



Multiple of Hybrid Lambda Iteration and Simulated Annealing Algorithm to Solve Economic Dispatch Problem with Ramp Rate Limit and Prohibited Operating Zones

Chiraphon Takeang¹ · Apinan Aurasopon¹

Received: 27 May 2018 / Revised: 18 September 2018 / Accepted: 24 October 2018 / Published online: 4 January 2019
© The Author(s) 2019

Abstract

Aims This paper presents a multiple hybrid methods combining the lambda iteration and simulated annealing methods (MHLSA) to solve an economic dispatch (ED) problem with smooth cost function characteristics. The constraints of the economic dispatch were load demand and transmission loss.

Methods The proposed MHLSA algorithm is a hybrid of the lambda iteration and simulated annealing methods and increases efficiency by adding multiple search mechanisms. It is a sequential execution of an individual hybrid algorithm of the lambda iteration and simulated annealing algorithm (HLSA) using only one personal microcomputer. The MHLSA algorithm introduces additional techniques for improvement of the search processes, such as the hybrid method, scope for multiple adaptive searches, and multiple searches.

Results To show its effectiveness, the MHLSA is applied to two systems consisting of three and six power generating units. The optimized results from MHLSA are compared with those from HLSA and from conventional approaches, such as bee colony optimization (BCO), hybrid particle swarm optimization (HPSO) and simulated annealing (SA). Results confirm that the proposed MHLSA approach is capable of obtaining the greater speed of convergence and a higher quality solution efficiently.

Conclusions Thus, it has great potential to be implemented in different types of power system optimization problem.

Keywords Multiple adaptive searches scope · Lambda iteration · Simulated annealing · Economic dispatch

1 Introduction

Increasing power demand, which has led to an increase in fuel costs over the years, has resulted in reducing operating costs of power systems becoming an important issue. One of the choices is to use the generator efficiently and economically. Operational economics, involving power generation and delivery, that minimizes power production costs, is called Economic Dispatch (ED). The objective of finding solutions to the ED problem is to control the committed generator's output to minimize total costs while the power demand and other constraints are satisfied.

Traditionally, the fuel cost function of a generator is represented by a single quadratic function. To solve the problem of ED, there are two approaches including numerical and meta-heuristics methods. Recently, many numerical conventional optimization methods have been tested, such as lambda iteration [1], gradient method [2], linear programming [3], interior point [4], lagrangian relaxation [5] and dynamic programming techniques [6]. These methods impose no restrictions on the nature of the cost curves and therefore can solve ED problems with inherently nonlinear and discontinuous costs. However, numerical methods can result in problems with complicated and large power systems where problems of dimensionality or local optimality might exist. Previously, meta-heuristics methods have been used to solve the economic dispatch and power electrical system problem, including the genetic algorithm (GA) [7–9], particle swarm optimization (PSO) [10–12], ant colony optimization (ACO) [13, 14], bee colony optimization (BCO) [15–17], the cuckoo search algorithm (CSA) [18], the shuffled frog leaping algorithm (SFLA) [19], the firefly

✉ Apinan Aurasopon
apinan.a@msu.ac.th

Chiraphon Takeang
chiraphon@rmutl.ac.th

¹ Department of Electrical and Computer Engineering,
Maharakham University, Maha Sarakham, Thailand

algorithm (FA) [20] and simulated annealing (SA) [21, 22]. These methods often provide fast and reasonable solutions, including global optimization with short time searching. Among them, the SA method is a probabilistic technique for approximating the global optimum of a given function. Specifically, it approximates global optimization in a large search space and has fast convergence. However, the conventional SA algorithm can have a limitation in reaching a global optimum solution in a reasonable computational time when the initial solution is far away from the region where an optimum solution is required. In the SA algorithm, the initial population is randomly generated which is the reason for computation and convergence taking a long time as the population is initiated too early. This problem has been solved by estimating an initializing value and defining the search scope for the random method. Lambda iteration has been used as an initial value instead of the traditional random value, the scope of the search has sometimes been defined with the SA algorithm properties then used to find the best solution. However, this method can have the problem of narrowing and inappropriately restricting the search scope. Determining the proper scope of the search is difficult, so repeated testing can waste time. This problem can be solved by using a “multiple searches” method. Multiple searches help to find the region where the most appropriate global optimum solution exists.

2 Problem Formulation

The economic dispatch problem, with prohibited operating zones, is approached by allocating system load demand commitments to groups of generators over the scheduled time period so that they have minimal generation costs while satisfying physical constraints and operating requirements.

2.1 Objective Function

The objective function, corresponding to the production cost, can be approximated to be a quadratic function of the active power outputs from the generating units. Symbolically, it is represented as in (1):

$$\text{Minimize: } F_T = \sum_{i=1}^N F_i(P_i) \quad (1)$$

Usually, the fuel cost of a thermal unit is expressed as a quadratic polynomial of its output (P_i) as below (2):

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

where a_i , b_i and c_i are the cost coefficients of the i th generator; P_i is the power output of the i th generator.

2.2 Constraints

The objective functions are subject to the following constraints:

(a) Power balance constraint

The load capacity is equal to the sum of the total amount of electricity demand with a total power loss in the transmission system is as below (3):

$$\sum_{i=1}^N (P_i) = P_D + P_{loss} \quad (3)$$

where P_D is the load demand and P_{loss} is the total transmission network losses, which is a function of the unit power outputs that can be represented using B coefficients as below (4):

$$P_{loss} = \sum_{i=1}^N \sum_{j=1}^N P_i B_{ij} P_j + \sum_{j=1}^N B_{0j} P_j + B_{00} \quad (4)$$

(b) Generator rating constraint

The output power of each generating unit has to lie between a lower and an upper bound. This is represented by a pair of inequality constraints as follow (5):

$$P_i^{\min} \leq P_i \leq P_i^{\max} \quad (5)$$

where, P_i^{\min} and P_i^{\max} are respectively the lower and upper bounds for power outputs of the i th generating unit.

(c) Ramp-rate limits and prohibited operating zone

The limit is defined as the rate at which the output power generated can't change beyond a certain rate due to mechanical inertia in the thermal units. A decision in the current time period will affect a decision at a later time period due to variation in power demands from the present to a future time. There exist two possible cases in the actual operation of the available units. Increasing generation condition in Eq. (6) and decreasing generation condition in Eq. (7):

$$P_i^0 - P_i \leq UR_i \quad (6)$$

$$P_i^0 - P_i \geq DR_i \quad (7)$$

In the equation, P_i^0 is the generation power that the generation unit generated in the previous step, DR and UR show the fall and rise of the border values, therefore, the capacity of the generators within these the ranges is as below (8):

$$\max(P_i^{\min}, P_i^0 - DR_i) \leq P_i \leq \min(P_i^{\max}, P_i^0 + UR_i) \quad (8)$$

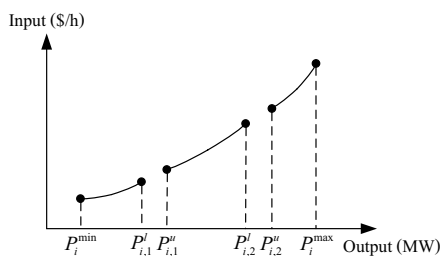


Fig. 1 Prohibited operating zone function cost curve

Performance of thermal power generators is restricted in the prohibited zone as the steam valve does not work properly at this time. Therefore, during this period, generators should not be operated. Zones for the operation of the generator are shown in Fig. 1.

Hence, dynamic economic dispatch problems with prohibited operating zones and with operational border values of the generation units, are solved as follows (9):

$$P_i = \begin{cases} P_i^{\min} \leq P_i \leq P_{i,1}^l \\ P_{i,j-1}^u \leq P_i \leq P_{i,j}^l, & j = 2, 3 \dots n_i \\ P_{i,n}^u \leq P_i \leq P_{i,1}^{\max} \end{cases} \quad (9)$$

3 Hybrid of Lambda Iteration (λ) and Simulated Annealing (HLSA)

HLSA is an initial estimate using the principle of equal cost (incremental cost: λ), defined boundaries around λ values, and performing each procedure according to the SA method. The SA, lambda iteration (λ) and HLSA for solving the ED problem are described below.

3.1 Simulated Annealing (SA) [23]

The Simulated Annealing (SA) algorithm is a nature-inspired method which is adapted from the process of gradual cooling of metal in nature. In the metallurgical annealing process, a solid is melted at high temperature until all molecules can move about freely and then a cooling process is performed until thermal mobility is lost. SA is a probabilistic technique for approximating the global optimum of a given function. Specifically, it is a meta-heuristic approach to approximating global optimization in a large search space. It is often used when the search space is discrete. For problems where finding an approximate global optimum is more important than finding a precise local optimum in a fixed amount of time, simulated annealing may be the preferable method. In general, Simulated

Annealing algorithms work as follows. At each time step, the algorithm randomly selects a solution close to the current one, measures its quality, and then decides to move to that solution or to stay with the current solution based on either one of two probabilities.

The basic step of the SA algorithm is presented with the following pseudo-code.

```

s ← s0; e ← E(s)
sbest ← s; ebest ← e
k ← 0
snew ← neighbour(s)
enew ← E(snew)
if P(e, enew, temp(k/kmax)) > random() then
s ← snew; e ← enew
if enew < ebest then
sbest ← snew; ebest ← enew
k ← k + 1
return sbest
    
```

Additional mechanisms for the SA method include controlled parameters for the SA algorithm including: starting Temperature, Final Temperature, Temperature Decrement and Iterations at each Temperature, which are described as follows.

(a) Starting Temperature

The starting temperature must be set to a reasonable value so that it is very likely to be accepted as the solution in the first step of the application of the algorithm. However, if the default temperature is too high, the SA step will not occur because of slow convergence, and generally, the optimization process will cause random. In contrast, if the initial temperature is very low, it is possible to get more local solutions. There are no specific methods for finding the right starting temperature that deals with the whole range of problems. It is advisable to increase the system temperature quickly to the point at which the worst solution percentage is acceptable and then thereafter to gradually decrease the temperature.

(b) Final Temperature

During the application of the SA algorithm, it is common to let the temperature fall to zero degrees. However, if the temperature decrease becomes an exponent, the SA step can work longer. Finally, the stopping threshold may be a low temperature or a suitable spot when the system is “frozen” at the current temperature.

(c) Temperature Decrement

Once the initial and final temperatures are set, it is necessary to find a way to change from the initial to the final temperature.

How the way in which the temperature is reduced is very important for the success of the algorithm. It is suggested that the following way is used to reduce the temperature (10):

$$T(t) = \frac{d}{\log(t)} \quad (10)$$

where d is a positive constant. The alternative is the geometric relationship of Eq. (11):

$$T(t) = a.t \quad (11)$$

where parameter a , is a constant near 1. In effect, its typical values range between 0.8 and 0.99.

(d) Iterations at Each Temperature

Optimizing the number of repetitions of the algorithm is very important. Using a certain number of iterations for each temperature is the proper solution. Indicate the actuation of a single repetition for each temperature, while the lowering of the temperature should occur at a very slow pace, which can be demonstrated by Eq. (12):

$$T(t) = \frac{t}{(1 + \beta.t)} \quad (12)$$

where, β takes a very low value.

3.2 Lambda Iteration (λ)

At the solution, multiple generators have the same marginal (or incremental) cost. This common marginal cost is equal to λ . If the cost difference between generators is different, by reducing capacity at higher cost margins and increasing capacity at lower cost margins, overall costs can be reduced. If demand changes, changes in total costs can be estimated from λ , and the solved value of λ can be used to evaluate P_i . Equation (2) is the fuel cost of power generation of each generator. This equation is used to calculate the incremental cost (λ) using the differential equation shown in Eq. (13):

$$\frac{d(F_i(p_i))}{d(p_i)} = \lambda \quad (13)$$

Total fuel cost is lowest when λ values are equal for each generator. The value of λ for the initial configuration of the system can be calculated by using the following (14) and the electric power of each generator can be calculated from Eq. (15):

$$\lambda = \frac{P_D + \sum_{i=1}^N \frac{b_i}{2a_i}}{\sum_{i=1}^N \frac{1}{2a_i}} \quad (14)$$

$$P_i = \frac{\lambda - b_i}{2a_i} \quad (15)$$

3.3 Hybrid of Lambda Iteration (λ) and Simulated Annealing (HLSA) for Solving the ED Problem

A hybrid of lambda iteration and simulated annealing is used to solve the economic dispatch problem by using the principle of equal cost (incremental cost: λ), an estimate of the initial value to narrow search scope and to find the most appropriate solution as SA properties around the estimates. The HLSA algorithm to solve the economic dispatch is described as follows:

Step 1: Specify the parameters of hybrid lambda iteration and simulated annealing including the initial temperature, the final temperature, cool shed, maximum number of iterations and the maximum number of successes.

Step 2: Calculate the value of λ for the initial configuration of the system using the SA algorithm. In this process, the initial λ is determined from Eq. (14).

Step 3: Find the lower and upper limits of the i th generating unit using the follow (16)–(17):

$$P_i^{\max} = \frac{\lambda - b_i}{c_i} + \left(\frac{\lambda - b_i}{c_i} \cdot \text{rank} \right) \quad (16)$$

$$P_i^{\min} = \frac{\lambda - b_i}{c_i} - \left(\frac{\lambda - b_i}{c_i} \cdot \text{rank} \right) \quad (17)$$

b_i and c_i are the cost coefficients of the i th generator and rank is the multiplier to determine the extent, the solution is more than 0 and less than 1.

Step 4: Random to the initial population (S_i) of the power output of the i th generation from step 3 can be expressed as follows (18):

$$P_i = P_i^{\min} + ((P_i^{\max} - P_i^{\min}) \cdot \text{rand}(1)) \quad (18)$$

The value must satisfy the system requirements by considering the system conditions. Capacity constraints of the generator must be within the minimum and maximum ranges of each generator.

Step 5: Create a random close proximity (S_j) and calculate the cost of production to compare with the value obtained from Step 4.

Step 6: If the new answer is better than the original answer, then the new answer will be chosen. But if the new answer is inferior to the original answer, calculate the deviation of cost $\Delta S = S_j - S_i$ and generate random numbers. Then accept the new solution (S_j) instead of the original solution (S_i).

Step 7: Proceed to temperature adjustment in the next state from Eq. (19):

$$T_K = r * T_{K-1} \quad (19)$$

Step 8: Repeat until all conditions have been met.

4 Multiple Hybrid of Lambda Iteration (λ) and Simulated Annealing (MHLSA)

Multiple searches help to find the regions with the most appropriate global solution. MHLSA uses multiple search methods of HLSA at the same time. The concept of MHLSA arises from the solution of the HLSA method in Sect. 3.3. In Step 3, configuring rank to get the right value is difficult, so repeated testing can waste time. Convergence speeds can be improved by implementing multiple searches. The MHLSA algorithm uses several initial solutions to maximize the probability of reaching the area with the most optimal solution. The procedure of the MHLSA algorithm is shown in Fig. 2, which consists of several independent HLSA algorithms. Additional mechanisms are as follows.

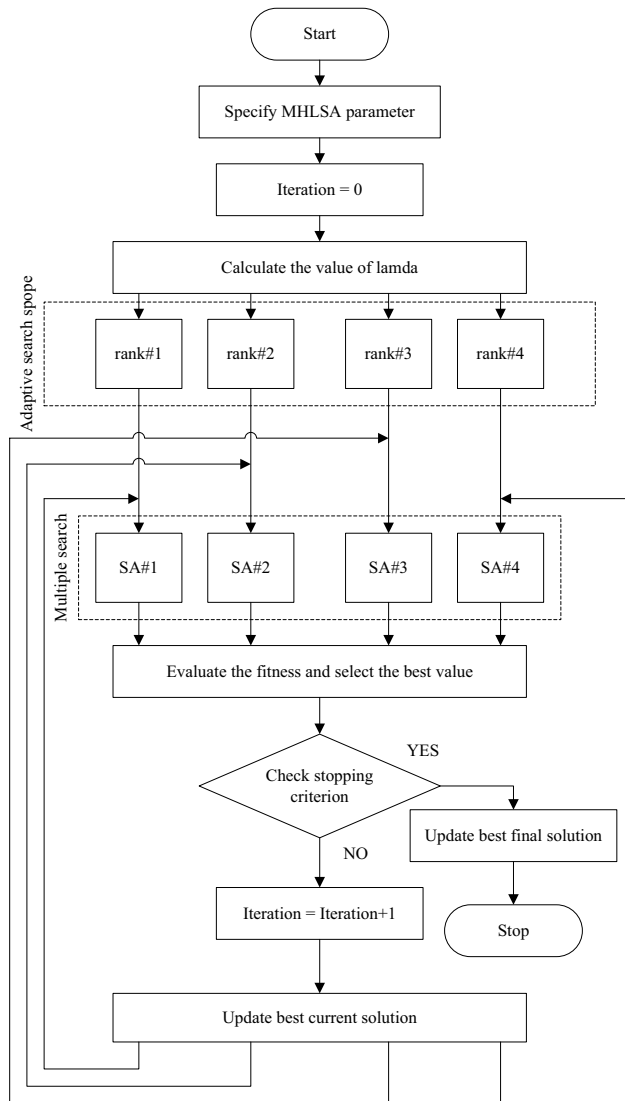


Fig. 2 Proposed HLIBCO algorithm

4.1 Specify Parameter

The parameters used in the SA algorithm are initial temperature, final temperature, cool shed, maximum number of tries, and maximum number of successes within. The parameter values used in SA, HLSA, and MHLSA to test the operation of the simulation are presented in Table 1.

4.2 Calculate the Value of Lambda (λ)

Lambda iteration is based on the principle of equal cost (incremental cost: λ), and an estimate of the initial value to narrow the search scope and finding the most appropriate solution as a swarm of bees around the estimates. Total fuel cost is lowest when λ is equal among the generators. The value of λ for the initial configuration of the system is calculated from Eq. (14).

4.3 Multiple Adaptive Search Scope

In this method, *rank* is a multiplier in scoping the answer. It is used to determine the lower and upper power limits of the i^{th} generating unit by defining the scope of the value of λ , which comes from Eqs. (16) and (17). The size of *rank* effects search answers. Determining the proper *rank* size of the search is difficult, so repeated testing can waste time. Adjusting the size of *rank* to multiple values at the same time provides a scalable range for the search. The adaptive search scope is a process for finding nearby locations. This process depends on the size of the *rank*. *Rank* is the variance, which is the key factor in the search. Hence the right size *rank* must be chosen. In general, low *rank* sizes can be used to reduce computational time due to narrower search spans. On the other hand, a *rank* size with a high value provides a more accurate solution but makes the calculation time longer. Nevertheless, the searched result may not reach the global optimum. In this method, the *rank* size can be adjusted based on the number iterations of the SA algorithm as defined in the MHLSA algorithm. As a result, the problem can be solved more accurately. Therefore, the adaptive search scope improves the calculation

Table 1 The parameters of SA, HLSA, and MHLSA

Parameters	Number
Initial temperature	100
Final temperature	5
Cool shed	0.83
Maximum number of tries	1000
Maximum number of successes within	50

speed and accuracy of the solution. Multiple *rank* values in this article are defined as follows: *rank* #1 = 0.1, *rank* #2 = 0.15, *rank* #3 = 0.2 and *rank* #4 = 0.25.

4.4 Multiple Search

In large-scale solutions, it may be necessary to use several computers at the same time to calculate the result. These solutions involve parallel searches. However, the multiple searches implemented in this paper are performed using a single personal computer [24]. The basic idea is to reduce the time spent searching results by improving the accuracy and relevance of each search, as well as the ability to manage the results. Multiple searches use a multitasking algorithm that combines algorithm features and additional search capabilities. The answer may come from one or more algorithms. Multiple searches are a way of taking advantage of the power of many algorithm, that may not be as eloquent as the traditional algorithm, and help to find the promising region where the global optimum solution exists. This concept is used to improve the efficiency of the HLSA algorithm after initial estimation by the lambda iteration and to define the range of multiple searches scope, as described in Sect. 4.3. The search algorithm for better algorithms is used in the SA algorithm step as described in Sect. 3.1. The sequence of execution starts from SA#1 to SA#4 for execution for finding better solutions.

4.5 Evaluate the Fitness and Select the Best Value

The fitness function is the specificity of the objective function that is used to summarize similar design approaches to achieve the goal that is set. After each round of testing or simulation, the idea was to bring the worst design solution out and rebuild it from the best design solution. The results of each solution have to be good enough to indicate an approximation of the overall specification, and this is built on using the fitness function to test or simulate the results of the solution. If the algorithm is poorly designed, it will come to an abnormal solution or will have difficulty converging. Moreover, the result must be computed quickly. The whole process involves “evaluating the fitness and selecting the best value”.

4.6 Check Stopping Criterion

In general, there are several possible conditions to stop a search. Here, stop search is used if any of the following three conditions are met: firstly, as defined by the limit time. That means that if the calculation time is more than or equal to the time limit, the search process will stop. Secondly, defined by the maximum allowable iterations—if the iteration number is greater than or equal to the maximum allowed number of

Table 2 Generator data in case 1

Unit	a_i	b_i	c_i	P_i^{\min}	P_i^{\max}
1	0.00525	8.663	328.13	50	250
2	0.00609	10.04	136.91	5	150
3	0.00592	9.76	59.16	15	100

Table 3 Ram rate limits and prohibited zone in case 1

Unit	P_i^0	UR_i	DR_i	Prohibited zone	
				Zone 1	Zone 2
1	215	55	95	[105–117]	[165–177]
2	72	55	78	[50–60]	[92–102]
3	98	45	64	[25–32]	[60–67]

iterations, the search process will stop. Finally, when the appropriate values meet the objective function within the constraints and conditions of the system.

5 Case Study

The proposed MHLISA algorithm has been applied to solving the economic dispatch problem with prohibited operating zones in two different test cases for verifying its feasibility. These are a three unit system and a six unit system. Each optimization method was implemented in the MATLAB program which was run on a 2.30 GHz Intel (R) Core (TM) i5 with 8 GB of RAM.

5.1 The First Case Study

A three unit system is considered and the system load demand was set as 300 MW. The system data are shown in Tables 2 and 3. The system loss coefficient matrix was as follows [25]:

$$B_{ij} = \begin{bmatrix} 0.000136 & 0.0000175 & 0.000184 \\ 0.0000175 & 0.000154 & 0.000283 \\ 0.000184 & 0.000283 & 0.00161 \end{bmatrix}$$

The results of the proposed method are divided into two parts: convergence speed and convergence with BCO, SA and HLSA methods and the best solution compared to EP, PSO and HPSO methods.

5.2 The Second Case Study

The test system for this case consisted of six thermal units. The constraints included the generation limits, fuel cost

Table 4 Generator data in case 2

Unit	a_i	b_i	c_i	P_i^{\min}	P_i^{\max}
1	0.0070	7.00	240	100	500
2	0.0095	10.0	200	50	200
3	0.0090	8.50	220	80	300
4	0.0090	11.0	200	50	150
5	0.0080	10.5	220	50	200
6	0.0075	12.0	190	50	120

Table 5 Ram rate limits and prohibited zone in case 2

Unit	P_i^0	UR_i	DR_i	Prohibited zone	
				Zone 1	Zone 2
1	340	80	120	[210–240]	[350–380]
2	134	50	90	[90–110]	[140–160]
3	240	65	100	[150–170]	[210–240]
4	90	50	90	[80–90]	[110–120]
5	110	50	90	[90–110]	[140–150]
6	52	50	90	[75–85]	[100–105]

coefficients, ramp-rate limits and prohibited operating zones of these thermal units, 26 buses and 46 transmission lines. The transmission loss was calculated using a B matrix as given by [26]. The results of the proposed method were divided into two parts: convergence speed and convergence with BCO, SA and HLSA methods and the best solution compared to PSO, HPSO and MSSA methods.

The characteristics of the six thermal units are given in Tables 4 and 5 [26].

$$B_{ij} = 10^{-1} \begin{bmatrix} 0.017 & 0.012 & 0.007 & -0.001 & -0.005 & -0.002 \\ 0.012 & 0.014 & 0.009 & 0.001 & -0.006 & -0.001 \\ 0.007 & 0.009 & 0.031 & 0.0 & -0.010 & -0.006 \\ -0.001 & 0.001 & 0.0 & 0.24 & -0.006 & -0.008 \\ -0.005 & -0.006 & -0.010 & -0.006 & 0.129 & -0.002 \\ -0.002 & -0.001 & -0.006 & -0.008 & -0.002 & 0.15 \end{bmatrix}$$

$$B_{0i} = 10^{-2} \times [-0.3908 \quad -0.1297 \quad 0.7047 \quad 0.0591 \quad 0.2161 \quad -0.6635]$$

$$B_{00} = 0.056$$

6 Simulation and Results

The convergence of speed is used as a basis for comparison. In this test, the parameters of SA, HLSA and MHLSA are shown in Table 1 and the following parameters are used in the BCO method:

Table 6 Results of MHLSA, HLSA, SA and BCO for three units

Unit output	BCO	SA	HLSA	MHLSA
P_1 (MW)	199.57	200.40	191.19	191.65
P_2 (MW)	78.97	78.15	85.30	85.12
P_3 (MW)	34.00	34.02	34.63	34.00
P_T (MW)	312.54	312.56	311.12	310.76
T_C (\$/h)	3631.66	3631.71	3618.14	3614.64
P_{Loss}	12.54	12.56	11.12	10.76
Time (S)	14.10	10.13	10.06	40.34

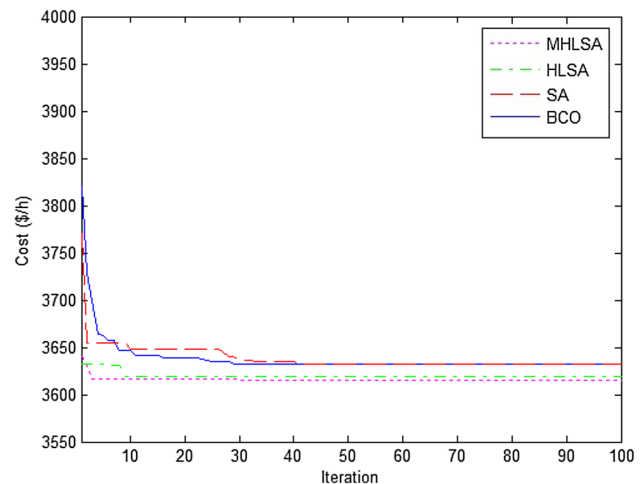


Fig. 3 Solution and convergence time of MHLSA, HLSA, SA, and BCO in case 1

- Population size (n) = 50
- Number of selected sites (m) = 20
- Number of best sites (e) = 5
- Number of bees around best sites (n_{ep}) = 50
- Number of bees around other sites (n_{sp}) = 50

Case 1, simulation results for the selected values using BCO, SA, HLSA, and the MHLSA methods. Case study 1, 200 trail runs were conducted, and the best solutions that satisfy the system constraints are shown in Table 6.

The total cost of the MHLSA method was less than that of the BCO, HLSA and SA methods and the loss in the transmission line was also lower. Computational and convergence times are shown in Fig. 3. The MHLSA method used fewer iterations to find the solution, and the convergence was faster than the BCO, SA and HLSA methods and the results were comparable to EP [27], PSO [27] and HPSO [27] (Table 7).

Case 2, This case study utilized a benchmark test system having 26 buses, 46 transmission lines, and 6 thermal generating units. The total system load was 1263 MW. 200 trail runs were conducted, and the best solutions are shown in

Table 7 Three generators test system: comparison of results

Unit output	EP	PSO	HPSO	MHLSA
P_1 (MW)	199.53	190.59	200.18	191.65
P_2 (MW)	75.68	85.77	76.26	85.12
P_3 (MW)	38.19	34.80	34.40	34.00
P_T (MW)	313.40	311.16	310.84	310.76
T_C (\$/h)	3641.70	3631.10	3623.11	3614.64
P_{Loss}	13.40	11.16	10.84	10.76

Table 8 Results of MHLSA, HLSA, SA and BCO for six generating unit

Unit output	BCO	SA	HLSA	MHLSA
P_1 (MW)	452.52	447.26	455.52	450.44
P_2 (MW)	171.56	170.47	171.62	170.83
P_3 (MW)	256.61	261.92	254.98	253.33
P_4 (MW)	138.45	138.53	134.60	137.81
P_5 (MW)	163.97	166.46	155.10	160.93
P_6 (MW)	92.29	90.55	99.62	98.02
P_T (MW)	1275.10	1275.20	1271.44	1271.36
T_C (\$/h)	15,439.61	15,439.80	15,394	15,392
P_{Loss}	12.09	12.20	8.44	8.36
Time (S)	15.56	10.26	10.07	60.02

Table 8 that satisfies the system constraints, including the power balance and the generator limits constraints, ramp rate limits, prohibited operating zones, and transmission losses. Table 8 presents a comparison between the results obtained by the BCO, SA, HLSA, and the MHLSA algorithms.

Computational and convergence times were as shown in Fig. 4, which clearly demonstrates the speed of the convergence approach, where the MHLSA method used through iteration to find fewer answers. Converge was faster than with the BCO, SA and HLSA methods.

A comparison of results using the proposed methods with those from PSO [27], HPSO [27] and MSSA [28] are shown in Table 9, which the proposed method gets a better than the other ways.

7 Conclusion

In this work, new mechanisms are implemented for solving different ED problems with ramp rate limits and prohibited operating zones. Many sophisticated techniques, such as initial estimation using the lambda iteration, adaptive search scopes, multiple searches, and simulated annealing have been added to the MHLSA, in order to enhance the search potential. The MHLSA algorithm has been proposed to solve the ED problem

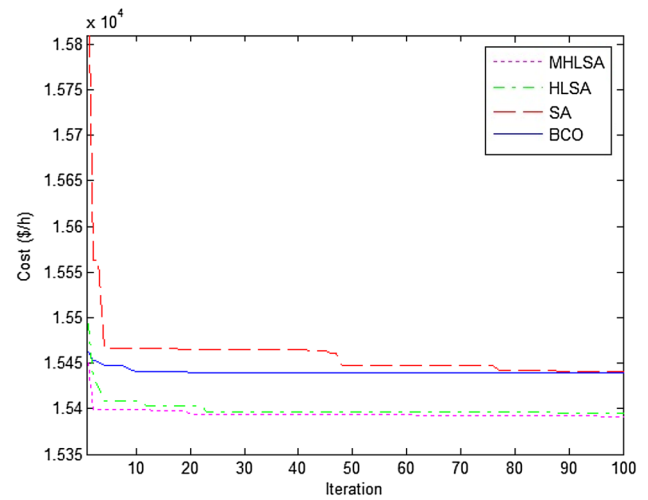


Fig. 4 Solution and convergence time of MHLSA, HLSA, SA and BCO in case 2

Table 9 Six generators test system: comparison of results

Unit output	MSSA	PSO	HPSO	MHLSA
P_1 (MW)	447.50	457.26	462.45	450.44
P_2 (MW)	173.32	160.72	184.53	170.83
P_3 (MW)	263.46	247.53	246.60	253.33
P_4 (MW)	139.07	131.52	108.83	137.81
P_5 (MW)	165.43	170.50	171.07	160.93
P_6 (MW)	87.13	106.62	98.50	98.02
P_T (MW)	1275.96	1274.15	1271.98	1271.36
T_C (\$/h)	15,449.90	15,433	15,404	15,392
P_{Loss}	12.96	11.15	8.98	8.36

by initial estimation using the lambda iteration method, using the SA method to find the best solution, and using multiple searches as a mechanism to increase efficiency in finding appropriate solutions. The MHLSA optimization mechanisms out-performed a number of other algorithms. The strength of the algorithm was demonstrated in two case studies involving different ED problems with ramp rate limits and prohibited operating zones. The convergence quality of MHLSA algorithm in these two case studies proved the robustness of the algorithm. Study results from the studies involving either three or six generating units confirmed that the MHLSA was superior to the HLSA, SA and BCO methods in terms of providing a high-quality solution, stable convergence characteristics, and good computation efficiency. The results clearly showed that the proposed algorithm yielded better results than the comparison method. Electric system operators can use this algorithm for optimization of their networks.

Acknowledgements The authors thank Maharakham University, Maharakham, Thailand, for the financial supporting of this project and I would like to thank Prof. Ian Warrington for encouragement and insightful comments.

OpenAccess This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- Wood AJ, Wollenberg BF (1984) Power generation operation and control. Wiley, New York
- Ramamoorthy M, Dhar RN, Mukherjee PK (1973) Reduced-gradient method for economic dispatch. *Electr Eng Proc Inst Electr Eng* 120(11):1419–1420
- Frag A, Al-Baiyat S, Cheng TC (1995) Economic load dispatch multi objective optimization procedures using linear programming techniques. *IEEE Trans Power Syst* 10(2):731–738
- Kimball JM, Clements KA, Davis PW, Nejdawi I (2002) Multiperiod hydrothermal economic dispatch by an interior point method. *Math Probl Eng* 8:33–42
- El-Keib AA, Ma H, Hart JL (1994) Environmentally constrained economic dispatch using the lagrangian relaxation method. *IEEE Trans Power Syst* 9(4):1723–1729
- Ringlee RJ, Williams DD (1962) Economic dispatch operation considering valve throttling losses, II-distribution of system loads by the method of dynamic programming. *IEEE Trans Power Appar Syst* 81(3):615–620
- Satyendra PS, Rachna T, Anubhar G (2014) Genetic algorithm for solving the economic load dispatch. *Int J Electr Electr Eng* 7(5):523–528
- Kumar V, Singh J, Singh Y, Sood S (2015) Optimal economic load dispatch using genetic algorithms. *Int J Electr Comput Energ Electron Commun Eng* 9(4):463–470
- Oluwadare SA, Iwasokun GB, Olabode O, Olusi O, Akinwonmi AE (2016) Genetic algorithm-based cost optimization model for power economic dispatch problem. *Br J Appl Sci Technol* 15(6):1–10
- Srikanth K, HariVamsi V (2016) Partical swarm optimization technique for dynamic economic dispatch. *Int J Res Eng Technol* 5(5):460–466
- Zhang H (2016) An improved particle swarm optimization algorithm for dynamic economic dispatch problems. *Int J Innov Res Eng Manag (IJIREM)* 3(4):264–266
- Lotfi H, Dadpour A, Samadi M (2017) Solving economic dispatch in competitive power market using improved particle swarm optimization algorithm. *Int J Smart Electr Eng* 6(1):35–41
- Thesia DK, Jangir P, Trivedi IN (2015) Economic emission dispatch problem solution using ant colony optimization of micro-grid in Island mode. *Int J Adv Eng Res Dev* 2(5):89–95
- Nwohu MN, Paul OO (2017) Evaluation of economic load dispatch problem in power generating stations by the use of ant colony search algorithms. *Int J Res Stud Electr Electron Eng (IJRSEEE)* 3(1):20–29
- Assareh E, Biglari M (2015) A novel approach to capture the maximum power generation from wind turbines using hybrid MLP neural network and bees algorithm (HNNBA). *IETE J Res* 62(3):368–378
- Tung NS, Chakravorty S (2016) Solution to economic power dispatch planning problem considering generator constraints using artificial bee colony algorithm. *Int J Hybrid Inf Technol* 9(5):399–406
- Mehta G, Singh RP, Yada VK (2017) Optimization of combined economic emission dispatch problem using artificial bee colony method. *Int J Cybern Inf (IJCI)* 6(1/2):107–118
- Nguyen TT, Vo DN, Dinh BH (2016) Cuckoo search algorithm for combined heat and power economic dispatch. *Electr Power Energy Syst ELSEVIER* 81:204–214
- Selvakumar K, Vijayakumar K, Sattianadan D, Boopathi CS (2016) Shuffled frog leaping algorithm (SFLA) for short term optimal scheduling of thermal units with emission limitation and prohibited operational zone (POZ) constraints. *Indian J Sci Technol* 9(42):1–6
- Panigrahi P, Mishra S, Pani SR (2015) Implementation of Firefly algorithm on economic load dispatch. *Int J Eng Res Technol (IJERT)* 4(04):376–380
- Venkatesh M, Raghutu R (2015) Economic load dispatch using simulated annealing algorithm. *Int J Eng Res Technol (IJERT)* 2(3):1961–1964
- Ismail Z, Farid B, Amel G (2017) Simulated annealing optimization for generation scheduling with cubic fuel cost function. *WSEAS Trans Inf Sci Appl* 14:64–69
- Vishwakarma KK, Dubey HM (2012) Simulated annealing based optimization for solving large scale economic load dispatch problems. *Int J Eng Res Technol (IJERT)* 1(3):1–8
- Sa-ngiamvibool W, Pothiya S, Ngamroo I (2011) Multiple tabu search algorithm for economic dispatch problem considering valve-point effects. *Electr Power Energy Syst* 33:846–854
- Naresh R, Dubey J, Sharma J (2004) Two-phase neural network based modeling frame work of constrained economic load dispatch. *IEE Proc Gen Trans Distrib* 151(3):373–378
- Khamsawang S, Jiriwibhakorn S (2009) Solving the economic dispatch problem using novel particle swarm optimization. *World Acad Sci Eng Technol Int J Electr Comput Eng* 3(3):529–539
- Prabakaran S, Senthilkumar V, Baskar G (2015) Economic dispatch using hybrid particle swarm optimization with prohibited operating zones and ramp rate limit constraints. *J Electr Eng Technol (JEET)* 10:30–40
- Elsayed WT, Hegazy YG, Bendary FM, El-bages MS (2016) Modified social spider algorithm for solving the economic dispatch problem. *Eng Sci Technol Int J* 19:1672–1681



Chiraphon Takeang He received a B.S. degree in Technical Education in Electrical Engineering from the King Mongkut's Institute of Technology North Bangkok, Thailand, in 2003, him M.S. degree in Electrical Technology form the King Mongkut's Institute of Technology North Bangkok, Bangkok, Thailand, in 2007. He is currently a Ph.D. student in Electrical and Computer Engineering at Mahasarakham University. His research interests include power system analysis, economic dispatch

problem and optimization technique.



Apinan Aurasopon He received his B.E. degree in Electronic Engineering form the Northeastern College, Khon Kaen, Thailand, in 1995, his M.E. and Ph.D. degree in Electrical Engineering from the King Mongkut's University of Technology Thonburi, Bangkok, Thailand, in 2003 and 2007, respectively. He was a lecturer in the Department of Electrical Engineering, Faculty of Engineering, Burapha University, Chonburi, Thailand, in 2007. He was transferred to the Faculty of Engineering, Mahasarakham University, Mahasarakham, Thailand, in 2008, where he is currently an Associate Professor. His current research interests include soft-switched converters, power factor and the application of electronics and computer for agriculture.