

## 비평활 발전비용함수를 가진 동적 경제급전문제의 교환 최적화

이 상 운\*

# A Swap Optimization for Dynamic Economic Dispatch Problem with Non-smooth Function

Sang-Un, Lee \*

### 요 약

본 논문은 동적 경제급전의 최적화 문제를 풀기 위해 교환 알고리즘을 제안하였다. 제안된 알고리즘은 첫 번째로, 발전단가  $C_i^{\max}/P_i^{\max}$ 가 비싼 발전기는 가동을 중지시키는 개념을 도입하여 총 요구량  $P_d$ 와 총 발전량  $\Sigma P_i$ 의 균형을 맞추었다. 다음으로 발전량을  $P_i = P_i \pm \Delta$ , ( $\Delta = 1.0, 0.1, 0.01, 0.001$ )에 대해  $\max [F(P_i) - F(P_i - \Delta)] > \min [F(P_j + \Delta) - F(P_j)]$ ,  $i \neq j$ 이면  $P_i = P_i - \Delta$ ,  $P_j = P_j + \Delta$ 로 발전량을 교환하는 방법을 적용하였다. 동적 경제급전 문제의 시험사례에 제안된 알고리즘을 적용한 결과 기존의 휴리스틱 알고리즘 최적화 발전비용을 크게 감소시켜 경제적인 이익을 극대화 시켰다.

▶ Keywords : 동적 경제급전, 최적화, 가동중지, 교환, 평활과 비평활함수

### Abstract

This paper proposes Swap algorithm for solving Dynamic Economic Dispatch (DED) problem. The proposed algorithm initially balances total load demand  $P_d$  with total generation  $\Sigma P_i$  by deactivating a generator with the highest unit generation cost  $C_i^{\max}/P_i^{\max}$ . It then swaps generation level  $P_i = P_i \pm \Delta$ , ( $\Delta = 1.0, 0.1, 0.01, 0.001$ ) for  $P_i = P_i - \Delta$ ,  $P_j = P_j + \Delta$  provided that  $\max [F(P_i) - F(P_i - \Delta)] > \min [F(P_j + \Delta) - F(P_j)]$ ,  $i \neq j$ . This new algorithm is applied and tested to the experimental data of Dynamic Economic Dispatch problem, demonstrating a considerable reduction in the prevalent heuristic algorithm's optimal generation cost and in the maximization of economic profit.

▶ Keywords : Dynamic Economic Dispatch, Optimization, Shutdown, Swap, smooth and non-smooth function

•제1저자 : 이상운

•투고일 : 2012. 08. 29. 심사일 : 2012. 09. 16. 게재확정일 : 2012. 10. 04.

\* 강릉원주대학교 멀티미디어공학과 (Dept. of Multimedia Eng., Gangneung-Wonju National University)

## I. 서론

전기에너지는 기본적으로 저장이 불가능하므로 모든 시간 대별로 총 발전량은 손실을 포함한 총 부하량과 동일 또는 여유분을 갖도록 준비하여야 한다. 만약, 발전량이 부하량보다 적을 경우 주파수 난조 등 전체 계통에 큰 문제가 발생한다.

주어진 시간대의 부하를 공급할 수 있도록 발전비용이 각각 다른  $n$ 대의 발전기 출력을 각각 어떻게 배분해야 전체 발전비용(생산비용)이 가장 경제적인가를 결정하는 과정을 경제부하배분(Economic Load Dispatch, ELD) 또는 간단히 경제급전(Economic Dispatch, ED)이라 한다. 전통적인 경제급전 문제는 단위시간에 대한 경제급전에 한정된 연구이며, 이를 정적경제급전(Static ED, SED)라 한다.[1] 반면에, 동적 경제급전(Dynamic ED, DED)은 24시간 범위에 대해 시간대별로 정적경제급전문제를 확대한 경우로, 실제 운영시 보다 현실적인 발전계획에 적용된다.[2] 경제급전 대상은 원자력, 화력(석탄, 중유, LNG), 열복합, 태양력, 수력, 조력 등 모든 발전기가 대상이 될 수 있다. 그러나 발전단가가 저렴하고 가동을 중지하고 재가동하는데 과도한 시간이나 비용이 소요되는 원자력이나 석탄 발전소는 가동을 중지시키지 않으며, 되도록 최대 발전량  $P_i^{max}$ 로 가동시킨다. 그러나 중유, LNG 등 재가동 시간이 짧고, 발전단가가 비싼 경우에는 경제적인 발전을 위해 발전 용량을  $P_i = 0$  또는  $P_i^{min} \leq P_i \leq P_i^{max}$ 로 조절할 수 있다.

$n$ 개의 발전기는 각기 다른 발전량  $P_i^{min} \leq P_i \leq P_i^{max}$ 을 가지고 있으며, 발전비용은  $F(P_i) = a_i P_i^2 + b_i P_i + c_i$  또는  $a_i P_i^2 + b_i P_i + c_i + |e_i \times \sin(f_i \times (P_i^{min} - P_i))|$  중 하나를 적용한다.[2] 그러나 국내에서는 아직까지 2차함수를 적용하여 라그랑주 완화법(Lagrangian relaxation)으로 해를 구하고 있다.[3] 즉, 각기 다른 발전비용을 가진  $n$ 개의 발전기를 가동하여 요구량  $P_d$ 를 만족시키기 위해 각 발전기의 발전량  $P_i$ 를 얼마로 조절하였을 때 총 발전비용  $\Sigma F(P_i)$ 를 최소화시켜 가장 경제적으로 발전할 수 있는가의 최적화 문제가 경제급전이다.

경제급전 최적화와 관련된 기존 알고리즘들은 대부분 수학적 최적화 기법인 라그랑주 완화법, 경사법, 감소된 경사법이나 인공지능 기법인 유전자 알고리즘, 신경망, 퍼지, 군집 알고리즘 등 비결정적이며, 휴리스틱 알고리즘으로 초기치를 랜덤하게 설정하고 수십 회 이상 수행한 결과 최소값을 가진 결

과로 결정하여 동일한 해를 검증하기가 사실상 불가능하다. 또한, 다양한 휴리스틱 알고리즘을 적용함에도 불구하고, 대부분의 결과들은 거의 유사한 결과를 얻고 있으며, 획기적으로 해를 개선한 알고리즘을 찾기가 현실적으로 불가능하다.

본 논문에서는 알고리즘이 간단하면서도 결정론적 방법으로 기존 알고리즘들이 제시한 해를 획기적으로 개선한 교환 최적화(Swap Optimization, SO) 알고리즘을 제안한다. 2장에서는 동적 경제급전과 관련된 연구와 문제점을 고찰해 본다. 3장에서는 간단하면서도 획기적으로 해를 개선할 수 있는 알고리즘을 제안한다. 4장에서는 제안된 알고리즘을 적용하여 성능을 검증한다.

## II. 관련연구와 문제점

주어진 시간대에서 수급 균형을 맞추기 위해 총 발전량  $P_T$ 는 총 부하량  $P_L$ 과 동일해야 한다. 이는 식 (1)과 같다.

$$P_T = P_L \tag{1}$$

여기서 총 부하량  $P_L = P_d$ 로 총 요구량이 되며,  $P_d$ 는 엄밀히 말해 수요 예측량+공급 예비율이다. 발전단가가 저렴한 원자력 발전량은 항상 최대 발전을 수행한다고 가정하여 경제급전의 대상으로 포함시키지 않으며, 그림 1과 같이 발전단가가 비싸고 용량 제한이 가능한 화력(유류, LNG, 석탄)이 경제급전의 대상이 된다.

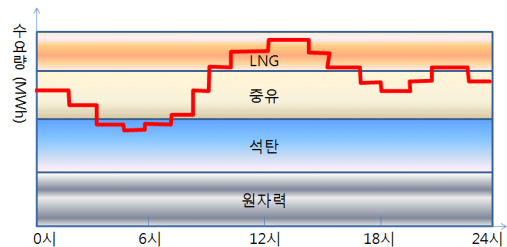


그림 1. 전형적인 시간대별 발전기 가동 현황  
Fig. 1. Conventional Generator Operation with 24-Hour

경제급전 대상 발전소의 발전기  $n$ 개가 존재하며, 각각의 발전비용이 차이가 발생한다고 가정한다. 이 경우 총 요구량  $\Sigma P_i^{min} \leq P_d \leq \Sigma P_i^{max}$ 를 충족시키기 위한 경제급전 문제의 목적함수는 경제급전 총 발전비용을 최소화시키는 것으로 식 (2)와 같다.[1,2]

$$\begin{aligned} \text{minimize } \Sigma F(P_i) &= \sum_{i=1}^n F(P_i) & (2) \\ \text{subject to } \Sigma P_i &= P_d \\ P_i^{min} &\leq P_i \leq P_i^{max} \end{aligned}$$

여기서  $F(P_i)$ 는 발전기  $i$ 의 발전비용 (\$/MWh)으로 식 (3)을 적용한다. 왜냐하면 발전기를 가동하면 기본적으로 최소 발전비용이 발생한다. 이후 발전 용량을 증가시키기 위해 밸브를 열면 진동 발생으로 인한 밸브효과 (valve-effect)로 발전비용  $F(P_i)$ 는 복잡한 비평활함수 (non-smooth function) 형태를 나타낸다.[1,2] 그러나 국내에서는 기울기  $2a_i P_i + b_i$ 를 쉽게 구하여 라그랑주 완화법을 적용하기 위해 간단히 식 (4)의 2차함수인 평활함수 (smooth function)를 적용하고 있다.

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

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

여기서  $P_i$ 는 발전기  $i$ 의 발전량 (KW)이며, 각 발전기의 출력 범위 제약조건은  $P_i^{\min} \leq P_i \leq P_i^{\max}$ 이다. 왜냐하면, 발전기가 가동되고 안전하게 연속적으로 운전이 가능한 최저 수준의 출력에 도달한 시점이  $P_i^{\min}$ 이다. 이는 석탄 발전기는 정격의 30~40%, 중유 발전기는 25~35% 정도이다.

표 1은 Attaviriyunap et al.[4]가 제시한 10-발전기의 동적 경제급전 예제 데이터이다.

표 1. 실험 데이터  
Table 1. Case Study Data

Data for the 10-Unit System							
$G_i$	$P_i^{\min}$ (MW/h)	$P_i^{\max}$ (MW/h)	$a$	$b$	$c$	$e$	$f$
1	150	470	0.00043	21.60	958.20	450	0.041
2	135	460	0.00063	21.05	1,313.60	600	0.036
3	73	340	0.00039	20.81	604.97	320	0.028
4	60	300	0.00070	23.90	471.60	260	0.052
5	73	243	0.00079	21.62	480.29	280	0.063
6	57	160	0.00056	17.87	601.75	310	0.048
7	20	130	0.00211	16.51	502.70	300	0.086
8	47	120	0.00480	23.23	639.40	340	0.082
9	20	80	0.10908	19.58	455.60	270	0.098
10	55	55	0.00951	22.54	692.40	380	0.094
Sum:	690	2,358					

Load Demand for 24 Hours

Hour	Load (MW)	Hour	Load (MW)	Hour	Load (MW)
1	1,036	9	1,924	17	1,480
2	1,110	10	2,072	18	1,628
3	1,258	11	2,146	19	1,776
4	1,406	12	2,220	20	2,072
5	1,480	13	2,072	21	1,924
6	1,628	14	1,924	22	1,628
7	1,702	15	1,776	23	1,332
8	1,776	16	1,554	24	1,184

표 1의 데이터들에 대해 경제급전 알고리즘을 연구한 기존의 연구 결과는 표 2에 제시하였다.[2,4-21] 표에서 다양한 휴리스틱 방법들을 적용한 결과 얻은 해 (발전비용)를 살펴보면 SQP[4]의 \$1,051.163에서 ICA[21]의 \$1,018,467까지 절감한 결과를 알 수 있다. 대부분의 논문들은 이전 연구 결과에 비해 단지 약간의 비용을 절감하면서 새로운 알고리즘들을 제시하고 있다.

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

Method	Cost(\$)	Iteration
Sequential Quadratic Programming (SQP) [4]	1,051,163.000	50,000
Evolutionary Programming (EP) [4]	1,048,638.000	50,000
Hybrid EP and SQP (EP-SQP) [4]	1,031,746.000	30,000
MDE [5]	1,031,612.000	-
HDE [6]	1,031,077.000	-
PSO-SQP [7]	1,030,773.000	30,000
AIS-SQP [8]	1,029,900.000	-
MHEP-SQP [9]	1,028,924.000	30,000
DGPSO [10]	1,028,835.000	30,000
PSO-SQP(C) [7]	1,027,334.000	30,000
IDE [11]	1,026,269.000	-
IPSO [12]	1,023,807.000	-
EPSO [13]	1,023,772.460	20,000
CMAES [14]	1,023,740.000	-
EPSO-GM [2]	<b>1,023,691.106</b>	<b>20,000</b>
CS-DE [15]	1,023,432.000	-
AIS [16]	1,021,980.000	-
ABC [17]	1,021,576.000	-
DE [18]	1,019,786.000	-
CDE [19]	1,019,123.000	-
HHS [20]	1,019,091.000	-
ICA [21]	<b>1,018,467.494</b>	<b>600</b>

### III. 교환 최적화 알고리즘

본 장에서는 식 (5)를 만족시키는 교환 최적화 (Swap Optimization, SO) 동적 경제급전 알고리즘을 제안한다.

$$\text{minimize } \Sigma F(P_i) = \sum_{i=1}^n F(P_i) \quad (5)$$

such that

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

subject to  $\Sigma P_i = P_d$

$$P_i^{\min} \leq P_i \leq P_i^{\max} \text{ (원자력, 유연탄, 국내탄)}$$

$$P_i = 0 \text{ or } P_i^{\min} \leq P_i \leq P_i^{\max} \text{ (얼북함, LNG, 수력, 중유)}$$

발전량을 2가지로 분류한 기준은 다음 근거에 기반하고 있다. 전력거래소의 전력통계정보시스템에 따르면 2011년 10월 현재 원자력(3.61원/KWh), 유연탄(46.87원), 무연탄 (69.82원),

LNG(122.86), 중유(215.88원)순이다. 한편의 2008년 12월 전력 계통 운영실적 자료에 따르면 원자력과 석탄은 기저부하를, 중유는 중간부하를, LNG는 기동 및 출력 조절이 용이하여 주 파수조절용으로, 수력은 용량이 적어 첨두부하를 담당하도록 운영되고 있다. 원자력과 석탄 발전기는 발전단가가 저렴하고, 재가동 시간이 과다하게 소요되어  $P_i^{min} \leq P_i \leq P_i^{max}$  를 적용해야 하나 LNG나 중유 등은 가동중지와 재가동이 비교적 쉽고, 발전단가가 비싸  $P_i = 0$ 도 가능하기 때문이다. 또한, 한편의 수력 원자력을 제외한 5개 자회사들은 모두 석탄화력, 유류와 가스화력, 내연 및 복합 발전소들을 모두 갖추고 있기 때문에 경제급전을 하기 위해서는 원료별 발전단가가 크게 영향을 미치기 때문이다.

제안 방법은 2단계로 수행된다. 첫 번째 단계는  $\Sigma P_i = P_d$  로 균형을 맞추는 단계로, 이 단계에서는  $P_i \leftarrow P_i^{max} - 1$ 에 대한  $r = F(P_i^{max})/P_i^{max}$ 에 의해 발전단가가 비싼 발전기는 발전을 중지시키는 개념을 적용하여 감소시키고, 잉여 발전량을  $P_i$ 와  $P_i - 1$ 간의 발전비용 차이가 최대인  $\max[F(P_i) - F(P_i - 1)]$ 에 대해  $P_i \leftarrow P_i - 1$ 로 감소시켜 최종적으로  $\Sigma P_i = P_d$ 의 균형을 맞춘다. 두 번째 단계는  $P_i \pm \Delta$ 의 발전비용에 대해  $\max[F(P_i) - F(P_i - \Delta)] > \min[F(P_j + \Delta) - F(P_j)]$ ,  $i \neq j$ 이면  $P_i = P_i - \Delta$ ,  $P_j = P_j + \Delta$ 로 발전량을 교환하는 방법으로 해를 개선한다. 제안된 알고리즘은 다음과 같이 수행된다.

Step 1. 각 발전기의 발전량  $P_i = P_i^{max} - 1$ 로 설정한다. 총 발전량에서 요구량을 뺀 값 ( $\Sigma P - P_d$ )에 대해 각 발전기의 발전단가  $r = F(P_i)/P_i$ 가 최대인 발전기는  $i$ 에 대해  $(\Sigma P - P_i) \geq P_d$  조건을 만족하면 가동을 중지 ( $P_i = 0$ )시킨다.

```

for i = 1 to n do /* 비가동 발전기 결정
     $P_i \leftarrow P_i^{max} - 1$ 로 설정.
     $F(P_i)$ 와  $r = F(P_i)/P_i$  계산.
while  $\Sigma P > P_d$  do
    max r 발전기 i에 대해
    if  $(\Sigma P - P_i) > P_d$  then  $P_i \leftarrow 0$ 
    
```

Step 2.  $\Sigma P - P_d > 0$ 인 값에 대해  $P_i \leftarrow P_i - 1$ 로 설정한 발전비용  $F(P_i)$ 와  $P_i - 1$ 의 발전비용  $F(P_i - 1)$ 의 비용 차이가 최대인 값  $\max[F(P_i) - F(P_i - 1)]$ 인 발전기에 대해  $\Sigma P - P_d = 0$ 일 때까지  $P_i \leftarrow P_i - 1$ 로 설정한다.

```

while  $\Sigma P_i = P_d$  do
     $P_{i(-1)} \leftarrow P_i - 1$ ,  $F(P_i - 1)$  계산
    if  $\max[F(P_i) - F(P_i - 1)]$  and
        ( $P_i > P_i^{min}$ ) then
         $P_i \leftarrow P_i - 1$ ,  $F(P_i) \leftarrow F(P_i - 1)$ ,
         $\Sigma P_i \leftarrow \Sigma P_i - 1$ 
    else if  $P_i = P_i^{min}$  then
        if  $(\Sigma P_i - P_i^{min}) < P_d$  then
             $P_i \leftarrow P_i^{min}$ 
        else  $P_i \leftarrow 0$ ,  $\Sigma P_i \leftarrow \Sigma P_i - P_i^{min}$ .
    
```

Step 3. 각 발전기의 발전량  $P_i$ 의  $\pm \Delta$  증감인  $P_i \pm \Delta$ 에 대해  $P_i = P_i - \Delta$ 의 최대 비용감소 발전기  $G_i$ 와  $P_j = P_j + \Delta$ 의 최소 비용 증가 발전기  $G_j$ 에 대해  $\max[F(P_i) - F(P_i - \Delta)] \leq \min[F(P_j + \Delta) - F(P_j)]$ ,  $i \neq j$ 일 때까지  $P_i = P_i - \Delta$ ,  $P_j = P_j + \Delta$ 로 교환한다. 여기서  $\Delta$ 는 1.0, 0.1, 0.01, 0.001로 4회에 걸쳐 수행되면서 해를 개선한다.

```

 $\Delta = 1.0, 0.1, 0.01, 0.001$ 에 대해 수행
while  $\max[F(P_i) - F(P_i - \Delta)] >$ 
 $\min[F(P_j + \Delta) - F(P_j)]$ ,  $i \neq j$  do
     $P_{i(-\Delta)} \leftarrow P_i - \Delta$ ,  $P_{i(+\Delta)} \leftarrow P_i + \Delta$ , ( $P_i < P_i^{max}$ )
     $F(P_{i(-\Delta)})$ ,  $F(P_{i(+\Delta)})$  계산
    if  $\max[F(P_i) - F(P_{i(-\Delta)})] > \min[F(P_{j(+\Delta)}) - F(P_j)]$ ,  $i \neq j$  then
         $P_i \leftarrow P_i - \Delta$ ,  $F(P_i) \leftarrow F(P_{i(-\Delta)})$ ,
         $P_j \leftarrow P_j + \Delta$ ,  $F(P_j) \leftarrow F(P_{j(+\Delta)})$ 
    else Exit
    
```

기존 알고리즘은  $P_i^{min} \leq P_i \leq P_i^{max}$  제약조건을 만족시키나 제안된 알고리즘은  $P_i = 0$ ,  $P_i^{min} \leq P_i \leq P_i^{max}$ 의 제약조건을 설정하였다. 이는 발전 단가가 비싸고 정지와 재가동 시간이 비교적 짧은 중유와 LNG 등 발전기는 가동을 중지시킬 수 있음을 의미한다. 또한, 발전기 종류, 최소 가동시간과 안전하게 연속적으로 운전이 가능한 최저수준의 출력  $P_i^{min}$ 까지 도달하는 가동비용도 실제 데이터가 없어 고려하지 않았다.

제안된 SO 알고리즘의 Step 1에서  $P_i = 0$ 로 설정하는 기준은 다음과 같이 6가지의 방법들 중에서 선택할 수 있으나  $r_2 = F(P_i^{max} - 1)/(P_i^{max} - 1)$ 가 가장 효율적인 기준임을 실험을 통해 얻었다.

$$r_1 = F(P_i^{max})/P_i^{max}$$

$$r_2 = F(P_i^{max} - 1)/(P_i^{max} - 1)$$

$$r_3 = [F(P_i^{max}) - c_i]/P_i^{max}$$

$$r_4 = [F(P_i^{\max}) - (a_i + b_i + c_i)] / (P_i^{\max} - 1)$$

$$r_5 = 2a_i P_i + b_i$$

$$r_6 = [F(P_i^{\max}) - F(P_i^{\min})] / (P_i^{\max} - P_i^{\min})$$

제안된 알고리즘은 기존의 휴리스틱 알고리즘과 비교시 다음과 같은 차이점이 있다.

- (1) 휴리스틱 알고리즘은 발전비용이 과다 소요되는 발전기가 존재하더라도 모든 발전기를 가동시키는 것으로 해를 얻는다. 반면에, SO 알고리즘은 고비용 발전기는 가동을 중단시키고, 저비용 발전기는  $P_i^{\max}$ 로 발전시킬 수 있다.
- (2) 휴리스틱 알고리즘들은 대부분 비결정적이며, 초기치가 다른 다수의 시행횟수에 대해 최소값을 선택하며 동일한 결과를 얻기가 불가능하다. 반면에 SO 알고리즘은 결정론적 알고리즘으로 항상 동일한 결과를 한 번에 얻는 장점이 있다.
- (3) 제안된 SO 알고리즘은 발전기 대수  $n$ 이 크거나 발전비용함수가 비평활함수로 복잡하더라도 이와 무관하게 간단히 수행될 수 있는 장점이 있다.

### IV. 적용 결과 및 분석

제안된 알고리즘을 표 1의 데이터에 대해 수행한 결과는 표 3에 SO로 제시되어 있다. 표 3에서  $G_5$  발전기는 01h에서 정지한 후 다시 05h에서 정지하는 형태를 나타내고 있어 정지 후 재가동하는 것 보다는 가능한 최저출력으로 가동하는 것이 보다 효율적일 수 있다. 이러한 경우에 대해 알고리즘을 재 수행한 결과는 표 4에 SO-1으로 제시되어 있다.

제안된 SO 알고리즘을 Hour 1에 대해 수행하여 해를 구한 결과는 표 5에 제시되어 있으며, 24 Hour 동안 수행한 결과와 표 2에 제시된 기존의 최적의 연구 결과인 ICA[21]와 발전비용을 비교한 결과는 표 6과 같다. 표 6으로부터 제안된 SO나 SO-1 알고리즘은 기존의 최적의 성능을 가진 ICA에 비해 비용을 크게 감소시켰다.

SO와 ICA의 발전비용을 시간대별로 비교한 결과는 그림 2에 제시되어 있으며, 24시간 동안의 ICA와 SO의 총 발전비용은 표 7에 제시되어 있다. 제안된 SO와 SO-1 알고리즘은 단 1회 수행으로 해를 구할 수 있으며, 기존의 휴리스틱 알고리즘들에 비해 비용도 크게 감소시켰음을 알 수 있다.

표 3. 교환 최적화 알고리즘 수행 결과  
Table 3. The Results of Swap Optimization Algorithm

	$P_i^{\min}$	$P_i^{\max}$	Hour												
			01h	02h	03h	04h	05h	06h	07h	08h	09h	10h	11h	12h	13h
G1	150	470	456.497	456.496	456.000	456.497	456.497	456.000	456.497	457.000	456.000	457.000	457.000	456.496	457.000
G2	135	460	0.000	0.000	0.000	396.800	460.000	459.000	460.000	459.000	459.000	459.000	459.000	460.000	459.000
G3	73	340	297.399	178.866	297.000	78.062	311.462	240.000	310.862	339.000	237.000	339.000	301.000	297.400	339.000
G4	60	300	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	299.000	299.000	299.000	300.000	299.000
G5	73	243	0.000	222.598	222.000	222.600	0.000	222.000	222.600	233.000	222.000	230.000	223.000	222.600	230.000
G6	57	160	152.514	122.450	154.000	122.450	122.450	122.000	122.450	159.000	122.000	159.000	159.000	153.924	159.000
G7	20	130	129.590	129.590	129.000	129.591	129.591	129.000	129.591	129.000	129.000	129.000	129.000	129.580	129.000
G8	47	120	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	119.000	120.000	0.000
G9	20	80	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	80.000	0.000
G10	55	55	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\Sigma P_i = P_d$			1,036	1,110	1,258	1,406	1,480	1,628	1,702	1,776	1,924	2,072	2,146	2,220	2,072

	Hour											
	14h	15h	16h	17h	18h	19h	20h	21h	22h	23h	24h	
G1	456.000	457.000	456.497	456.497	456.497	456.497	456.497	456.497	456.497	459.409	456.496	
G2	459.000	459.000	460.000	460.000	460.000	460.000	460.000	460.000	460.000	0.000	0.000	
G3	237.000	339.000	162.864	88.862	236.862	340.000	340.000	232.862	236.862	340.000	252.864	
G4	299.000	0.000	0.000	0.000	0.000	0.000	300.000	300.000	0.000	0.000	0.000	
G5	222.000	233.000	222.599	222.600	222.600	229.912	225.912	222.600	222.600	243.000	222.600	
G6	122.000	159.000	122.450	122.450	122.450	160.000	160.000	122.450	122.450	160.000	122.450	
G7	129.000	129.000	129.590	129.591	129.591	129.591	129.591	129.591	129.591	129.591	129.590	
G8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
G9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
G10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
$\Sigma P_i = P_d$	1,924	1,776	1,554	1,480	1,628	1,776	2,072	1,924	1,628	1,332	1,184	

표 4. G5 발전기 가동시 교환 최적화 알고리즘 수행 결과  
Table 4. The Results of Swap Optimization Algorithm with G5 Generator Operating

	$P_i^{\min}$	$P_i^{\max}$	Hour												
			01h	02h	03h	04h	05h	06h	07h	08h	09h	10h	11h	12h	13h
G1	150	470	456.497	456.496	456.000	456.497	456.497	456.000	456.497	457.000	456.000	457.000	456.496	457.000	
G2	135	460	0.000	0.000	0.000	396.800	460.000	459.000	460.000	459.000	459.000	459.000	460.000	459.000	
G3	73	340	104.862	178.866	297.000	78.062	88.862	240.000	310.862	339.000	237.000	339.000	301.000	297.400	
G4	60	300	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	299.000	299.000	299.000	299.000	
G5	73	243	222.600	222.598	222.000	222.600	222.600	222.000	222.600	233.000	222.000	230.000	223.000	222.600	
G6	57	160	122.450	122.450	154.000	122.450	122.450	122.000	122.450	159.000	122.000	159.000	159.000	153.924	
G7	20	130	129.591	129.590	129.000	129.591	129.591	129.000	129.591	129.000	129.000	129.000	129.000	129.580	
G8	47	120	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	119.000	120.000	
G9	20	80	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	80.000	
G10	55	55	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	
$\Sigma P_i = P_d$			1,036	1,110	1,258	1,406	1,480	1,628	1,702	1,776	1,924	2,072	2,146	2,220	

	Hour										
	14h	15h	16h	17h	18h	19h	20h	21h	22h	23h	24h
G1	456.000	457.000	456.497	456.497	456.497	456.497	456.497	456.497	456.497	456.497	459.409
G2	459.000	459.000	460.000	460.000	460.000	460.000	460.000	460.000	460.000	460.000	0.000
G3	237.000	339.000	162.864	88.862	236.862	340.000	340.000	232.862	236.862	340.000	252.864
G4	299.000	0.000	0.000	0.000	0.000	0.000	300.000	300.000	0.000	0.000	0.000
G5	222.000	233.000	222.599	222.600	222.600	229.912	225.912	222.600	222.600	243.000	222.600
G6	122.000	159.000	122.450	122.450	122.450	160.000	160.000	122.450	122.450	160.000	122.450
G7	129.000	129.000	129.590	129.591	129.591	129.591	129.591	129.591	129.591	129.591	129.590
G8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
G9	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
G10	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
$\Sigma P_i = P_d$	1,924	1,776	1,554	1,480	1,628	1,776	2,072	1,924	1,628	1,332	1,184

표 5. Hour 1에 대한 SO 수행 과정  
Table 5. Swap Optimization Process for Hour 1

단계	$\Sigma P_i$	$\Sigma F(P_i)$
초기치	2,348	59,547.647
Step 1	2,294	57,574.688 ( $G_{10}$ )
	2,215	54,761.774 ( $G_8$ )
	2,096	51,164.178 ( $G_5$ )
	1,797	43,448.033 ( $G_1$ )
	1,338	31,868.875 ( $G_2$ )
1,096	25,847.119 ( $G_3$ )	
Step 2	1,084	25,317.712
	1,042	24,196.864
	1,041	24,174.755
	1,036	24,089.247
Step 3	$\Delta = \pm 1.0$	1,036
	$\Delta = \pm 0.1$	1,036
	$\Delta = \pm 0.01$	1,036
	$\Delta = \pm 0.001$	1,036

본 논문은 다양한 원자재 (원자력, 무연탄, 유연탄, LNG, 중유, 등)로 발전을 하는 발전기들에 대해 주어진 전력 요구량 (수요량)을 발전하기 위해 발전 원가를 얼마나 절감할 수 있는가에 대한 경제급전에 초점을 맞추었다. 이는 공급량 이 수요량을 충족시키지 못하여 발생하는 흑서기와 흑한기의 전력난과는 무관하다. 또한, 본 연구는 발전기의 단위 시간

표 6. 알고리즘 비교  
Table 6. Comparison of Algorithms

Hour	Load	ICA (\$)	SO (\$)	비용 절감	SO-1 (\$)	비용절감
1	1,036	28,238.754	24,061.840	4,176.914	24,756.755	3,481.999
2	1,110	29,828.077	26,112.278	3,715.799	26,112.278	3,715.799
3	1,258	33,347.045	29,381.792	3,965.253	29,381.792	3,965.253
4	1,406	36,296.715	33,758.693	2,538.022	33,758.693	2,538.022
5	1,480	37,991.334	35,218.371	2,772.963	35,898.089	2,093.245
6	1,628	41,387.159	39,176.806	2,210.353	39,176.806	2,210.353
7	1,702	42,844.529	40,532.808	2,311.721	40,532.808	2,311.721
8	1,776	44,600.484	42,590.137	2,010.347	42,590.137	2,010.347
9	1,924	47,885.318	46,813.439	1,071.879	46,813.439	1,071.879
10	2,072	51,887.342	50,162.676	1,724.666	50,162.676	1,724.666
11	2,146	53,788.277	52,431.771	1,356.506	52,431.771	1,356.506
12	2,220	55,605.118	55,157.253	447.865	55,157.253	447.865
13	2,072	51,357.359	50,162.676	1,194.683	50,162.676	1,194.683
14	1,924	47,818.061	46,813.439	1,004.622	46,813.439	1,004.622
15	1,776	44,649.659	42,590.037	2,059.622	42,590.037	2,059.622
16	1,554	39,816.706	37,495.137	2,321.569	37,495.137	2,321.569
17	1,480	37,983.869	35,898.089	2,085.780	35,898.089	2,085.780
18	1,628	41,294.355	39,176.806	2,117.549	39,176.806	2,117.549
19	1,776	44,374.060	42,590.037	1,784.023	42,590.037	1,784.023
20	2,072	51,862.515	50,162.676	1,699.839	50,162.676	1,699.839
21	1,924	47,915.540	46,813.439	1,102.101	46,813.439	1,102.101
22	1,628	41,280.418	39,376.806	1,903.612	39,376.806	1,903.612
23	1,332	34,952.455	31,552.620	3,399.835	31,552.620	3,399.835
24	1,184	31,462.345	27,911.574	3,550.771	27,911.574	3,550.771
Total		1,018,467,494	955,941,200	52,526,294	967,315,833	51,151,661

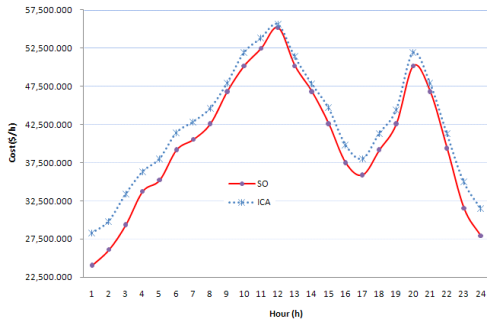


그림 2. 24시간 발전비용 경향 비교  
Fig. 2. Comparison of Cost Trend During 24-Hours

표 7. 최적화 결과 비교  
Table 7. Comparison of the Optimization Results

Method	Cost(\$)	Iteration
ICA (21)	1,018,467.494	600
SO (G5 가동중지)	<b>966,620.918</b>	<b>1</b>
SO-1 (G5 가동)	<b>967,315.833</b>	<b>1</b>

당 증가와 감소시킬 수 있는 발전량의 한계인 증감발출 (ramp rate)은 고려하지 않았으며, 비효율적인 발전기는 발전을 정지시키는 개념을 적용하였다. 만약, 경제급전이 모든 발전기는 최소 출력을 유지하는 개념을 적용하고, 증감발출 또한 고려해야 한다면 제안된 알고리즘은 산업현장에서 즉시 적용하는데 한계가 있을 수 있다.

### V. 결론 및 향후 연구과제

본 논문은 24시간 동안 가장 경제적인 비용으로 전력을 생산할 수 있는 동적 경제급전의 최적화 문제의 해를 구하는 알고리즘을 제안하였다. 기존 알고리즘들은 휴리스틱 방법들을 적용하여 다수의 시행 결과 얻은 해 들 중에서 최소값을 선택하는 방법을 적용하거나 2차 함수의 기울기를 이용하는 수학적 방법을 적용하고 있다. 반면에, 본 논문에서 제안한 교환 알고리즘은 항상 동일한 결과를 얻을 수 있으며, 알고리즘도 간단함을 알 수 있다.

제안된 SO 알고리즘은  $\Sigma P_i^{min} \leq P_d \leq \Sigma P_i^{max}$ 에 대해  $\Sigma P_i = P_d$ 가 되도록 균형을 맞추기 위해 발전비용이 과다 소요되는 비효율 발전기를 가동중지 시키는 개념을 적용하고, 나머지 여유분에 대해  $P_i - (P_i - 1)$ 의 1MW에 대한 발전비용이 최대로 감소되는 발전기의 발전용량을 감소시키는 방법을 적용하였다. 일단,  $\Sigma P_i = P_d$ 가 되면  $P_i - \Delta$ 와  $P_i + \Delta$ 에 대해 발전비용 최대 감소 발전기  $i$ 와 발전비용 최소 증가 발

전기  $j$ 에 대해  $P_i \leftarrow P_i - \Delta$ ,  $P_j \leftarrow P_j + \Delta$ 로 교환하는 방법을 적용하였다. 이와 같은 방법을 적용한 결과 기존의 휴리스틱 알고리즘의 24시간 동적 경제급전의 비용을 크게 감소시켰다.

추후 발전기의 출력 제약조건과 증감발출을 고려하여 산업현장에서 즉시 활용할 수 있는 알고리즘에 대한 연구를 수행할 예정이다.

### 참고문헌

- [1] 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.
- [2] P. Sriyangong, "A Hybrid Particle Swarm Optimization Solution to Ramping Rate Constrained Dynamic Economic Dispatch," World Academy of Science, Engineering and Technology, Vol. 47, pp. 374-379, 2008.
- [3] J. S. Kim, "Year 2009 Education of KPX Power Trading, System Operational Principle Division, VI. Operational Generation Plan," Korea Power Exchange Central Control Center, 2009.
- [4] P. Attaviriyapap, 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.
- [5] F. Li, R. Morgan, and D. Williams, "A Hybrid Genetic Approaches to Ramping Rate Constrained Dynamic Economic Dispatch," Electric Power System Research, Vol. 43, pp. 97-103, 1997.
- [6] X. Yuan, L. Wang, Y. Yuan, Y. Zhang, B. Cao, and B. Yang, "A Modified Differential Evolution Approach for Dynamic Economic Dispatch with Valve Point Effects," Energy Conversion and Management, Vol. 49, pp. 3447-3453, 2008.
- [7] T. A. A. Victoire and A. E. Jeyakumar, "Reserve Constrained Dynamic Dispatch of Units With Valve-Point Effects," IEEE Trans. Power System, Vol. 20, pp. 1273-1282, 2005.
- [8] S. Hemamalini and S. Simon, "Maclaurin Series-based Lagrangian Method for Economic Dispatch with Valve-point Effect," IET General Transmission Distribution, Vol. 3, No. 9, pp. 859-871, 2009.
- [9] T. A. A. Victoire and A. E. Jeyakumar, "A Modified Hybrid

- EP-SQP Approach for Dynamic Dispatch with Valve-point Effect," *International Journal of Electrical Power & Energy Systems*, Vol. 27, pp. 594-601, 2005.
- [10] T. A. A. Victoire and A. E. Jeyakumar, "Deterministically guided PSO for Dynamic Dispatch Considering Valve-point Effect," *Electric Power Systems Research*, Vol. 73, pp. 313-322, 2005.
- [11] Y. Lu, J. Zhou, H. Qin, Y. Wang, and Y. Zhang, "Chaotic Differential Evolution Methods for Dynamic Economic Dispatch with Valve-point Effects," *Engineering Applications of Artificial Intelligence*, Vol. 24, pp. 378-387, 2011.
- [12] H. Dakuo, G. Dong, F. Wang, and Z. Mao, "Optimization of Dynamic Economic Dispatch with Valve-point Effect using Chaotic Sequence based Differential Evolution Algorithms," *Energy Conversion and Management*, Vol. 52, pp. 1026-1032, 2011.
- [13] P. Sriyanyong, "An Enhanced Particle Swarm Optimization for Dynamic Economic Dispatch Problem Considering Valve-Point Loading," In *Proceeding of the Fourth IASTED International Conference on Power and Energy Systems (AsiaPES 2008)*, pp. 167-172, 2008.
- [14] N. Nomana and H. Iba, "Differential Evolution for Economic Load Dispatch Problems," *Electric Power System Research*, vol. 78, pp. 1322-1331, 2008.
- [15] R. Balamurugan and S. Subramanian, "Differential Evolution-based Dynamic Economic Dispatch of Generating Units with Valve-point Effects," *Electric Power Components and Systems*, Vol. 36, pp. 828-843, 2008.
- [16] S. Hemamalini and S. P. Simon, "Dynamic Economic Dispatch using Maclaurin Series based Lagrangian Method," *Energy Conversion and Management*, Vol. 51, pp. 2212-2219, 2010.
- [17] Hindi, K. S., and M. R. Ab Ghani, "Dynamic Economic Dispatch for Large Scale Power Systems: A Lagrangian Relaxation Approach," *Electric Power and Energy Systems*, Vol. 13, pp. 51-56, 1991.
- [18] R. Balamurugan and S. Subramanian, "An Improved Differential Evolution based Dynamic Economic Dispatch with Non-smooth Fuel Cost Function," *Journal of Electrical Systems*, Vol. 3, pp. 151-161, 2007.
- [19] J. C. Lee, W. M. Lin, G. C. Liao, and T. P. Tsao, "Quantum Genetic Algorithm for Dynamic Economic Dispatch with Valve-point Effects and Including Wind Power System," *Electrical Power and Energy Systems*, Vol. 33, pp. 189-197, 2011.
- [20] X. S. Han, H. B. Gooi, and D. S. Kirschen, "Dynamic Economic Dispatch: Feasible and Optimal Solutions," *IEEE Trans. Power System*, Vol. 16, No. 1, pp. 22-28, 2001.
- [21] A. Rabiei, A. Soroudi, and B. Mohammadi, "Imperialist Competition Algorithm for Solving Non-convex Dynamic Economic Power Dispatch," *26th International Power System Conference*, Tehran-Iran, pp. 1-8, 2011.

### 저자 소개



이 상 운(Sang-Un, Lee)

1983년 ~ 1987년 :

한국항공대학교 항공전자공학과 (학사)

1995년 ~ 1997년 :

경상대학교 컴퓨터과학과 (석사)

1998년 ~ 2001년 :

경상대학교 컴퓨터과학과 (박사)

2003.3 ~ 현재 :

강릉원주대학교 멀티미디어공학과 부교수

관심분야 : 소프트웨어 프로젝트 관리,

소프트웨어 개발 방법론,

소프트웨어 신뢰성, 신경망,

뉴로-퍼지, 그래프 알고리즘

e-mail : [sulee@gwmu.ac.kr](mailto:sulee@gwmu.ac.kr)