Economic Dispatch with Linear Decreasing and Staircase Incremental Cost Functions by Micro Genetic Algorithms

W. Ongsakul and J. Tippayachai
Department of Electrical Engineering
Sririndhorn International Institute of Technology
Thammasat University, Pathumthani 12121, Thailand

ABSTRACT- In this paper, a Micro Genetic Algorithm (MGA) to solving the economic dispatch (ED) problem with linear decreasing and staircase incremental cost (IC) functions for combined cycle (CC) units is proposed. To demonstrate its advantage, the proposed MGA methods with two different encoding schemes are tested and compared to the unconstrained Brute Force (BF), Simple Genetic algorithm (SGA), Merit Order Loading (MOL), and equal-lambda based Newton methods on five combined cycle units. As the ramp rate constraints are taken in account, the proposed MGA solutions are ensured to be feasible. Test results indicate that the solutions are found to be close to the optimal solution of the unconstrained BF method and its total fuel costs are lower than those of the SGA, MOL, and Newton methods.

KEYWORDS: Micro Genetic Algorithm (MGA), Economic Dispatch (ED), Brute Force (BF), Simple Genetic Algorithm (SGA), Merit Order Loading (MOL), Newton Method.

1. Introduction

Economic dispatch (ED) is used to determine the optimal schedule of on-line generating outputs so as to meet the load demand at the minimum operating cost. The existing ED program, a standard function of the Energy Management System (EMS), National Control Center (NCC), Electricity Generating Authority of Thailand (EGAT), is applicable only for monotonically increasing incremental cost (IC) functions which is limited in flexibility. Neither linear decreasing IC functions nor staircase IC functions can be handled by the program [1].

At present, combined cycle (CC) units of EGAT system are always scheduled to serve the base load. However, during light load periods when there are no other generating units that can further reduce their outputs, there is a need to vary the CC units in an economical and smooth manner instead of a zero-one discrete basis. Therefore, if the linear decreasing IC and decreasing staircase IC functions (or non-convex and non-smooth input-output functions) of the CC units are included in the database, the conventional ED program based on the equal lambda methodology cannot determine the optimal solution [2].

Ongsakul [1,2] proposed the Merit Order Loading (MOL) method based on the unit lambda values at
the highest operating outputs to solving the ED problem with linear decreasing IC and decreasing staircase IC of CC units. It was shown that the proposed MOL solution was close to the optimal solution and the real time implementation was valid on the existing EMS of EGAT without violating the ramp rate constraints. However, the monthly fuel costs of CC units of EGAT are in the order of several billion Thai baht, a few percent improvement of the existing ED program for CC units can lead to substantial fuel cost savings. Accordingly, this paper will investigate a Micro Genetic algorithm (MGA) to solving the ED with linear decreasing IC and decreasing staircase IC which are non-monotonically increasing functions.

Bakirtzis et al [3] proposed the Genetic algorithm to solving ED problem without convexity restrictions on the generator cost functions with valve point loading. The proposed method outperformed the dynamic programming in terms of computing times for the generating units ranging from 9 to 72. But the success rate to the optimal solution was still only 40% for the 72 generating unit system.

Wong et al [4] proposed the Genetic/Simulated-annealing approaches to ED problem. The algorithm was developed based on the combination of the incremental genetic-algorithm approach and the simulated-annealing technique. It was shown to be computationally faster than the earlier simulated annealing based method on a 13 generator practical system.

Sheble et al [5,6] proposed the Genetic algorithm (GA) and refined Genetic algorithm (RGA) to solving ED with valve point loading. Several techniques were explored to enhance the efficiency and accuracy such as mutation prediction, elitism, interval approximation and penalty factors. Test results however were shown on a very small size three-bus system.

Chen et al [7] proposed the large-scale ED by genetic algorithm with a normalized lambda encoding method. The network losses, ramp rate constraints, and prohibited zone were also taken into account. The proposed GA was tested on the system sizes ranging from 5 to 40 units. Even though this encoding scheme is practical for large scale implementation, its application is restricted to monotonically increasing IC functions only.

In this paper, a Micro Genetic algorithm (MGA) to solving the ED with linear decreasing IC and decreasing staircase IC functions of CC units is proposed. The ramp rate constraints are also taken into account to insure the feasibility of solutions. Two different encoding methods including concatenated and embedded methods are investigated. The MGA is tested and compared to Simple Genetic algorithm (SGA), MOL, Newton and unconstrained BF on a five CC unit system.

Two types of cost functions used for CC generating units at EGAT can be summarized as follows [1].

Second order polynomial cost function for a CC unit: One combined cycle unit consists of a series of single-cycle gas turbines in conjunction with a heat-recovery steam generator (HRSG). In a closed cycle operation mode, a cost function is obtained by curve-fitting among at least three test points using the least squares method. In fact, the second order polynomial cost function for a CC unit is estimated by \( a + bP + cP^2 \), where the coefficient \( c < 0 \), and IC is a linear decreasing function \( (b+2cP) \). This phenomenon is explained by the characteristics of a CC unit in such a way that the higher the output is, the better the efficiency will be.

Piecewise linear cost functions for a CC unit: To obtain extreme accuracy, instead of using only one quadratic equation to represent the cost function of a CC unit, several piecewise linear functions are employed to represent cost functions for each mode of the closed cycle operation. The piecewise linear cost functions and their decreasing staircase IC functions of one of the closed cycle modes are as follows.

\[
C_i(P) = \begin{cases} 
  a_{i1} + b_{i1}P_i, & P_{min} \leq P_i \leq P_{lim}, \\
  a_{i2} + b_{i2}P_i, & P_{lim} \leq P_i \leq P_{max} 
\end{cases} (1a)
\]

\[
IC_i(P) = \begin{cases} 
  b_{i1}, & P_{min} \leq P_i \leq P_{lim}, \\
  b_{i2}, & P_{lim} \leq P_i \leq P_{max} 
\end{cases} (2a)
\]

where \( b_{i1} > b_{i2} > 0 \).

The organization of the paper is as follows. The ED problem formulation is introduced in Section 2. Section 3 describes the SGA and MGA. The experimental results on a five unit system are given in Section 4. Conclusion is given in the last section.

2. Economic Dispatch Formulation

2.1 Basic Economic Dispatch Formulation

The conventional ED problem is to minimize the total cost function. The problem is formulated as:
Minimize $C_T = \sum_{i=1}^{N} C_i (P_i)$, subject to a power balance constraint:

$$\sum_{i=1}^{N} P_i = P_D + P_{\text{loss}}(P_1, ..., P_N),$$

and operating limit constraints:

$$P_{i\text{min}} \leq P_i \leq P_{i\text{max}}, \quad i = 1, ..., N,$$

where,

$C_T$ = total fuel cost (baht/hr),

$C_i (P_i)$ = cost of the $i$th generating unit (baht/hr),

$P_i$ = real power output of the $i$th generating unit (MW),

$P_D$ = total load demand (MW),

$P_{\text{loss}}(P_1, ..., P_N)$ = total transmission line loss (MW),

$N$ = total number of on-line units to be dispatched,

$P_{i\text{min}}$ = minimum power output of the $i$th unit (MW),

$P_{i\text{max}}$ = maximum power output of the $i$th unit (MW).

### 2.2 Economic Dispatch Formulation with Ramp Rate Limits

Ramp rate of generating units are due to the fact that CC generating outputs can be not adjusted instantaneously. Therefore, to reflect the actual operating process, ED problem should include the ramp rate limits to ensure the feasibility of the solutions.

![Figure 1. Two Possible Situations of On-line $i$th Unit](image)

\[ P_i(t) - P_i(t-1) \leq U_{R_i} \quad \text{(a)} \]

\[ P_i(t-1) - P_i(t) \leq D_{R_i} \quad \text{(b)} \]

As shown in Figure 1, the inequality constraints of ramp rate limits are given as:

1. if the $i$th generation unit output increases (see Figure 1a)

2. if the $i$th generation unit output decreases (see Figure 1b)

Combining equations (3), (4), (5), (6) and (7), the constrained ED problem formulation becomes:

$$\text{Minimize } C_T = \sum_{i=1}^{N} C_i (P_i),$$

Subject to:

$$\sum_{i=1}^{N} P_i(t) = P_D + P_{\text{loss}}(P_1(t), ..., P_N(t)),$$

$$P_{i\text{low}}(t) \leq P_i(t) \leq P_{i\text{high}}(t), \quad i = 1, ..., N.$$

Where,

$L_{\text{low}}(i)$ = the possible lowest power output of the $i$th unit at time $t$ ($\text{Max}(P_{i\text{min}}, P_i(t-1) - D_{R_i})$),

$L_{\text{high}}(i)$ = the possible highest power output of the $i$th unit at time $t$ ($\text{Min}(P_{i\text{max}}, P_i(t-1) + U_{R_i})$),

In this paper, the total power loss is neglected, thus $P_{\text{loss}}(P_1(t), ..., P_N(t)) = 0$.

### 3. Genetic Algorithms

Genetic algorithm (GA) is essentially a searching method based on the concept of natural selection and natural genetics. GA searches on encoded bit strings (usually binary representations) called individuals rather than the real data points in solution space. GA has the ability to solve non-smooth, non-continuous, and non-differentiable cost functions which is not possible to obtain the optimal solution by a classical Lagrange method. GA uses the objective function to evaluate the performance, not its derivatives or auxiliary equations; and it has ability to exploit prior knowledge from previous solution guess to increase the performance of future solutions. Furthermore, GA exploits probability transition rules rather than deterministic rules [8].

#### 3.1 Simple Genetic Algorithm

In this paper, we shall use the Simple Genetic algorithm (SGA) developed by Goldberg [8] to solve the ED problem for comparison. The outputs of the $N$ generating units have to satisfy the power balance constraint, operating limit constraints, and ramp rate constraints. For arbitrary free unit outputs $P_i$, $P_{i\text{low}}(t) \leq P_i(t) \leq P_{i\text{high}}(t), i = 1, ..., N-1$, it is assumed that the $N$th reference unit power output is constrained by the power balance equation as:

$$P_N(t) = P_D(t) - \sum_{i=1}^{N-1} P_i(t)$$

### 3.1.1 Encoding and Decoding
In this paper, concatenated and embedded encoding methods are explored.

\[ P_{1,IJ}(t) = \text{Max}(100, 230 - 80) = 150 \text{ MW} \]
\[ P_{1,hi,h}(t) = \text{Min}(250, 230 + 50) = 250 \text{ MW} \]
\[ B_1 = (2^2*0) + (2^1*1) + (2^0*1) + (2^3*1) + (2^4*1) + (2^5*1) + (2^6*1) + (2^7*1) = 509. \]
\[ P_1(t) = 150 + [(509*(250 - 150)/(210 - 1))] = 199.76 \text{ MW}. \]

Therefore, each free unit is represented by 15 bits. The more the number of bits, the higher the resolution.

Unit 1  Unit 2  Unit 3
[aaaaaaabbbbbbbbccecccccc]  
\[ (a) \text{ Concatenated Encoding} \]

Unit 1  Unit 2  Unit 3
[abcabcabcabcabcabcabcabc]  
\[ (b) \text{ Embedded Encoding} \]

As shown in Figure 2, each unit output of \(N-1\) free units is encoded in a binary based string normalized over its operating range. The concatenated encoding method stacks each unit's normalized string in series with each other to constitute the string individual. Each unit's string structure is assigned by the same number of \(n\) bits. On the other hand, the embedded encoding method employs the same binary system and encoding as the concatenated one. The only difference is that the assigned bit structures of each unit string are embedded within each other throughout the individual. Each individual consists of a series of smaller string structures (3 instead of 10). For both methods, a string individual has \(n(N-J)\) bits.

For a concatenated encoding example, three generating unit power outputs are encoded in a 30 bit string individual as:

\[ 011111110110110110110100011011 \]

To obtain the actual generating power output of each unit for fitness function evaluation to be discussed, we need to decode each of 10 bit string to the decimal value by,

\[ P_i(t) = P_{i,\text{low}}(t) + [B_i*(P_{i,\text{high}}(t) - P_{i,\text{low}}(t))/(2^n - 1)] \]

Where,

\( P_i(t) \) = decimal integer value of converted binary string of the \(i\)th unit,
\( B_i \) = number of bits representing each unit output.

For example, the first generating unit power output, encoded by the first ten bits of the binary string individual, has \( UR_i \) and \( DR_i \) as 50 and 80 MW/hr respectively and suppose \( P_1(t-1) \) is 230 MW, \( P_{1,\text{min}} = 100, \) and \( P_{1,\text{max}} = 250. \) Then

\[ P_{1,\text{low}}(t) = \text{Max}(100, 230 - 80) = 150 \text{ MW}, \]
\[ P_{1,\text{high}}(t) = \text{Min}(250, 230 + 50) = 250 \text{ MW}, \]
\[ B_1 = (2^2*0) + (2^1*1) + (2^0*1) + (2^3*1) + (2^4*1) + (2^5*1) + (2^6*1) + (2^7*1) + (2^8*1) = 509. \]

Therefore, \( P_1(t) = 150 + [509*(250 - 150)/(210 - 1)] \)
\[ = 199.76 \text{ MW}. \]

SGA randomly generates initial \( NP \) (a specified population size) string individuals. For each generation (or iteration), SGA performs four basic operations: fitness function evaluation, reproduction, crossover, and mutation.

3.1.2 Fitness Function Evaluation

The performance in finding optimum solution of SGA is mainly related to the highest single fitness value. The fitness function including the power balance constraint of the \(j\)th individual is:

\[ f_j = 0.5/C_{st_j} + 0.5/P_{\text{Pow}_j}, j=1,...,NP \] (11)

Where,

\[ P_{\text{Pow}_j} = 1 + (\Sigma_{i=1}^{N} P_i(t)-P_{d}(t))^2/P_{d}(t), \]
\[ C_{st_j} = 1 + k*(C_{st}(P_1(t),...,P_N(t)) - C_{st_{\text{min}}})^2, \]
\[ P_i(t) = \text{real power output of the } i\text{th unit of the } j\text{th individual at time } t. \]
\[ C_{st_{\text{min}}} = \text{minimum total fuel cost at } P_{i,\text{min}} i=1,...,N \]
\( \text{(bhat)}, \)
\[ k = \text{a constant value.} \]

The fitness function evaluates a power balance difference and a total fuel cost difference. By experiment, \( k = 0.001 \) is used to scale down the total fuel cost difference squared component. Otherwise, the cost difference squared component will dominate the power balance difference squared over the total load demand component. After decoding the \( j\)th individual to \([P_1(t),...,P_N(t)]\) and substituting in Eq. (9),

if \( P_{d}(t) < P_{N,\text{low}}(t), P_{d}(t)=P_{N,\text{low}}(t), \) and
if \( P_{d}(t) > P_{N,\text{high}}(t), P_{d}(t)=P_{N,\text{high}}(t). \)

We use \([P_1(t),...,P_N(t)]\) to evaluate the fitness value in Eq. (11) even though the power balance may not be satisfied. By using Eq. (11), the fitness value is ensured to be in the range of 0 to 1.

3.1.3 Reproduction

For example, the first generating unit power output, encoded by the first ten bits of the binary string individual, has \( UR_i \) and \( DR_i \) as 50 and 80 MW/hr respectively and suppose \( P_1(t-1) \) is 230 MW, \( P_{1,\text{min}} = 100, \) and \( P_{1,\text{max}} = 250. \) Then

\[ P_{1,\text{low}}(t) = \text{Max}(100, 230 - 80) = 150 \text{ MW}, \]
\[ P_{1,\text{high}}(t) = \text{Min}(250, 230 + 50) = 250 \text{ MW}, \]
\[ B_1 = (2^2*0) + (2^1*1) + (2^0*1) + (2^3*1) + (2^4*1) + (2^5*1) + (2^6*1) + (2^7*1) + (2^8*1) = 509. \]

Therefore, \( P_1(t) = 150 + [509*(250 - 150)/(210 - 1)] \)
\[ = 199.76 \text{ MW}. \]

SGA randomly generates initial \( NP \) (a specified population size) string individuals. For each generation (or iteration), SGA performs four basic operations: fitness function evaluation, reproduction, crossover, and mutation.

The performance in finding optimum solution of SGA is mainly related to the highest single fitness value. The fitness function including the power balance constraint of the \(j\)th individual is:

\[ f_j = 0.5/C_{st_j} + 0.5/P_{\text{Pow}_j}, j=1,...,NP \] (11)

Where,

\[ P_{\text{Pow}_j} = 1 + (\Sigma_{i=1}^{N} P_i(t)-P_{d}(t))^2/P_{d}(t), \]
\[ C_{st_j} = 1 + k*(C_{st}(P_1(t),...,P_N(t)) - C_{st_{\text{min}}})^2, \]
\[ P_i(t) = \text{real power output of the } i\text{th unit of the } j\text{th individual at time } t. \]
\[ C_{st_{\text{min}}} = \text{minimum total fuel cost at } P_{i,\text{min}} i=1,...,N \]
\( \text{(bhat)}, \)
\[ k = \text{a constant value.} \]

The fitness function evaluates a power balance difference and a total fuel cost difference. By experiment, \( k = 0.001 \) is used to scale down the total fuel cost difference squared component. Otherwise, the cost difference squared component will dominate the power balance difference squared over the total load demand component. After decoding the \( j\)th individual to \([P_1(t),...,P_N(t)]\) and substituting in Eq. (9),

if \( P_{d}(t) < P_{N,\text{low}}(t), P_{d}(t)=P_{N,\text{low}}(t), \) and
if \( P_{d}(t) > P_{N,\text{high}}(t), P_{d}(t)=P_{N,\text{high}}(t). \)

We use \([P_1(t),...,P_N(t)]\) to evaluate the fitness value in Eq. (11) even though the power balance may not be satisfied. By using Eq. (11), the fitness value is ensured to be in the range of 0 to 1.
This reproduction method is based on the biased roulette wheel or “survival of the fittest” aspect [8]. The bigger the string individual fitness value, the higher the probability to have copies of them in the mating pool. For a population size \( NP \), a reproduction probability and cumulative reproduction probability of the \( i \)th string individual with fitness value \( f_i \), are

\[
    p_{np,i} = \frac{f_i}{\sum f_j}; \quad \text{Cumulative } p_{np,i} = \sum_{j=1}^{i} p_{np,k}
\]

(12)

Where \( j = 1, ..., NP \). The selected parents' string individuals to be copied to the mating pool are those having cumulative \( p_{np} \) just above the real number randomized between 0 and 1.

### 3.1.4 Crossover

Crossover is a process of exchanging bits between two string individuals. In particular, two individuals from the mating pool are randomly selected as parent individuals based on the biased roulette wheel. Then arbitrary positions on both individuals are chosen for crossing locations, where the exchanges of bits take place. A crossing mask is employed to determine the crossing locations. Two parent individuals will exchange their bits at every location where the corresponding position in the mask is one. As an example, it is assumed that two 30 bit string parent individuals are selected for two-point and uniform crossover as follows:

**Position:** 1234567890 1234567890 1234567890 1234567890 1234567890 1234567890 1234567890

**Parent1:** 0111111111 1111111111 1111000000 0010011011

**Parent2:** 1111010110 1111011011 1111101101 1111000000 1111000000

For the two-point crossover, the mask comprises one set of 1's bits surrounded by two sets of 0's bits [9]. Two crossing locations are arbitrarily selected from bit positions 2 to 30. If the crossing locations are 5 and 23, we will have

**Position:** 1234567890 1234567890 1234567890 1234567890 1234567890 1234567890 1234567890

**Mask:** 0000111111 1111111111 1111000000 0000000000

**Offspring1:** 0111010110 1111011011 1111001100 1010011011

**Offspring2:** 1111010110 1111011011 1111001100 1010011011

The uniform crossover generates new offspring individuals to participate in genetic process. Without crossover, the fittest individual is obtained from the initial random population. However, every individual to cross with another one after reproduction (crossover probability = 1.0), then we might lose many superior individuals. It has been shown in [10] that the convergence rate of uniform crossover is faster than the two-point crossover. Hence, the uniform crossover is used in this paper.

### 3.1.5 Mutation

Mutation is a process of flipping bits in a randomly chosen offspring string individual at random positions after performing crossover. In other words, it is a toggle from 0 to 1 or vice versa in a binary based system. Mutation is designed to give the offspring individual characteristics which do not exist in parent individuals. The rate of mutation is much lower than that of crossover since it is considered to be a secondary role. It is about 0.001 to 0.01 depending upon the types of applications. For instance, if we have offspring individual bits mutated at positions 2, 15, and 21, then

**Mutation Should be employed with caution since the high mutation rate will deteriorate the search performance.**

### 3.2 Micro Genetic Algorithm

Micro GA, one of the variation of the conventional SGA, was originally proposed by Krishnakumar [11]. In addition to performing four basic operations, MGA uses the elitism principle and checks the convergence of population at the end of each generation. Since any of the four basic operations do not guarantee that the new population string individuals are always better. Elitism guarantees that the best string individual survives. If the best individual in the current generation is worse
Table 1. Input-Output Characteristic of Rayong CC Units

<table>
<thead>
<tr>
<th>RY unit no.</th>
<th>( P_{\text{min}} ) (MW)</th>
<th>( P_{\text{max}} ) (MW)</th>
<th>Fuel cost (baht/Gcal)</th>
<th>Input-output coefficient (GCal/hr)</th>
<th>Ramp rate (MW/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( A_i )</td>
<td>( B_i )</td>
</tr>
<tr>
<td>1</td>
<td>100</td>
<td>300</td>
<td>273.8</td>
<td>-123.26930</td>
<td>3.1770742</td>
</tr>
<tr>
<td>2</td>
<td>100</td>
<td>300</td>
<td>273.8</td>
<td>-127.79030</td>
<td>3.2410401</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>150</td>
<td>273.8</td>
<td>-30.27881</td>
<td>2.2650570</td>
</tr>
<tr>
<td>4</td>
<td>100</td>
<td>300</td>
<td>273.8</td>
<td>-127.79030</td>
<td>3.2410401</td>
</tr>
</tbody>
</table>

Table 2. Input-Output Characteristic of Khanom CC Units

<table>
<thead>
<tr>
<th>KN unit no.</th>
<th>( P_{\text{min}} ) (MW)</th>
<th>( P_{\text{min}} ) (MW)</th>
<th>Fuel cost (baht/Mbtu)</th>
<th>Input-output coefficient (Mbtu/hr)</th>
<th>Ramp rate (MW/hr)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>( A_i )</td>
<td>( B_i )</td>
</tr>
<tr>
<td>1.1</td>
<td>376</td>
<td>495</td>
<td>79.413</td>
<td>450.5580</td>
<td>7.3659</td>
</tr>
<tr>
<td>1.2</td>
<td>-</td>
<td>495</td>
<td>678</td>
<td>820.2499</td>
<td>6.6173</td>
</tr>
</tbody>
</table>

than the previous generation, the best individual of previous generation will be randomly replaced to any individual of the current generation that would be parent individuals of the next generation. Consequently, MGA with elitism guarantees that at least the best individual exists until the last generation.

MGA checks the convergence after applying elitism. If the best string individual at each generation, which has the highest fitness value, has the total number of bit difference from the other individuals less than 5% of the total number of bits in the population size \((N \times (N-1) \times NP))\), that population is converged. Then MGA needs to copy the best individual to the next generation whereas the other individuals are all re-initialized [11].

In general, MGA performs well on a very small size of population. By experiment, the mutation and uniform crossover probability are 0.00 and 0.50 respectively. The attractive aspect of MGA is that it requires a relatively smaller population size than the SGA which results in less computation time. Moreover, once the population is converged, MGA tries to explore another set of solutions which would result in a higher chance to bail the solutions out of the local optimal solutions.

4. Experimental Results

For cost functions data, four units of Rayong (RY) CC and one unit of Khanom (KN) CC power plants [2] are used as benchmark data because their IC functions are linear decreasing and decreasing staircase functions, respectively. The low and high operating limits, coefficients of input-output functions, ramp rates, as well as the gas fuel costs of RY and KN units are shown in Tables 1 and 2, respectively. It is noted that an input-output function is given as \( A_i + B_i P_i + C_i P_i^2 \) and the cost function is calculated by multiplying the input-output function by the fuel cost.

Table 3. Parameter Selection of SGA and MGA

<table>
<thead>
<tr>
<th>Method</th>
<th>Pop. size</th>
<th>Crossover prob.</th>
<th>Mutation prob.</th>
<th>Maximum gen. limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>SGA with concatenated (SGAC)</td>
<td>90</td>
<td>0.5</td>
<td>0.02</td>
<td>1500</td>
</tr>
<tr>
<td>SGA with embedded (SGAE)</td>
<td>55</td>
<td>0.45</td>
<td>0.01</td>
<td>1500</td>
</tr>
<tr>
<td>MGA with concatenated (MGAC)</td>
<td>5</td>
<td>0.5</td>
<td>0</td>
<td>500</td>
</tr>
<tr>
<td>MGA with embedded (MGAE)</td>
<td>5</td>
<td>0.5</td>
<td>0</td>
<td>500</td>
</tr>
</tbody>
</table>

According to our experiments, the parameters selection which are suitable for our ED problem are shown in Table 3. These parameters include the population size, crossover probability, mutation probability, and maximum generation limits. SGA and MGA with concatenated and embedded encoding methods are investigated and compared. Note SGA retains the best individual for each generation and its solution is selected from the best individuals of all generations.
Computer programs for the SGA and MGA were developed in Pascal programming language. The experimental results obtained for increasing load demands from 776 to 1728 MW are compared to the unconstrained BF, MOL, and equal-lambda based Newton results. The load demand step is 10 MW. The ramp rate constraints in MW/hr are taken into account for each step.

<table>
<thead>
<tr>
<th>Method</th>
<th>Total fuel costs (baht)</th>
<th>Total cost difference from BF (baht)</th>
<th>% total cost difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>BF</td>
<td>68,364,198</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MGAC</td>
<td>68,484,219</td>
<td>120,021</td>
<td>0.1756</td>
</tr>
<tr>
<td>MGAE</td>
<td>68,489,662</td>
<td>125,464</td>
<td>0.1835</td>
</tr>
<tr>
<td>SGAC</td>
<td>68,521,126</td>
<td>156,928</td>
<td>0.2295</td>
</tr>
<tr>
<td>SGAE</td>
<td>68,521,520</td>
<td>157,322</td>
<td>0.2301</td>
</tr>
<tr>
<td>MOL Prim</td>
<td>68,665,609</td>
<td>301,411</td>
<td>0.4409</td>
</tr>
<tr>
<td>MOL Pmax</td>
<td>68,750,475</td>
<td>386,277</td>
<td>0.5650</td>
</tr>
<tr>
<td>Newton</td>
<td>70,366,638</td>
<td>2,002,440</td>
<td>2.9291</td>
</tr>
</tbody>
</table>

As shown in Table 4, the percentage of total fuel cost difference of all load demand steps of MGA with concatenated and embedded encoding methods are 0.1756% and 0.1835% higher than the unconstrained BF whereas SGA with concatenated and embedded encoding methods are 0.2295% and 0.2301% higher. MGA with concatenated encoding method has a lower fuel cost than the MOL based on the unit lambda values at $P_{\text{min}}$ and $P_{\text{max}}$ results in [2] by 0.2653% and 0.3894%, respectively. This improvement of ED program would therefore result in substantial fuel cost savings.

5. Conclusions

A Micro Genetic algorithm (MGA) method to solving the ramp rate constrained economic dispatch problem with linear decreasing IC and staircase IC functions is proposed. The MGA outperforms SGA in terms of lower total fuel costs and faster computing times since it has a higher chance to obtain the optimal solution with a relatively smaller size of population. It is shown that the total fuel cost of MGA is less than MOL, leading to substantial fuel cost savings. Real-time implementation of ED for EGAT CC units with linear decreasing and staircase IC functions by MGA is potentially viable.

Acknowledgments

The authors would like to thank the National Electronics and Computer Technology Center (NECTEC) for their partial support under contract NT-38-06-CST-13-01. In addition, the authors would like to thank the Electricity Generating Authority of Thailand (EGAT) for the test data. Finally, the authors would like to thank Mr. Thakolsri for their assistance in programming work.

References


Biographies

Dr. Weerakorn ONSAKUL received his B.Eng. degree in Electrical Engineering from Chulalongkorn University, Bangkok, Thailand in 1988. He received his M.S. and Ph.D. degrees in Electrical Engineering from Texas A&M University in 1991 and 1994 respectively. He joined the department of Electrical Engineering, Sirindhorn International Institute of Technology, Thammasat University, Thailand in 1995. He is currently an assistant professor there. His current interests is in power system operation & control, computer applications to power systems, parallel processing applications, applications of genetic algorithms to power system optimization, and transmission pricing.

Mr. Jarurote TIPPA Y ACHAI received his B.Eng. degree in Electrical Engineering from Sirindhorn International Institute of Technology, Thammasat University, Thailand in 1999. He has been working as a research assistant for the department of Electrical Engineering since April 1999. His current interest is in applications of genetic algorithms to power generation economic dispatch.


