

<http://dx.doi.org/10.7236/IIBC.2014.14.3.191>

IIBC 2014-3-27

선형 근사 평활 발전 비용함수를 이용한 경제급전 문제의 최적화

Optimization of Economic Load Dispatch Problem Using Linearly Approximated Smooth Fuel Cost Function

이상운*

Sang-Un Lee*

요약 본 논문은 복잡한 비평활 발전비용함수를 가진 경제급전의 최적화 문제를 풀기 위해 단순히 선형 근사함수를 이용하는 방법을 제안하였다. 제안된 알고리즘은 비평활 발전비용 함수를 선형으로 근사시키고, 요구량이 현재의 발전량을 초과하는 경우 발전단가가 비싼 발전기의 가동을 중지시키고, 발전단가가 보다 큰 발전기의 발전량을 감소시켜 요구량과 발전량의 균형을 맞추는 개념을 도입하였다. 경제급전 문제의 시험사례로 빈번히 활용되고 있는 데이터에 대해 제안된 알고리즘을 적용한 결과 기존의 휴리스틱 알고리즘의 최적화 해를 획기적으로 감소시킬 수 있었으며, 현재 실무적으로 적용되고 있는 2차 평활함수 근사법과 유사한 결과를 얻었다.

Abstract This paper proposes a simple linear function approximation method to solve an economic load dispatch problem with complex non-smooth generating cost function. This algorithm approximates a non-smooth power cost function to a linear approximate function and subsequently shuts down a generator with the highest operating cost and reduces the power of generator with more generating cost in order to balance the generating power and demands. When applied to the most prevalent benchmark economic load dispatch cases, the proposed algorithm is found to dramatically reduce the power cost than does heuristic algorithm. Moreover, it has successfully obtained results similar to those obtained through a quadratic approximate function method.

Key Words : Economic load dispatch, Optimization, Shutdown, Smooth and non-smooth function, Function approximation

1. 서론

전기에너지는 기본적으로 저장이 불가능하므로 시간 대별로 총 부하량 (수요 예측량)과 일치하도록 총 발전량을 맞추어야 한다. 만약, 발전량이 부하량보다 적을 경우 주파수 난조와 정전 발생으로 전력계통에 큰 문제를 발

생시키고, 산업 전반에 걸쳐 경제적 손실도 막대해진다.

각기 다른 발전비용 함수 $F(P_i)$ 를 가진 n 대의 발전기 G_i , ($i=1,2,\dots,n$)가 존재한다. 이 경우, 주어진 시간대의 부하 P_d 를 충족시키기 위한 전체 발전량 $\sum_{i=1}^n P_i$ 를 발전하기 위해 각 발전기의 출력 P_i 를 어떻게 효율적으로 배

*정회원, 강릉원주대학교 과학기술대학 멀티미디어공학과
접수일자 : 2013년 2월 11일, 수정완료 : 2014년 5월 3일
게재확정일자 : 2014년 6월 13일

Received: 11 February, 2014 / Revised: 3 May, 2014 /

Accepted: 13 June, 2013

*Corresponding Author: sulee@gwnu.ac.kr

Dept. of Multimedia Eng., Gangneung-Wonju National University, Korea

분해야 가장 적은 비용이 소요되는가를 결정 (경제적 발전)하는 문제를 경제부하배분 (Economic Load Dispatch, ELD) 또는 간단히 경제급전 ((Economic Dispatch, ED) 이라 한다.

2005년 기준 우리나라의 전력시장 정산금액 총 17조 원 중 경제급전 정산금액은 약 8조원으로 47%를 차지하고 있다. 따라서 경제급전은 동일한 발전량을 유지함에 도 불구하고 최적의 알고리즘을 적용하여 소요비용을 최대한 절약할 수 있는 문제로 전력산업에서 중요한 최적화 문제 가운데 하나이다.

경제급전 최적화와 관련하여 초기에는 대부분 수학적 최적화 기법으로는 라그랑주 완화법 (Lagrangian relaxation) 이나 경사법 (Gradient method)을 적용하였으나 더 이상의 최적화 성능을 갖지를 못해 인공지능 기법을 연구하고 있다. 인공지능 기법으로는 유전자 알고리즘 (Genetic algorithm, GA), 신경망 (neural network, NN), 퍼지 (Fuzzy), 군집 알고리즘 (Particle Swarm Optimization, PSO)등 비결정적이며, 휴리스틱 알고리즘을 적용하고 있다.

경제급전 문제는 다른 분야의 최적화 문제보다도 어려운 분야이다. 왜냐하면 발전비용이 고정된 상수값이 아니라 출력에 높이기 위해 밸브를 열 때 발생하는 진동으로 인한 밸브효과 (valve-effect)로 인해 $F(P_i) = a_i P_i^2 + b_i P_i + c_i + |e_i \times \sin(f_i \times (P_i^{\min} - P_i))|$ 로 2차함수보다도 복잡한 비선형 형태를 갖기 때문이다. 이러한 복잡한 비선형 발전비용 함수에 대해 수학적 방법은 $F'(P_i) = a_i' P_i^2 + b_i' P_i + c_i'$ 로 근사시킨 평활함수 (smooth function)에 대한 기울기 $2a_i' P_i + b_i'$ 를 이용해 해를 구한다.

인공지능 기법은 초기치를 랜덤하게 설정하고 수 많은 시행 결과 최소값을 해로 결정하기 때문에 동일한 해를 검증하기가 사실상 불가능하다. 또한, 다양한 휴리스틱 알고리즘을 적용함에도 불구하고, 대부분의 결과들은 거의 비슷한 결과를 얻고 있으며, 획기적으로 해를 개선한 알고리즘을 찾기가 현실적으로 불가능하다.

본 논문에서는 평활 근사함수에 대해 수학적이거나 인공지능 알고리즘들에 비해 경제급전 문제의 해를 개선할 수 있으면서도 쉽게 발전량을 구하는 알고리즘을 제안한다. 2장에서는 경제급전과 관련된 연구의 문제점과 실제 적용사례를 고찰해 본다. 3장에서는 평활근사 함수에 대한 경제급전 문제의 해를 간단히 구하는 알고리즘을 제안한다. 4장에서는 제안된 알고리즘을 사례들에 적용하

여 성능을 검증한다.

II. 관련연구 문제점과 실제 적용사례

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 ΣP_i 는 식 (1)과 같이 수급 균형을 맞추기 위해 총 부하량 P_d 와 동일해야 한다. 여기서 총 부하량 P_d 는 수요 예측량과 공급예비율의 합이다.

$$\sum_{i=1}^n P_i = P_d \quad (1)$$

발전기 n 개는 원자력, 유연탄, 무연탄, 중유, LNG, 수력, 풍력, 조력, 양수, 태양력 등 발전단가가 상이하며, 비록 동일한 종류의 발전기라 하더라도 성능 차이로 인해 발전효율성이 차이가 발생할 수 있다. 따라서 각 발전기의 발전비용에는 차이가 발생한다고 가정한다. 또한, 총 요구량 P_d 는 $\Sigma P_i^{\min} \leq P_d \leq \Sigma P_i^{\max}$ 범위를 갖는다고 가정한다.

경제급전 문제의 목적함수는 경제급전 총 발전비용을 최소화시키는 것으로 식 (2)와 같다.^[1]

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

여기서 $F(P_i)$ 는 발전기 i 의 발전비용 (\$/MWh)으로 식 (3)으로 표현된다. 기본적으로는 발전량 P_i 에 따른 발전비용은 $a_i P_i^2 + b_i P_i + c_i$ 이나 발전량을 증가시키기 위해 밸브를 열면 진동이 발생하여 발전량이 감소하며, 이를 상쇄하기 위한 비용 $|e_i \times \sin(f_i \times (P_i^{\min} - P_i))|$ 이 추가로 소요된다. 이는 그림 1에 제시되어 있다.^[1-3]

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

$$F'(P_i) = a_i' P_i^2 + b_i' P_i + c_i' \quad (4)$$

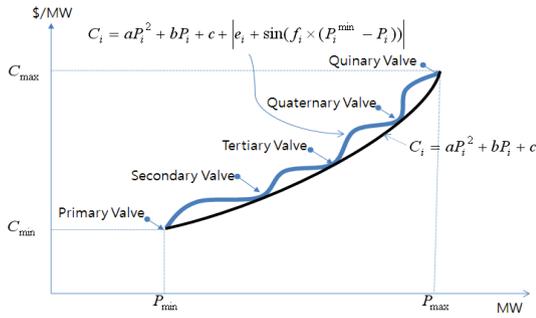


그림 1. 발전비용 함수
 Fig. 1. Power Cost Function

여기서 P_i 는 발전기 i 의 발전량 (KW)이며, 각 발전기의 출력 범위 제한조건은 $P_i^{min} \leq P_i \leq P_i^{max}$ 이다. 일단 발전기가 가동된 이후 안전하게 연속적으로 운전이 가능한 최저수준의 출력에 도달해야 발전기를 전력계통에 연결시켜 전력을 공급하게 되며, 이 시점이 P_i^{min} 이다. 이는 석탄 발전기는 정격의 30 ~ 40%, 중유 발전기는 25 ~ 35% 정도로 알려져 있다.

한국전력공사의 2010.12월 기준 원자력(39.7원/KWh)과 유연탄(60.9원)과 국내탄(110,1원)은 기저부하를, 열복합(126.7원)과 중유(184.7원)는 중간부하를, LNG(147.2원)는 기동 및 출력 조절이 용이하여 주파수조절용으로, 수력(171.4원)은 용량이 적어 첨두부하를 담당하도록 운영되고 있다. 또한, 기저부하를 담당하는 석탄 발전기는 발전단가가 저렴하고, 재가동시 10시간 이상 소요되며, 가동중지와 재가동에 어려움이 있어 원자력발전소와 더불어 가능한 최대 발전량을 유지한다. 따라서, 경제급전의 주요 대상 발전기는 LNG나 중유 발전기가 된다.

기존의 관련연구들은 식 (3)을 최적화하기 위한 인공지능법을 적용하나 $P_i^{min} \leq P_i \leq P_i^{max}$ 로 한정시키고 있다. 이는 고비용 발전기에 대해서도 항상 발전을 수행한다는 가정하에 최적화를 수행하는 단점이 있다.

반면에, 2011년 현재 한국전력공사의 전력거래소에서는 식 (4)의 2차 평활함수를 적용한 라그랑주 완화법으로 RSS (Resource Scheduling & Commitment)를 수행하고 있다. 또한, 수요에 따른 발전기 가동중지 (기동중지)와 발전량을 배분하고 있다. 즉, 고비용 발전기에 대해서는 현실적으로 $P_i = 0$ 도 적용하기 때문에, 수학적 방법보다 성능이 좋은 인공지능법에 적용되는 이론적 최적화 제약 조건 $P_i^{min} \leq P_i \leq P_i^{max}$ 은 현실과 괴리를 나타내고 있다.

표 1은 Coelho와 Lee^[3]가 제시한 13-발전기와 40-발전기의 사례이다.

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	f	
1	0	680	0.00028	8.10	550	300	0.035
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	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	0.02028	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	$P_d = 10,500 MW$				

Coelho와 Mariani^[2]는 13-발전기 데이터의 최소 발전 총량 $\Sigma P_i^{min} = 550 MW$, 최대 발전 총량은 $\Sigma P_i^{max} = 2,960 MW$, 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. 이 사례에서는 발전기 사용 연료가 제시되지 않고 있어 $P_i = 0$ 이 될 수 있는

표 2. 시험사례의 최적화 결과
Table 2. Results of Optimization for the Test Cases

Case with 13 Generators and Load Demand of 1,800 MW				
Method	Min Cost	Mean Cost	Max Cost	St Dev
Artificial Immune System (IS) ^[1]	17,972.90	17,976.14	17,989.84	5.94
Cultural IS (CIS) ^[1]	17,964.68	17,974.28	17,989.23	4.81
Fuzzy CIS (FCIS) ^[1]	17,964.37	17,974.50	17,989.81	3.19
Chaotic Differential Evolution and Quadratic Programming (DEC-SQP) ^[2]	17,963.94	17,973.13	17,984.81	1.97
Improved Genetic Algorithm (IGA) ^[4]	18,063.58	18,096.40	18,293.47	45.79
Modified Particle Swarm Optimization (MPSO) ^[5]	17,973.34	-	-	-
Evolutionary Programming (EP) ^[6]	17,994.07	-	-	-
Particle Swarm Optimization (PSO) ^[7]	18,030.72	-	-	-
Hybrid Evolutionary Programming with SQP (HEP-SQP) ^[7]	17,991.03	-	-	-
Hybrid Particle Swarm with SQP (HPS-SQP) ^[7]	17,969.93	-	-	-
Pattern Search (PS) ^[8]	17,969.17	-	-	-

Case with 13 Generators and Load Demand of 2,520 MW				
Method	Min Cost	Mean Cost	Max Cost	St Dev
Artificial Immune System (IS) ^[1]	24,170.57	24,193.72	24,289.03	35.52
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
Self-Adaptive Differential Evolution (SDE) ^[9]	24,164.05	24,168.28	24,200.05	-
Distributed Tabu Search Algorithm (DTSA) ^[10]	24,169.95	-	-	-

Case with 40 Generators and Load Demand of 10,500 MW				
Method	Min Cost	Mean Cost	Max Cost	St Dev
Artificial Immune System (IS) ^[1]	121,489,110	121,737.73	122,163.56	106.88
Cultural Immune System (CIS) ^[1]	121,423,680	121,717.97	122,147.93	144.81
Fuzzy CIS (FCIS) ^[1]	121,446,710	121,662.67	121,863.39	138.04
Chaotic Differential Evolution and Quadratic Programming (DEC-SQP) ^[2]	121,741,970	122,295.12	122,839.29	386.18
New Particle Swarm Optimization Solution to Nonconvex (NPSO-LRS) ^[11]	121,664,430	122,209.31	122,981.59	-
Embedded Evolutionary Programming-PSO (CEP-PSO) ^[12]	123,670,000	124,145.60	124,900.00	-
Evolutionary Programming (EP) ^[6]	122,624,350	-	-	-
Particle Swarm Optimization (PSO) ^[7]	122,930,450	-	-	-
Modified Particle Swarm Optimization (MPSO) ^[13]	122,252,265	-	-	-
Hybrid Evolutionary Programming with SQP (HEP-SQP) ^[7]	122,379,630	-	-	-
Hybrid Particle Swarm with SQP (HPS-SQP) ^[7]	122,094,670	-	-	-
Improved PSO (IPSO) ^[14]	121,495,702	121,699.301	122,168.116	155.619
Genetic Algorithm (GA) ^[15]	119,732,250	-	-	-

대상 발전기는 알 수 없다. 표 1의 시험사례들에 대해 최적화 과정을 수행한 연구 결과는 표 2에 제시되어 있다.

전량을 $P_i = 0$ 로 설정하고, 남은 $\Sigma P_i - P_d > 0$ 에 대해서는 다음 우선순위 발전기에 대해 $P_i = P_i - (\Sigma P_i - P_d)$ 로 설정한다. 제안된 알고리즘은 MW 단위로 수행된다.

III. 선형 평활 근사함수의 최적화 알고리즘

본 장에서는 비평활 발전비용함수 $F(P_i) = a_i P_i^2 + b_i P_i + c_i + |e_i \times \sin(f_i \times (P_i^{\min} - P_i))|$ 를 2차 평활함수 $F'(P_i) = a_i' P_i^2 + b_i' P_i + c_i'$ 에 비해 단순하면서도 유사한 성능을 갖는 선형 평활함수 $F''(P_i) = b_i'' P_i + c_i''$ 로 근사시킨 값을 적용하는 알고리즘을 제안한다.

제안 알고리즘은 발전 정지 기준으로 $F(P_i^{\max})/P_i^{\max}$ 인 최대값 비율과 $2a_i' P_i^{\max} + b_i'$ 또는 b_i'' 의 기울기를 적용하여 최대값 발전기 순서로 $\Sigma P_i - P_i \geq P_d$ 인 G_i 의 발

Step 1. $P_i \leftarrow P_i^{\max}$ 로 설정

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

$$F''(P_i) = b_i'' P_i + c_i''$$

$F(P_i^{\max}), F''(P_i^{\max})$ 계산

Step 2. $\Sigma P_i \geq P_d$ 가 될 때까지 다음 기준의 최대값 우선순위 발전기 G_i 에 대해 $\Sigma P_i - P_i \geq P_d$ 이면 $P_i = 0$ 로 설정. 다음 표의 r_{kl} 을 구함.
($k = 0, 1, 2, l = 0, 1, 2$)

$$r_{k0} = F(P_i^{\max})/P_i^{\max}$$

$$r_{k1} = b_i''$$

	k	l		
		0	1	2
비평활함수	0	r_{00}	-	-
1차 평활함수	1	r_{10}	r_{11}	-

Step 3.의 3가지 기준에 대해 $\Sigma P - P_d > 0$ 이면 다음 최대값 우선순위 발전기 G_i 에 대해 $P_i = P_i - (\Sigma P_i - P_d)$ 로 설정, 총 발전비용 $\Sigma F(P_i)$ 를 구함.

단, $P_i - (\Sigma P_i - P_d) \geq P_i^{\min}$ 이면 $P_i = P_i - (\Sigma P_i - P_d)$ 로, 그렇지 않으면 $P_i = P_i^{\min}$, 다음 우선순위 발전기에 대해 $P_i = P_i - (\Sigma P_i - P_d)$ 로 설정.

Step 4. Step 3의 5개 기준에 대해 총 발전비용이 최소가 되는 값을 해로 결정.

$$\min \{ \Sigma Fr_{00}(P_i), \Sigma Fr_{10}(P_i), \Sigma Fr_{11}(P_i), \Sigma Fr_{20}(P_i), \Sigma Fr_{22}(P_i) \}$$

제안된 알고리즘은 결정론적인 알고리즘으로 항상 동일한 결과를 얻으며, 발전비용이 아무리 복잡한 함수로 표현되어도 이에 영향을 받지 않으며, 발전기 대수 n 이 큰 값이라도 이와 무관하게 간단히 수행될 수 있는 장점이 있다.

IV. 적용 결과 및 분석

실험은 2차 평활함수와 1차 (선형) 평활함수의 성능을 비교하고자 한다. 표 1의 13-발전기 비평활 발전비용 함수에 대해 2차 평활함수와 1차 평활함수로 근사시킨 함수 데이터는 표 3에 제시되어 있다. 예로, G_4 발전기의 평활 근사함수를 구한 결과는 그림 2와 같다. 표 1의 40-발전기 비평활 발전비용 함수에 대해 2차 평활함수와 1차 평활함수로 근사시킨 함수 데이터는 표 4에 제시되어 있다.

표 3. 13-발전기의 평활 근사함수

Table 3. Approximated smooth function for 13 Generators

$a_i' P_i^2 + b_i' P_i + c_i'$ & $b_i'' P_i + c_i''$ for 13 Generators						
G_i	P_i^{\min}	P_i^{\max}	a_i'	b_i', b_i''	c_i', c_i''	R^2
1	0	680	0.00034	8.07289	742.07005	0.99680
			-	8.30033	715.90934	0.99675
2	0	360	0.00095	8.01022	438.22236	0.99518
			-	8.35150	417.80285	0.99507
3	0	360	0.00085	8.03267	403.91677	0.99728
			-	8.33902	385.58654	0.99718
4	60	180	0.00012	8.36093	309.36368	0.98926
			-	8.92333	260.94341	0.99667
5	60	180	0.00012	8.36093	309.36368	0.98926
			-	8.92333	260.94341	0.99667
6	60	180	0.00012	8.36093	309.36368	0.98926
			-	8.92333	260.94341	0.99667
7	60	180	0.00012	8.36093	309.36368	0.98926
			-	8.92333	260.94341	0.99667
8	60	180	0.00012	8.36093	309.36368	0.98926
			-	8.92333	260.94341	0.99667
9	60	180	0.00012	8.36093	309.36368	0.98926
			-	8.92333	260.94341	0.99667
10	40	120	0.00535	9.55635	168.53698	0.99040
			-	8.91487	181.25970	0.99006
11	40	120	0.00535	9.55635	168.53698	0.99040
			-	8.91487	181.25970	0.99006
12	55	120	0.00535	9.55635	168.53698	0.99040
			-	8.91487	181.25970	0.99006
13	55	120	0.00535	9.55635	168.53698	0.99040
			-	8.91487	181.25970	0.99006
Sum:	550	2,960	$P_d = 1,800 MW$ and $P_d = 2,520 MW$			

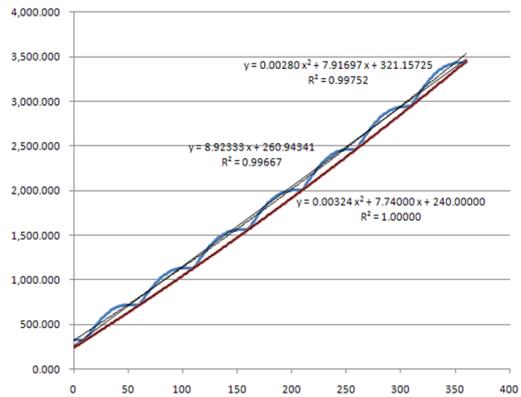


그림 2. G_4 발전기의 발전비용함수 근사

Fig. 2. Approximated smooth fuel cost function for G_4

3장에서 제안된 선형평활 근사 함수와 2차 평활 근사 함수로 최적화를 수행한 결과는 표 5에 제시되어 있다.

표 4. 40-발전기의 평활 근사함수
Table 4. Approximated smooth function for 40 Generators

$a_i' P_i^2 + b_i' P_i + c_i'$ & $b_i'' P_i + c_i''$ for 40 Generators						
G_i	P_i^{\min}	P_i^{\max}	a_i'	b_i', b_i''	c_i', c_i''	R^2
1	36	114	0.00162	7.22882	151.80479	0.98481
			-	7.41381	148.32094	0.98477
2	36	114	0.00162	7.22882	151.80479	0.98481
			-	7.41381	148.32094	0.98477
3	60	120	0.02706	6.23638	391.81778	0.98194
			-	9.50360	327.41465	0.98415
4	80	190	0.00885	8.15071	473.02383	0.99281
			-	9.83306	420.02380	0.98086
5	47	97	0.00994	7.42169	185.57745	0.96544
			-	6.45750	201.00460	0.96398
6	68	140	0.01427	7.51359	300.44004	0.99445
			-	9.61114	254.16334	0.99156
7	110	300	0.00420	7.86504	421.19928	0.99404
			-	9.12575	338.57370	0.99277
8	135	300	0.00477	6.90635	536.55637	0.99313
			-	8.33628	465.29807	0.99117
9	135	300	0.00558	6.51535	600.33637	0.99289
			-	8.18828	516.96857	0.99011
10	130	300	0.00617	12.73644	871.08853	0.99774
			-	14.58354	778.73231	0.99666
11	94	375	0.00518	12.97334	747.62629	0.99860
			-	14.91434	626.63684	0.99746
12	94	375	0.00572	12.87334	767.11629	0.99862
			-	15.01684	633.50434	0.99725
13	125	500	0.00360	12.78241	1079.19452	0.99810
			-	14.58277	929.46527	0.99708
14	125	500	0.00691	9.12241	1926.19492	0.99746
			-	12.57777	1638.82444	0.99244
15	125	500	0.00647	9.43241	1894.09492	0.99749
			-	12.66777	1625.02110	0.99315
16	125	500	0.00647	9.43241	1894.09492	0.99749
			-	12.66777	1625.02110	0.99315
17	220	500	0.00270	8.15777	824.33860	0.99546
			-	9.50542	712.25850	0.99412
18	220	500	0.00270	8.13777	826.17860	0.99544
			-	9.48542	714.09850	0.99410
19	242	550	0.00337	7.97390	864.63746	0.99645
			-	9.64860	694.93184	0.99400
20	242	550	0.00337	7.79190	864.61748	0.99645
			-	9.64860	694.93184	0.99400
21	254	550	0.00299	6.55893	95.31019	0.98519
			-	8.20543	844.65605	0.99252
22	254	550	0.00299	6.55893	95.31019	0.98519
			-	8.20543	844.65605	0.99252
23	254	550	0.00285	6.58993	1003.88019	0.99514
			-	8.15843	860.27155	0.99268
24	254	550	0.00285	6.58993	1003.88019	0.99514
			-	8.15843	860.27155	0.99268
25	254	550	0.00278	7.02893	1010.67019	0.99558
			-	8.53993	870.58430	0.99345
26	254	550	0.00278	7.02893	1010.67019	0.99558
			-	8.53993	870.58430	0.99345
27	10	150	0.51840	3.82022	1113.29078	0.99990
			-	81.58083	-81.76435	0.94209
28	10	150	0.51840	3.82022	1113.29078	0.99990
			-	81.58083	-81.76435	0.94209
29	10	150	0.51840	3.82022	1113.29078	0.99990
			-	81.58083	-81.76435	0.94209
30	47	97	-0.00994	7.42169	185.57745	0.96544
			-	6.45750	201.00460	0.96398
31	60	190	0.00104	6.67370	300.89938	0.98536
			-	6.87202	294.65283	0.98531
32	60	190	0.00104	6.67370	300.89938	0.98536
			-	6.87202	294.65283	0.98531
33	60	190	0.00104	6.67370	300.89938	0.98536
			-	6.87202	294.65283	0.98531
34	90	200	-0.00289	9.69189	194.70866	0.98933
			-	9.12082	213.84823	0.98966
35	90	200	-0.00289	9.36789	203.41866	0.98594
			-	8.79082	222.55823	0.98566
36	90	200	-0.00289	9.36789	203.41866	0.98594
			-	8.79082	222.55823	0.98566
37	25	110	0.02299	5.11161	375.49374	0.99101
			-	7.63998	329.56158	0.98370
38	25	110	0.02299	5.11161	375.49374	0.99101
			-	7.63998	329.56158	0.98370
39	25	110	0.02299	5.11161	375.49374	0.99101
			-	7.63998	329.56158	0.98370
40	242	550	0.00337	7.79390	864.63748	0.99645
			-	9.64860	694.93184	0.99400
Sum:	4,817	12,722	-	$P_d = 10,500 MW$		

13 Unit ($P_d = 1,800.MW$)와 13 Unit ($P_d = 2,520.MW$)에서는 2차 평활함수-최대값 비율과 1차 평활함수-최대값 비율이 동일한 결과를 나타내었다. 단지, 40 Unit ($P_d = 10,500.MW$)에서는 2차 평활함수-기울기가 가장 좋은 결과를 얻었으나 차이는 미미한 것으로 나타내었다.

결론적으로 2차 평활함수나 1차 평활함수 어떤 것을 사용하여도 유사한 결과를 얻기 때문에 보다 단순한 1차 평활함수를 적용하는 것이 보다 간단히 해를 구할 수 있다.

V. 결론

본 논문은 전력발전분야에서 가장 경제적인 비용으로 전력을 생산할 수 있는 경제급전의 최적화 문제에 대한 최적의 해를 선형 근사함수로 구하는 알고리즘을 제안하였다. 기존 알고리즘들은 휴리스틱 방법들을 적용하여 다수의 시행 결과 얻은 해들 중에서 최소값을 선택하는 방법으로, 동일한 해를 구하지 못해 검증이 사실상 불가능한 단점이 있다.

반면에, 본 논문에서는 복잡한 발전비용 함수를 단순히 선형함수로 근사시키고 최대값 비율이나 기울기 기준을 적용하여 해를 구하는 방법을 제안하였다.

제안된 방법은 현재 실무적으로 적용되고 있는 2차 평활함수 근사 방법과 유사한 결과를 나타내고 있으며, 추가적으로 보다 단순하고 빠르게 해를 구하는 방법으로 실무적으로 보다 활용도가 뛰어남을 알 수 있다.

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] 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, May. 2006.

표 5. 시험사례의 가동중지 균형-교환 최적화 알고리즘 결과
 Table 5. Results of the -Balance and Swap Optimization Algorithm for Test Cases

Case Study 13 Units and $P_d = 1,800 MW$																
구분	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}	G_{13}	ΣP_i	$\Sigma F(P_i)$ 근사값(실제값)	
P_i^{\min}, P_i^{\max}	0 680	0 360	0 360	60 180	60 180	60 180	60 180	60 180	60 180	40 120	40 120	55 120	55 120	2,960	29,583.599	
비평활함수-최대값비율	680	360	360	0	0	0	0	0	60	100	120	120	0	0	1,800	- (17,598.514)
2차평활함수-최대값비율	680	360	360	0	0	0	60	160	180	0	0	0	0	1,800	17,466.606 (17,427.968)	
2차평활함수-기울기	680	40	0	180	180	180	180	180	180	0	0	0	0	1,800	17,950.157 (18,602.160)	
1차평활함수-최대값비율	680	360	360	0	0	0	60	160	180	0	0	0	0	1,800	17,526.509 (17,427.968)	
1차평활함수-기울기	680	360	360	0	0	0	0	0	0	40	120	120	120	1,800	17,465.333 (17,526.498)	

Case Study 13 Units and $P_d = 2,520 MW$																
구분	G_1	G_2	G_3	G_4	G_5	G_6	G_7	G_8	G_9	G_{10}	G_{11}	G_{12}	G_{13}	ΣP_i	$\Sigma F(P_i)$ 근사값(실제값)	
P_i^{\min}, P_i^{\max}	0 680	0 360	0 360	60 180	60 180	60 180	60 180	60 180	60 180	40 120	40 120	55 120	55 120	2,960	29,583.599	
비평활함수-최대값비율	680	360	360	0	160	180	180	180	180	120	120	0	0	2,520	- (24,839.529)	
2차평활함수-최대값비율	680	360	360	165	180	180	180	180	180	0	0	0	55	2,520	24,612.850 (24,936.764)	
2차평활함수-기울기	680	360	345	180	180	180	180	180	180	0	0	0	55	2,520	25,608.906 (25,094.470)	
1차평활함수-최대값비율	680	360	360	165	180	180	180	180	180	0	0	0	55	2,520	24,914.931 (24,936.764)	
1차평활함수-기울기	680	360	360	0	0	100	180	180	180	120	120	120	120	2,520	24,933.228 (25,072.127)	

Case Study 40 Units and $P_d = 10,500 MW$												
G_i	$P_{i,\min}$	$P_{i,\max}$	비평활함수-최대값비율		2차평활함수-최대값비율		2차평활함수-기울기		1차평활함수-최대값비율		1차평활함수-기울기	
			P_i	$F(P_i)$	P_i	$F(P_i)$	P_i	$F(P_i)$	P_i	$F(P_i)$	P_i	$F(P_i)$
1	36	114	114	978.156	114	978.1563	114	978.1563	114	978.1563	114	978.1563
2	36	114	114	978.156	114	978.1563	114	978.1563	114	978.1563	114	978.1563
3	60	120	120	1,544.653	120	1,544.6534	120	1,544.6534	120	1,544.6534	120	1,544.6534
4	80	190	190	2,353.689	190	2,353.6891	190	2,353.6891	190	2,353.6891	190	2,353.6891
5	47	97	97	853.178	97	853.1776	97	853.1776	97	853.1776	97	853.1776
6	68	140	140	1,596.464	140	1,596.4643	140	1,596.4643	140	1,596.4643	140	1,596.4643
7	110	300	300	3,216.424	300	3,216.4240	300	3,216.4240	300	3,216.4240	300	3,216.4240
8	135	300	300	3,052.310	300	3,052.3095	300	3,052.3095	300	3,052.3095	300	3,052.3095
9	135	300	300	3,071.990	300	3,071.9895	300	3,071.9895	300	3,071.9895	300	3,071.9895
10	130	300	0	0.000	0	0.0000	0	0.0000	0	0.0000	0	0.0000
11	94	375	0	0.000	0	0.0000	0	0.0000	0	0.0000	0	0.0000
12	94	375	0	0.000	0	0.0000	0	0.0000	0	0.0000	0	0.0000
13	125	500	0	0.000	0	0.0000	0	0.0000	0	0.0000	0	0.0000
14	125	500	500	8,219.407	500	8,219.4073	278	5,039.2558	500	8,219.4073	500	8,219.4073
15	125	500	278	5,039.331	278	5,039.3308	500	8,232.3073	278	5,039.3308	278	5,039.3308
16	125	500	500	8,232.307	500	8,232.3073	500	8,232.3073	500	8,232.3073	500	8,232.3073
17	220	500	500	5,525.294	500	5,525.2937	500	5,525.2937	500	5,525.2937	500	5,525.2937
18	220	500	500	5,517.134	500	5,517.1337	500	5,517.1337	500	5,517.1337	500	5,517.1337
19	242	550	550	6,271.211	550	6,271.2111	550	6,271.2111	550	6,271.2111	550	6,271.2111
20	242	550	550	6,271.191	550	6,271.1911	550	6,271.1911	550	6,271.1911	550	6,271.1911
21	254	550	550	5,575.329	550	5,575.3293	550	5,575.3293	550	5,575.3293	550	5,575.3293
22	254	550	550	5,575.329	550	5,575.3293	550	5,575.3293	550	5,575.3293	550	5,575.3293
23	254	550	550	5,558.049	550	5,558.0493	550	5,558.0493	550	5,558.0493	550	5,558.0493
24	254	550	550	5,558.049	550	5,558.0493	550	5,558.0493	550	5,558.0493	550	5,558.0493
25	254	550	550	5,785.664	550	5,785.6643	550	5,785.6643	550	5,785.6643	550	5,785.6643
26	254	550	550	5,785.664	550	5,785.6643	550	5,785.6643	550	5,785.6643	550	5,785.6643
27	10	150	0	0.000	0	0.0000	0	0.0000	0	0.0000	0	0.0000
28	10	150	0	0.000	0	0.0000	0	0.0000	0	0.0000	0	0.0000
29	10	150	0	0.000	0	0.0000	0	0.0000	0	0.0000	0	0.0000
30	47	97	97	853.178	97	853.1776	97	853.1776	97	853.1776	97	853.1776
31	60	190	190	1,643.991	190	1,643.9913	190	1,643.9913	190	1,643.9913	190	1,643.9913
32	60	190	190	1,643.991	190	1,643.9913	190	1,643.9913	190	1,643.9913	190	1,643.9913
33	60	190	190	1,643.991	190	1,643.9913	190	1,643.9913	190	1,643.9913	190	1,643.9913
34	90	200	200	2,101.017	200	2,101.0170	200	2,101.0170	200	2,101.0170	200	2,101.0170
35	90	200	200	2,043.727	200	2,043.7270	200	2,043.7270	200	2,043.7270	200	2,043.7270
36	90	200	200	2,043.727	200	2,043.7270	200	2,043.7270	200	2,043.7270	200	2,043.7270
37	25	110	110	1,220.166	110	1,220.1661	110	1,220.1661	110	1,220.1661	110	1,220.1661
38	25	110	110	1,220.166	110	1,220.1661	110	1,220.1661	110	1,220.1661	110	1,220.1661
39	25	110	110	1,220.166	110	1,220.1661	110	1,220.1661	110	1,220.1661	110	1,220.1661
40	242	550	550	6,271.211	550	6,271.2111	550	6,271.2111	550	6,271.2111	550	6,271.2111
계	4,817	12,722	10,500.00	- (118,484.313)	10,500	116,619.48 (118,484.313)	10,500	116,612.313 (118,477.138)	10,500	114,870.826 (118,484.313)	10,500	114,870.826 (118,484.313)

- [3] 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, No. 5, pp. 297-307, Jun. 2008.
- [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, Nov. 2003.
- [5] 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, Aug. 2005.
- [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, Feb. 2003.
- [7] 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, Sep. 2004.
- [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, Mar. 2008.
- [9] 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, Mar. 2007.
- [10] K. S. Pothiya, and S. Boonseng, "Distributed Tabu Search Algorithm for Solving the Economic Dispatch Problem", *TENCON*, Vol. C, pp. 484-487, Nov. 2004.
- [11] A. I. Selvakumar, and K. Thanushkodi, "A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems", *IEEE Trans. on Power Systems*, Vol. 22, No. 1, pp. 42-51, Feb. 2007.
- [12] 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, Oct. 2004.
- [13] 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, Feb. 2005.
- [14] 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, Jun. 2006.
- [15] 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, Oct. 2011.

저자 소개

이 상 윤(정회원)



- 1987년 : 한국항공대학교 항공전자공학(학사)
- 1997년 : 경상대학교 컴퓨터학과(석사)
- 2001년 : 경상대학교 컴퓨터학과(박사)
- 2003년 : 강원도립대학 컴퓨터응용과 전임강사

- 2004년~2007.2 : 국립 원주대학 여성교양과 조교수
- 2007.3~현재 : 강릉원주대학교 멀티미디어공학과 부교수
- 관심분야 : 소프트웨어 프로젝트 관리, 개발 방법론, 분석과 설계 방법론, 시험 및 품질보증, 소프트웨어 신뢰성, 그래프 알고리즘
- E-mail : sulee@gwnu.ac.kr