \$/h)
BSA		<b>\$ 17,567.943</b> <b>(-392,428 \$/h)</b>	<b>\$ 24,164.051</b> <b>(0.000 \$/h)</b>	<b>\$ 121,491.013</b> <b>(-13,689 \$/h)</b>

## V. 결론

본 논문은 전력발전분야에서 가장 경제적인 비용으로 전력을 생산할 수 있는 경제급전의 최적 해를 구하는 발전정지-교환 알고리즘을 제안하였다. 기존 알고리즘들은 휴리스틱 방법들을 적용하여 다수의 시행 결과 얻은 해들 중에서 최소값을 선택하는 방법을 적용하였음에도 불구하고 최적 해를 구하지 못하였다.

반면에, 본 논문에서는 발전정지 기준을 적용한 방법으로 항상 최소의 비용으로 동일한 결과를 얻을 수 있는 간단하면서도 결정론적인 방법을 제안하였다.

제안된 방법은 3개의 시험사례들에 적용한 결과 기존의 휴리스틱 방법들에 비해 획기적으로 해를 개선하였다.

## References

[1] R. Goncalves, C. Almeida, J. Kuk, and M. Delgado, "Solving Economic Load Dispatch Problem by Natural Computing Intelligent Systems", 15th International Conference on Intelligent System Applications to Power Systems (ISAP), pp. 1-6, 8-12, Nov. 2009.

[2] L. S. Coelho and C. S. Lee, "Solving Economic Load Dispatch Problems in Power Systems using Chaotic and Gaussian Particle Swarm Optimization Approaches," International Journal of Electrical Power & Energy Systems, Vol. 30, Issue. 5, pp. 297-307, June, 2008.

[3] L. S. Coelho and V. C. Mariani, "Combining of

Chaotic Differential Evolution and Quadratic Programming for Economic Dispatch Optimization with Valve-Point Effect", IEEE Trans. on Power Systems, Vol. 21, No. 2, 2006.

[4] S. H. Ling, H. K. Lam, F. H. F. Leung, and Y. S. Lee, "Improved Genetic Algorithm for Economic Load Dispatch with Valve-point Loadings", The 29th Annual Conference of the IEEE Industrial Electronics Society, Vol. 1, pp 442-447, 2003.

[5] T. A. A. Victoire and A. E. Jeyakumar, "Hybrid PSO-SQP for Economic Dispatch with Valve-point Effect", Electric. Power Systems Research, Vol. 71, No. 1, pp. 51-59, 2004.

[6] N. Sinha, R. Chakrabarti, and P. K. Chattopadhyay, "Evolutionary Programming Techniques for Economic Load Dispatch", IEEE Trans. on Evolutionary Computing, Vol. 7, No. 1, pp. 83-94, 2003.

[7] Y. Hou, L. Lu, X. Xiong, and Y. Wu, "Economic Dispatch of Power Systems Based on the Modified Particle Swarm Optimization Algorithm", IEEE/PES Transmission and Distribution Conference and Exhibition, pp. 1-6, 2005.

[8] J. S. Al-Sumait, J. K. Sykulski, and A. K. Al-Othman, "Solution of Different Types of Economic Load Dispatch Problems Using a Pattern Search Method," Electric Power Components and Systems. Vol. 36, No. 3, pp. 250-265, 2008.

[9] L. S. Coelho and V. C. Mariani, "An Improved Harmony Search Algorithm for Power Economic Load Dispatch," Energy Conversion and Management, Vol. 50, pp. 2522-2526, 2009.

[10] K. S. Pothiya, and S. Boonseng, "Distributed Tabu Search Algorithm for Solving the Economic Dispatch Problem", TENCON, Vol. C, pp. 484-487, 2004.

[11] R. Balamurugan and S. Subramanian, "Self-Adaptive Differential Evolution Based Power Economic Dispatch of Generators with Valve-Point Effects and Multiple Fuel Options",

International Journal of Computer Science and Engineering, Vol. 1, No. 1, pp. 10-17, 2007.

[12] A. A. El-Fergany, "Solution of Economic Load Dispatch Problem with Smooth and Non-Smooth Fuel Cost Functions Including Line Losses Using Genetic Algorithm," International Journal of Computer and Electrical Engineering, Vol. 3, No. 5, pp. 706-710, 2011.

[13] N. Sinha and B. Purkayastha, "PSO Embedded Evolutionary Programming Technique for Nonconvex Economic Load Dispatch", IEEE PES-Power Systems Conference and Exposition, Vol. 1, pp. 66-71, 2004.

[14] J. B. Park, K. S. Lee, J. R. Shin, and K. Y. Lee, "A Particle Swarm Optimization for Economic Dispatch with Non-smooth Cost Function", IEEE Trans. Power System, Vol. 20, No. 1, pp. 34-42, 2005.

[15] A. I. Selvakumar, and K. Thanushkodi, "A New Particle Swarm Optimization Solution to Non-convex Economic Dispatch Problems", IEEE Trans. on Power Systems, Vol. 22, No. 1, pp. 42-51, 2007.

[16] J. B. Park, W. N. Lee, and J. R. Sin, "An Improved Particle Swarm Optimization for Economic Dispatch Problems with Non-Smooth Cost Functions," International Journal of Innovations in Energy Systems and Power, Vol. 1, No. 1, pp. 1-7, 2006.

[17] A. Khamseh and YY. Alinejad-Beromi, "Hybrid CLONAL Selection Algorithm with SA for Solving Economic Load Dispatch with Valve-Point Effect," Canadian Journal on Electrical and Electronics Engineering, Vol. 2, No. 10, pp. 463-467, 2011.

[18] P. H. Chen and H. C. Chang, "Large-Scale Economic Dispatch by Genetic Algorithm," IEEE Trans. on Power Systems, Vol. 10, No. 4, pp. 1919-1926, 1995.

[19] P. Attaviriyanupap, H. Kita, E. Tanaka, and J. Hasegawa, "A Hybrid EP and SQP for Dynamic

Economic Dispatch with Non-smooth Fuel Cost Function," IEEE Trans. on Power System, Vol. 17, No. 2, pp. 411-416, 2002.

### 저자 소개

#### 이 상 운(정회원)



- 1987년 : 한국항공대학교 항공전자공학과 (학사)
  - 1997년 : 경상대학교 컴퓨터과학과 (석사)
  - 2001년 : 경상대학교 컴퓨터과학과 (박사)
  - 2003년 : 강원도립대학 컴퓨터응용과 전임강사
  - 2004년 ~ 2007년 2월 : 국립 원주대학 여성교양과 조교수
  - 2007년 3월 ~ 현재 : 강릉원주대학교 멀티미디어공학과 부교수
- <관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 그래프 알고리즘>