1. Chiraphon TAKEANG, 2. Apinan AURASOPON

Faculty of Engineering, Rajamangala University of Technology Lanna, Lampang, Thailand.
Faculty of Engineering, Mahasarakham University, Mahasarakham 44150, Thailand.
ORCID: 1. 0000-0001-6700-9454, 2. 0000-0002-4898-303X

doi:10.15199/48.2021.09.24

A New Hybrid Algorithm combining Ant Lion Optimization and Particle Swarm Optimization to Solve an Economic Dispatch Problem with non-smooth cost function

***Abstract**. This paper presents a new hybrid algorithm which is a combination of ant lion optimization (ALO) and particle swarm optimization (PSO) to solve an economic dispatch (ED) problem with non-smooth cost function characteristic. In the proposed algorithm, HALO-PSO, ALO method is used to find the initial value and PSO is used to find the best solutions causing it provides faster and more accurate results compared to conventional methods. To show its effectiveness, the HALO-PSO was applied to test two systems consisting of either 6 or 13 power generating units. Results confirm that the proposed HALO-PSO algorithm is capable of obtaining rapid convergence and a high quality solution efficiently.*

Streszczenie.** W artykule przedstawiono nowy algorytm hybrydowy, który jest kombinacją optymalizacji Ant Lion (ALO) i optymalizacji roju cząstek (PSO) w celu rozwiązania problemu ekonomicznej dystrybucji (ED) z niegładką charakterystyką funkcji kosztu. W proponowanym algorytmie HALO-PSO, metoda ALO służy do znalezienia wartości początkowej, a PSO służy do znalezienia najlepszych rozwiązań, dzięki czemu zapewnia szybsze i dokładniejsze wyniki w porównaniu do metod konwencjonalnych. Aby wykazać jego skuteczność, HALO-PSO został zastosowany do przetestowania dwóch systemów składających się z 6 lub 13 jednostek wytwórczych. Wyniki potwierdzają, że proponowany algorytm HALO-PSO jest w stanie skutecznie uzyskać szybką konwergencję i wysokiej jakości rozwiązanie.* (Nowy algorytm hybrydowy łączący optymalizację Ant Lion i optymalizację roju cząstek w celu rozwiązania ekonomicznego problemu dystrybucji z funkcją kosztów nierównomiernych***)

Keywords: Ant Lion Optimization, Particle Swarm Optimization, Hybrid Algorithm, Economic Dispatch.
Słowa kluczowe: ekonomiczna dystrybucja energii, algorytm rojowy, algoryt,m hybrydowy.

Introduction

The power system should operate under a high economic level to be competitive in terms of production costs. A key aim in addressing this critical concern for electrical system operation is optimization with unit commitment, while economic dispatch (ED) is an important subsection of the unit commitment process, it is necessary to efficiently obtain a fast, high-quality solution from ED. The ED is production level allocation to the various building blocks in the system to meet the load demand in the most economical manner without any system violation or the constraints of each unit, so that the lowest total production cost and able to deliver enough electrical power to meet the system requirements. Minimizing overall production costs is the primary objective of ED. The ED method is that generating units with different fuel consumption and different power generation capacity that must have the lowest combined fuel cost. The ED problem in the electrical system is determined by the fuel cost as a quadratic function. The practical ED problem is caused by a number of fuel valve point effects that are represented as a non-smooth optimization problem. With equality and inequality constraints, it is difficult to find the problem of global optimal values.

Regarding the ED problem, there were several of traditional methods that have been applied to handle this problem such as: Dynamic Programming [1], Linear Programming [2], Lagrangian Relaxation [3], etc. These methods often provide answers that are stuck at the local optimum, making it difficult to solve ED problems with non-smooth cost function. However, there were some attempts to find the new methodology for dealing with this difficulty. Recently meta-heuristics was used to solve economic dispatch problems and power system issues such as Tabu Search (TS) [4], Ant Colony Optimization (ACO) [5]-[6], Genetic Algorithm (GA) [7]-[9], Bee Colony Optimization (BCO) [10]-[12], Simulated Annealing (SA) [13]-[14], Ant Lion Optimization (ALO) [15]-[18], Shuffled Frog Leaping Algorithm (SFLA) [19]-[20] and Particle Swarm Optimization (PSO) [21]-[23]. These methods have obtained a lot of attention from many researchers due to their ability to find the best and an almost global optimal solution. However, these methods are random search. The key strategy of these methods is random search, which results in selecting the results to be searched in the next set, and this is the reason for long computation and convergence when the initial results are of no quality due to randomness. This issue was addressed by estimating initial values and defining search boundaries for random methods. Of all the above methods, a hybrid model can be built on the powerful properties of the ALO and PSO algorithms to achieve better results than using each separately. The ALO is an optimization technique developed recently [24] by Seyedali Mirjalili, the main inspiration of the ALO algorithm comes from the foraging behaviour of ant-lion's larvae, which mimics interaction between ant-lions and ants in the trap. It is simple in concept, few in parameters and easy in implementation. The weakness of the ALO algorithm is its simplicity, thus resulting in the algorithm covering local areas in the search area. The PSO was developed by Kennedy and Eberhart [25] based on simulations of simple social systems such as birds and fish schooling. The main problem with the original PSO was the speed towards initial population selection and premature convergence, although PSO as a random algorithm could be applied to ED problems with non- smooth cost function. These can occur when the best particles are in the worst solution group during the search process. As a result, randomly finding a new population in the next generation to solve it takes longer to compute because the algorithm has a bad convergence behavior. Consequently, convergence acceleration and local optimal avoidance have become two important and interesting goals in PSO research. There are many researches that modify the behavior of the PSO algorithm to improve its performance [26]-[29].

In this paper, a new hybrid algorithm combining ALO with PSO (HALO-PSO) is used to solve both static economic dispatches with non-smooth cost function. The proposed approach aims to enhance the exploitation of the ALO algorithm. The initial value is reassigned in place of the original randomization derived from the results of the ALO, set a new search scope around the initial value and it uses nice features from the PSO algorithm to find the best solution. The proposed approach focuses on minimizing the total fuel cost of all electric power generating units while meeting the conditions, constraints and power balance of system. The possibility of the proposed method is demonstrated by using two case studies with six generators and thirteen generators operating under static economic dispatch and non-smooth function conditions. The results of the proposed method are compared with previously published methods.

Problem Formulation of ED with Non-Smooth Cost Function

The ED problem is one of the nonlinear programming sub-problems of unit commitment. It is a short term determination of the optimal output of a number of power plants. To meet the system load at the lowest cost depending on delivery and operational constraints. The ED problems are solved by specialized computer software. This should be in line with the operational and system constraints of the available resources and the associated transmission capacity, the details are as follows.

Objective functions

The ED solutions are aimed at minimizing the total fuel cost of power plants subject to the operating constraints of a power system. The objective function is formulated as follows:

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

where F_T is the total generation cost, N is the number of generators committed to the operating system P_i is the power output of the i^{th} generator and F_i is the generation cost function of i^{th} generator is usually expressed as a quadratic polynomial as follows:

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

where a_i, b_i and c_i are the cost coefficients of the i^{th} generator. The smooth cost function is shown in this equation. The input-output characteristics or cost functions of generators are estimated by using quadratic or piecewise quadratic functions individually, under the assumption that the unit incremental cost curve is a linear and increases piece-by-piece repeatedly. However, real input-output characteristics display higher-order nonlinearities and discontinuities due to valve-point loading in fossil fuel burning plant. The valve-point loading has been modelled in a recurring rectified sinusoidal function, such as the one show in figure 1.

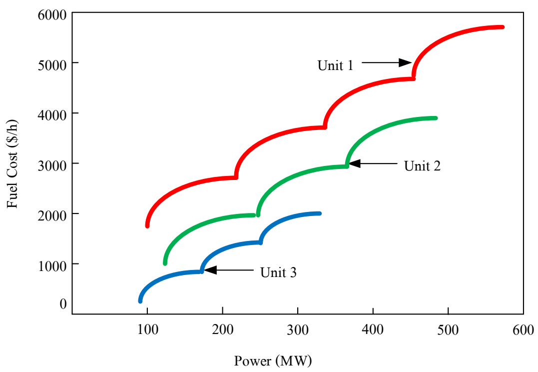

Fig.1. Characteristic of the non-smooth cost function

To model the cost function of generators in a more empirical manner, valve point effect is considered where the input-output, curve is not linear but consists of ripples as a result of the sharp increase in losses due to the wire drawing effects, which occur as each steam admission valve starts to open. For more accurate modeling, cost functions are derived from ripple curve. When a generator is with multiple valve points, the cost curve is not smooth. The assumption that the smooth cost function becomes null results in an erroneous result. The valve points effect can be taken into account by adding a sine term as in equation (3).

(3) $$F_i(P_i) = a_i P_i^2 + b_i P_i + c_i + \left| e_i \times \sin(f_i \times (P_{i,\min} - P_i)) \right|$$

where e_i and f_i are the cost coefficients of generator i^{th} reflecting valve-point effects.

Constrain

The objective functions are subject to the following constraints.

Power balance constraint

The total load capacity is equal to the sum of the total electrical demand with total power loss in the transmission system as:

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

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 follows:

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

Generation limits constraint

The output power of each generator unit must be between an upper and lower bound. This is represented by a pair of inequality constraints as:

(6) $$P_i^{\min} \le P_i \le P_i^{\max}$$

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

Hybrid Algorithm Combining Ant Lion Optimization and Particle Swarm Optimization to Solve an Economic Dispatch Problem

The presented new hybrid algorithm is a combination of the ALO and PSO method that is used to solve both static economic dispatches with non-smooth cost function. The idea of this algorithm is to use the solutions obtained from the ALO instead of the original random value. The solution or initial value is modified by the factor, $rank$, producing a new search scope that is around it. Finally, the nice feature of PSO algorithm is used to find the best solution. The details of all methods are described below.

Ant lion optimization

Ant Lion Optimizer (ALO) proposed by Sayedali Mirjalili in 2015 [24]. The ALO algorithm is inspired by the life cycle of the ant lion, which mimics the hunting mechanisms of ant lion in nature. The larva of the ant lion digs a cone-shaped hole in the sand, moving it along a circular path and discarding the sand by using its large jaws. After digging up the trap, the larva hides under the bottom of the cone and waits for the insect to get trapped in the cone-shaped hole. When the prey is caught, it will be pulled down the bottom of the hole and eaten. Therefrom, the ant lions throw the leftovers outside the pit and renovated the pit for the next hunt. The ant lion optimizer has very few parameters to adjust because the ALO is a population-based algorithm. There are five main steps of the algorithm such that random walk of ants, building traps, entrapment of ants in traps, catching preys, and re-building traps. Each step can be simulated in a mathematical equation and described as follows.

Random walk of ants

The interaction between the lion ant and the ant in the trap was modelled on the ALO algorithm. For such interaction models, the ants must move through the search area and the ant lions are allowed to hunt them and become stronger using traps. Wherewith ants move stochastically in nature when searching for food, randomly following the movements of ants can be simulated as follows:

(7) $$X(t)=[0, cums(2r(t_1)-1, cums(2r(t_2)-1, ..., cums(2r(t_n)-1]$$

where $cums$ calculates the cumulative sum, n is the maximum number of repetitions. t is the step of a random walk and $r(t)$ is a random function defined as Equation (8)

(8) $$r(t)=\begin{cases} 1, if \dots rand > 0.5 \\ 0, if \dots rand \le 0.5 \end{cases}$$

Where, $rand$ is a random number generated with a uniform distribution in the range of [0, 1]. Ant positions are stored and used during the optimization process in the following matrix:

(9) $$M_{ant}=\begin{bmatrix} ant_{1,1} & ant_{1,2} & \cdots & ant_{1,d} \\ ant_{2,1} & ant_{2,2} & \cdots & ant_{2,d} \\ \vdots & \vdots & & \vdots \\ ant_{n,1} & ant_{n,2} & \cdots & ant_{n,d} \end{bmatrix}$$

where, M_{ant} is matrix to save the position of each ant , ant_{ij} is value of j^{th} variable (dimension) of i^{th} ant, n is number of ants and d is number of variables. Random walk of ants are being normalized to keep them moving within the search space using the following equation:

(10) $$X_i^t=\frac{(X_i^t-a_i)\times(d_i-c_i^t)}{(d_i^t-a_i)}+c_i$$

where a_i indicates the minimum of random walk of i^{th} variable, d_i is the maximum of random walk in i^{th} variable, c_i^t is the minimum of i^{th} variable at t^{th} iteration, and d_i^t indicates the maximum i^{th} variable at t^{th} iteration.

Trapping in ant lion's pits

The equation used to describe the mathematical pattern of Trapping in ant lion's pits is as follows:

(11) $$c_i^t=Antlion_j^t+c^t$$

(12) $$d_i^t=Antlion_j^t+d^t$$

where c^t is the minimum of all variables at t^{th} iteration, d^t indicates the vector including the maximum of all variables at t^{th} iteration, c_i^t is the minimum of all variables for i^{th} ant, d_i^t is the maximum of all variables for i^{th} ant, and $Antlion_j^t$ shows the position of the selected j^{th} antlion at t^{th} iteration.

Building trap and sliding ants toward ant lion

The operator of the roulette wheel simulates the hunting abilities of the ant lion for selecting ant lions based on their fitness during iterations. This mechanism gives the ant lion a great opportunity to capture prey. Ants are necessary to move randomly and ant lion be able to build traps proportional to their fitness. Ant lions will shoot sands outwards the center of the pit, once the ant falls into the trap. This behavior makes the ant slides down in the trap. The radius of ant's random walk is reduced and it can be written as follows:

(13) $$c^{t(new)}=\frac{c^t}{I}$$

(14) $$d^{t(new)}=\frac{d^t}{I}$$

Where I is the ratio expressed as equation (15).

(15) $$I=10^w\times\frac{t}{T}$$

Where t is the number of iterations in the current cycle, T is the maximum number of iterations and w is the constant determined by the current iteration (w = 2 when $t>$ 0.1T, w = 3 when $t>$ 0.5T, w = 4 when $t>$ 0.75T, w = 5 when $t>$ 0.9T and w = 6 when $t>$ 0.95T). In general, the value of w is a constant that can be adjusted to achieve the correctness of the answer. These equations slow down the response radius and the oscillation of the answer, which improving the ant's position and mimicking the ant's trapping process within the hole.

Catching prey and re-building the pit

The final stage of the hunt is when the ant is pulled to the bottom of the pit and it is captured by the lion's jaw. Onwards, the ant lion pulls the ants into the sand and eat them. The ant lion then needs to be updated with its trap or position to be the latest location of prey to increase its chances of catching new prey. Equation (16) describes a trap improvement or repositioning.

(16) $$Antlion_j^t=Ant_i^t \;\; if \;\; f(Ant_i^t)>f(Antlion_j^t)$$

where $Antlion_j^t$ indicates the position of selected j^{th} ant lion at t^{th} iteration, Ant_i^t shows the position of i^{th} ant at t^{th} iteration and t shows the current iteration.

Elitism

An important feature of this algorithm is Elitism that is an evolution that allows the algorithm to maintain the obtained best solution at any stage of the optimization process. In this algorithm, the best ant lion obtained in each iteration is recorded and considered the elite. The elitism operation can be performed by using Eq. (17)

(17) $$Ant_i^t=\frac{R_A^t+R_E^t}{2}$$

where R_A^t is the random walk around the ant lion selected by the roulette wheel at t^{th} iteration and R_E^t is the random walk around the elite at t^{th} iteration and Ant_i^t

indicates the position of i^{th} ant at t^{th} iteration. The ALO algorithm for solving ED problems can be described in a hierarchical manner as follows.

Step 1: Initialize the first population of ant and ant lions randomly.

Step 2: Calculate the fitness of ants and ant lions (Objective Function, Power Balance Constraint, Power Losses and Generator Rating Constraint) using equations (1)-(6).

Step 3: Find the best ant lions and assume it as the elite (best solution).

Step 4: Select an ant lion using Roulette wheel.

Step 5: Update the radius of ant's random walk using equations (13) and (14).

Step 6: Create a random walk and normalize it using equations (7) and (10).

Step 7: Update the position of ant using equation (17).

Step 8: Calculate the fitness of all ants.

Step 9: Replace an ant lion with its corresponding ant become fitter using equation (16).

Step 10: Improves the location of the ant lions if it has a better suitable function.

Step 11: Check the outage conditions.

Step 12: Stop working when getting the right answer, or go back to the first process if the answer doesn't meet the conditions.

Particle swarm optimization

Kennedy and Eberhart developed the PSO algorithm that is inspired by the social behavior of organisms. The key strategy is to study the methods of foraging for the survival of fish and birds. The experience and intellectual behavior of the herd leaders will enable the herd to survive. The same behavior can be incorporated in a group of points in the search space of an optimization problem. Populations of points in search areas were developed to mimic fish or bird flocks, with each point in the population is called particle. The algorithm mimics the swarming behavior by recognizing and improving the best position of every particle to be the best personal position and the best among all particles as the global best. Particles are repositioned by moving simultaneously in the search area. Three factors, inertia, personal best position of the particle and global best position of the group affect the direction and speed of the movement of all particles. When, the position of all particles in motion, the personal best position of individual particles and global best position of the swarm are updated. The iteration number is varied depending on the inertia. Therefore, the velocity of each particle is given by the follow:

(18) $$V_i^{t+1} = WV_i^t + c_1 * rand_1 * (pbast_i - x_i^t) + c_2 \; rand_2 * (gbast_i - x_i^t)$$

The term $rand_1 * (pbast_i - x_i^t)$ is called particle memory influence and the term $rand_2 * (gbast_i - x_i^t)$ is called swarm influence.

Where V_i^t is velocity of particle i^{th} at iteration t, c_1 and c_2 are accelerating constants, $pbast_i$ is personal best of particle i^{th}, $gbast_i$ is global best of the group, W is Inertia Weight and x_i^t is current position of particle i^{th} at iteration t.

The position of the particles is updated by the following equation:

(19) $$x_i^{t+1} = x_i^t + V_i^{t+1}$$

Inertia weight can be done by using the equation (20), where the inertia weight is increased linearly with iteration count, so that the acceleration towards optimal solution is rapid during the final stages.

(20) $$W = W_{\max} - \frac{W_{\max} - W_{\min}}{iter_{\max}} \times iter$$

$W_{\max}$ and $W_{\min}$ are inertia weight factor. $iter_{\max}$ is the maximum number of iterations allowed and $iter$ is the iteration number. The main parameters of this algorithm are the number of particles, particle dimension, particle velocity interval ($V_{\max}, V_{\min}$), $W_{\max}$ and $W_{\min}$, c_1 and c_2, particle position interval ($X_{\max}$, $X_{\min}$). In this research paper, the parameters for the PSO algorithm are shown in Table 1.

Table 1. The parameters used within PSO

Parameters	**Number**
Population size (*N*)	300
Inertia weight factor ($W_{\max}$)	0.9
Inertia weight factor ($W_{\min}$)	0.4
Acceleration constant (c_1, c_2)	1.9
Limit of change in velocity ($V_{\max}$)	0.5 $P^{\max}$
Limit of change in velocity ($V_{\min}$)	-0.5 $P^{\min}$
Rank	0.15

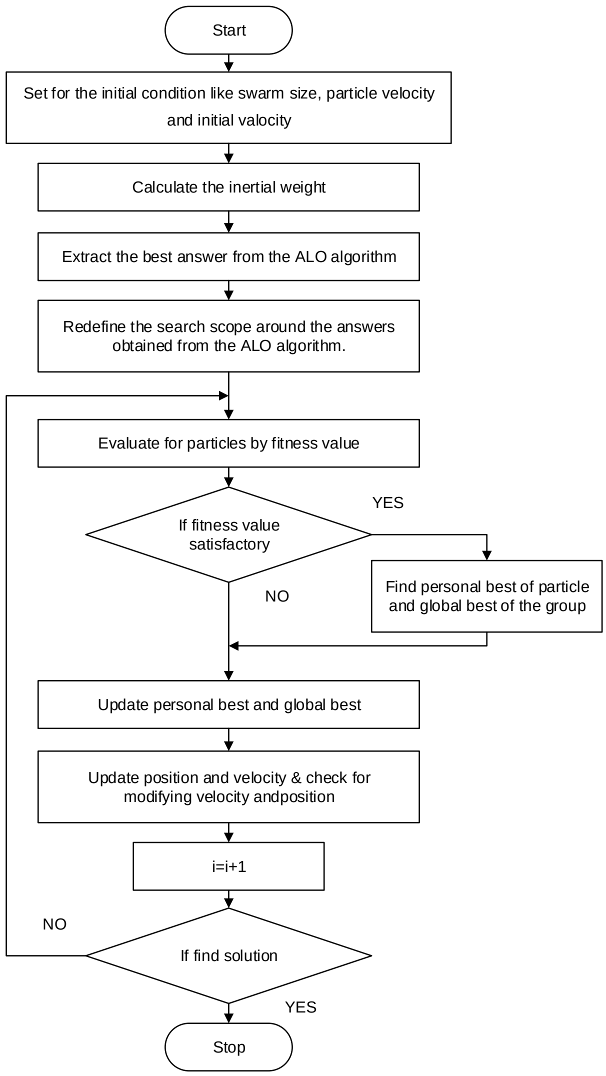

Fig. 2. Proposed HALO-PSO.

Hybrid algorithm combining ant lion optimization and particle swarm optimization

The best answer from ALO ($P_{i(ALO)}$) is used to define the initial values for HALO-PSO and redefines the search scope around $P_{i(ALO)}$. This process results in optimal problem solving and shortens the search time for answers. Figure 2 shows the working procedure.

The steps of the proposed HALO-PSO are described as follows:

Step 1: Configure the parameters of HALO-PSO as shown in Table 1.

Step 2: Calculate the inertial weight by equation (20).

Step 3: Extract the best answer from the ALO algorithm ($P_{i(ALO)}$).

Step 4: Define dimension of new initial boundary ($P_i^{\max}(new), P_i^{\min}(new)$) for the particle, population, and other individual particle variables, which are generated randomly in the permissible range of equations (21)-(22).

(21) $$P_i^{\max}(new) = P_{i(ALO)}(1+rank)$$

(22) $$P_i^{\min}(new) = P_{i(ALO)}(1-rank)$$

Where $P_i^{\max}(new)$ and $P_i^{\min}(new)$ are the new maximum and minimum power output of the i^{th} generator unit and $rank$ is a multiplier with interval [0–1]. In this article, the $rank$ is used as in Table 1. Create a population (*N*) of the output power of the i^{th} generator based on the system constraints as equation (23).

(23) $$P_i = P_i^{\min}(new) + ((P_i^{\max}(new) - P_i^{\min}(new)) \times rand(0,1))$$

Step 5: Find personal best of particle $(pbast)$ and global best of the group $(gbast)$ and update personal best and global best.

Step 6: Adjust each individual V_i speed by using equation (18).

Step 7: Edit x_i position using equation (19).

Step 8: Appraise the fitness function for the population using the objective function for the system. The best fitness value is denoted as global best $(gbast)$.

Step 9: Check the downtime conditions and increase the number of iterations. Go back to step 5 if the condition is not met and if the condition is met, stop the operation.

Case Studies

To verify the feasibility, the proposed HALO-PSO was applied to the economic dispatch problems with two different test cases. Two case studies consist of a six-unit test system and a thirteen-unit test system. Each optimization method was implemented in a MATLAB program, which runs on a TOSHIBA Satellite P745, Intel (R) Core (TM) i5, 2.30 GHz with 8 GB of RAM.

The first case study

The test system for this case consisted of six thermal units, 26 buses and 46 transmission lines, including system limits that have been set, generators rating constraints, power balance constraints and it has a power demand of 1263 MW. Table 2 shows the generator feature of each and the B-coefficient matrix was as follows [30].

$$B_{ij} = 1\times 10^{-5}\begin{bmatrix} 0.17 & 1.2 & 0.7 & -0.1 & -0.5 & -0.2 \\ 1.2 & 1.4 & 0.9 & 0.1 & -0.6 & -0.1 \\ 0.7 & 0.9 & 3.1 & 0 & -0.1 & -0.6 \\ -0.1 & 0.1 & 0.0 & 0.2 & -0.6 & -0.8 \\ -0.5 & -0.6 & -0.1 & -0.6 & 12.9 & 0.2 \\ -0.2 & -0.1 & -0.6 & -0.8 & -0.2 & 15 \end{bmatrix}$$

$$B_{0i} = 10^{-3}\times[-0.3908\ \ -0.1297\ \ 0.7047\ \ 0.0591\ \ 0.2161\ \ -0.6635]$$

$$B_{00} = 0.056$$

Table 2. Generator characteristics in case 1

Unit	a_i	b_i	c_i	e_i	f_i	$P_i^{\max}$	$P_i^{\max}$
1	0.0070	7.00	240	300	0.035	100	500
2	0.0095	10.0	200	200	0.042	50	200
3	0.0090	8.50	220	400	0.042	80	300
4	0.0090	11.0	200	159	0.063	50	150
5	0.0080	10.5	220	150	0.063	50	200
6	0.0075	12.0	190	150	0.063	50	120

The second case study

This system contained of 13 thermal generating units and the characteristic of all thermal generating units with valve point effect are given in Table 3. It is tested with 13-unit system having non-convex solution spaces and total load demands of 1800 MW, it does not include transmission loss.

Table 3. Generator characteristics in case 2

Unit	a_i	b_i	c_i	e_i	f_i	$P_i^{\min}$	$P_i^{\min}$
1	0.00028	8.10	550	300	0.035	680	0
2	0.00056	8.10	309	200	0.042	360	0
3	0.00056	8.10	307	200	0.042	360	0
4	0.00324	7.74	240	150	0.063	180	60
5	0.00324	7.74	240	150	0.063	180	60
6	0.00324	7.74	240	150	0.063	180	60
7	0.00324	7.74	240	150	0.063	180	60
8	0.00324	7.74	240	150	0.063	180	60
9	0.00324	7.74	240	150	0.063	180	60
10	0.00284	8.60	120	100	0.084	120	40
11	0.00284	8.60	120	100	0.084	120	40
12	0.00284	8.60	120	100	0.084	120	55
13	0.00284	8.60	120	100	0.084	120	55

Simulation Results

HALO-PSO was tested to assess performance with two case non-smooth cost ED problem with 6 and 13 generators. The ALO and PSO were used in comparison with different solutions obtained from random. To assess the effectiveness of each method, all search algorithms are executed at the same time interval. So, the fastest convergence would be a powerful way. To compare the effectiveness of all methods, convergence speed, elapsed time, and the results of the answers were used to evaluate performance. The configurations in Table 1 was generated iterative trials on several times, taking into account the speed of finding answers and the quality of the answers.

Simulation results in case 1

Three methods (ALO, PSO and HALO-PSO) were used to test their comparative performance in terms of the quality of the results and the speed of convergence to the answers. Each generator has a power output function as a non smooth cost function under 1236 MW of system power requirements. The methods that offer the best solution are shown in Table 4. Figure 3 shows the convergence characteristics of all methods. The results of the HALO-PSO method are compared with the IASFLA [31], GWO [32] and ICA-PSO [33] methods as shown in Table 5.

Table 4. Results of six units system in case 1

Unit	ALO	PSO	**HALO-PSO**
P_1 (MW)	446.72	451.20	**474.29**
P_2 (MW)	187.86	192.82	**155.01**
P_3 (MW)	274.69	250.57	**254.40**
P_4 (MW)	133.49	139.37	**148.84**
P_5 (MW)	158.02	155.10	**157.99**
P_6 (MW)	74.31	86.26	**83.94**
P_T (MW)	1275.09	1275.32	**1247.47**
F_T ($/h)	15433.80	15440.5	**15429.30**
P_{loss} (MW)	12.09	12.32	**11.47**

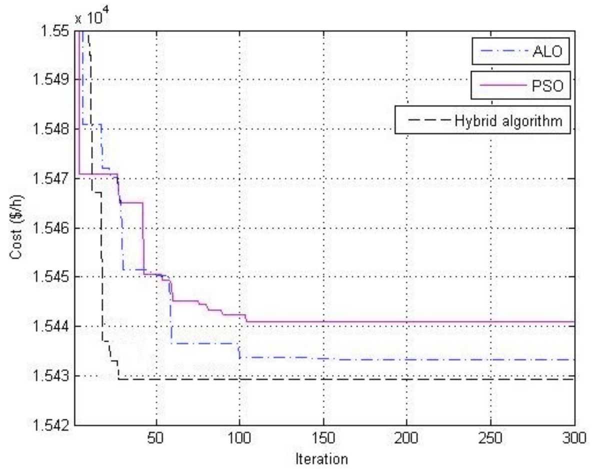

Fig.3. Convergence curve of the system in case 1

Table 5. Comparison of results with other optimization methods evaluated in case 1

Unit	IASFLA	GWO	ICA-PSO	**HALO-PSO**
P_1 (MW)	446.72	447.16	447.06	**474.29**
P_2 (MW)	175.78	173.57	173.19	**155.01**
P_3 (MW)	264.61	263.46	263.91	**254.40**
P_4 (MW)	140.29	138.37	139.02	**148.84**
P_5 (MW)	160.93	156.60	165.61	**157.99**
P_6 (MW)	87.10	87.30	86.63	**83.94**
P_T (MW)	1275.43	1275.46	1257.42	**1247.47**
F_T ($/h)	**15442.00**	**15443.00**	**15442.65**	**15429.30**
$Ploss$(MW)	12.33	12.46	12.42	**11.47**

Table 4 shows that HALO-PSO is more capable of finding optimal points in the search area compared to ALO and the PSO method. The answer given by HALO-PSO is 15429.30, which is the lowest cost value compared to the other results in the table. In Figure 3, the solution convergence ALO method is an optimal cost of 105 iterations, PSO of 100 iterations or more, while HALO-PSO converges to an answer is an optimal cost of less than 25 iteration. It can be seen that the HALO-PSO had the fastest convergence speed. Likewise, Table 5 clearly shows that the cost functions achieved by the HALO-PSO method were significantly better than those obtained by the IASFLA, GWO, and ICA-PSO methods.

Simulation results in case 2

In test system 2, there were 13 thermal generating units wich non smooth cost function and need to support a load demand of 1800 MW. The HALO-PSO results are compared with the obtained results from ALO and PSO in terms of minimum build cost and convergence speed. This example has a more complex search area compared to the previous one. The results are shown in Table 6, which satisfy the constraints of the generation units. The convergence of values for fitness functions and cost functions achieved by the HALO-PSO method when comparing ALO and PSO as shown in Figure. 4. Finally, Table 7 shows the results of the HALO-PSO method compared to the CSA [34], EBS [35] and HBCO [36] methods.

Table 6. Results of fifteen units system in case 2

Unit	ALO	PSO	**HALO-PSO**
P_1 (MW)	365.52	641.89	**537.76**
P_2 (MW)	177.85	182.09	**259.39**
P_3 (MW)	327.28	113.42	**174.65**
P_4 (MW)	107.84	80.86	**90.17**
P_5 (MW)	87.73	125.95	**126.93**
P_6 (MW)	109.55	77.45	**81.78**
P_7 (MW)	132.93	128.08	**97.93**
P_8 (MW)	119.82	104.72	**139.23**
P_9 (MW)	136.78	123.72	**84.48**
P_{10} (MW)	45.21	43.19	**42.42**
P_{11} (MW)	40.86	49.03	**46.97**
P_{12} (MW)	80.03	62.67	**60.90**
P_{13} (MW)	68.60	66.93	**57.39**
P_T (MW)	1800	1800	**1800**
F_T ($/h)	**17962.3**	**17955.9**	**17933.2**

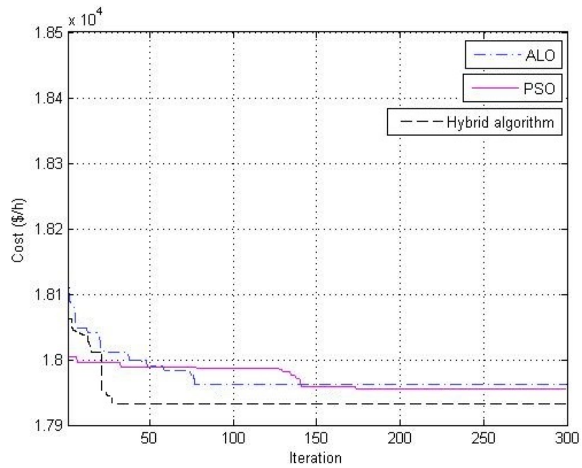

Fig.4. Convergence curve of the system in case 2

Table 7. Comparison of results with other optimization methods evaluated in case 2

Unit	CSA	EBS	HBCO	**HALO-PSO**
P_1 (MW)	369.06	628.32	502.64	**537.76**
P_2 (MW)	227.73	149.59	326.12	**259.39**
P_3 (MW)	62.18	222.73	251.77	**174.65**
P_4 (MW)	108.77	109.88	88.22	**90.17**
P_5 (MW)	107.44	60.00	88.26	**126.93**
P_6 (MW)	120.00	109.86	88.27	**81.78**
P_7 (MW)	163.74	109.87	88.24	**97.93**
P_8 (MW)	156.24	109.87	88.16	**139.23**
P_9 (MW)	138.67	109.87	88.16	**84.48**
P_{10} (MW)	108.71	40.00	40	**42.42**
P_{11} (MW)	115.76	40.00	40	**46.97**
P_{12} (MW)	62.26	55.00	40	**60.90**
P_{13} (MW)	59.35	55.00	55	**57.39**
P_T (MW)	1800	1800	1800	**1800**
F_T ($/h)	**18809**	**17963.81**	**17946.55**	**17933.2**

Table 6 shows that the HALO-PSO method has the ability to find optimal points in a larger search area than the previous case study compared to the proposed ALO and PSO methods. The HALO-PSO method gives an optimal cost value of 17933.2, which is the best in comparison with other methods. Looking at Fig. 3, the ALO method converges to the optimal cost from 80 iterations onwards, the PSO method from 140 iterations onwards, while the HALO-PSO method totals less than 35 iterations. In the same way, from Table 7, when comparing the cost functions achieved by the HALO-PSO method, it can be seen that the proposed method yields significantly better results than the CSA, EBS, and HBCO functions obtained by the CSA, EBS, and HBCO methods.

Conclusion

A new global search method was used for different ED problems within two non-smooth cost function case studies. It aims to use the good qualities of the PSO method to find the best solution and use ALO to optimize the solution. Optimization techniques include several complex methods such as initial estimation using ALO, a combination of ALO and PSO methods, and reduction of search area as HALO-PSO methods are used in the proposed method. The proposed mechanism in HALO-PSO makes the algorithm more efficient than the other recently reported algorithms. In two case studies used to find solutions to ED problems with costly non-smooth functions, it has been proven that the proposed algorithm is robust and efficient. In terms of finding high-quality solutions with convergence characteristics, stable and efficient responses from the case of 6 and 13 unit confirm that HALO-PSO is much superior to ALO and PSO methods. This method provides faster and more accurate results compared to conventional methods. In case studies, the proposed methods provide better results compared to IASFLA, GWO, ICA-PSO, CSA, EBS and HBCO methods. The numerical results show clearly that the proposed algorithm produces better results. The electrical operator can use this algorithm for optimization.

Acknowledgments

This research project is financially supported by Mahasarakham University (Fast Track 2021) and thank you Rajamangala University of Technology Lanna for supporting this creative work to be accomplished well.

***Authors**: The authors are with the Faculty of Engineering, Rajamangala University of Technology Lanna, Lampang, Thailand, and Faculty of Engineering, Mahasarakham University, Thailand.*
E-mail: Chiraphon@rmutl.ac.th; Dr. Chiraphon Takeang
E-mail: apinan.a@msu.ac.th; Assoc. Prof. Dr. Apinan Aurasopon
The correspondence address is: E-mail: apinan.a@msu.ac.th

REFERENCES

[1] R.J. Ringlee and D.D. Williams. "Economic dispatch operation considering valve throttling losses, II-distribution of system loads by the method of dynamic programming," *IEEE Trans. Power Appar. Syst.*, (1963), pp. 615-620.

[2] A. Farag, S. Al-Baiyat and T. C. Cheng, "Economic Load Dispatch Multi objective Optimization Procedures Using Linear Programming Techniques," *IEEE Transactions On Power Systems*, 10(1995), No. 2, pp. 731-738.

[3] A. A. El-Keib, H. Ma, and J.L. Hart. "Environmentally Constrained Economic Dispatch Using the LaGrangian Relaxation Method," *IEEE Transactions on Power Systems*, 9(1994), no. 4, pp. 1723-1729.

[4] Vijay Kumar Joshi, "Optimization of Economic Load Dispatch Problem by using Tabu Search Algorithm," *International Journal of Latest Trends in Engineering and Technology*, 8 (2017), Issue 4-1, pp. 182-187

[5] M. N. Nwohu and Osaremwinda Osarobo Paul, "Evaluation of Economic Load Dispatch Problem in Power Generating Stations by the Use of Ant Colony Search Algorithms," *International Journal of Research Studies in Electrical and Electronics Engineering (IJRSEEE)*, 3(2017), Issue 1, pp. 20-29.

[6] Vishnu Suresh, Przemyslaw Janik and Michal Jasinski, "Meta-heuristic approach to optimal power flow using mixed integer distributed ant colony optimization," *ARCHIVES OF ELECTRICAL ENGINEERING*, 69(2020), Issue 2, pp. 335-348.

[7] S. A. Oluwadare, G. B. Iwasokun, O. Olabode, O. Olusi, and A. E. Akinwonmi, "Genetic Algorithm-based Cost Optimization Model for Power Economic Dispatch Problem," *British Journal of Applied Science & Technology*, 15(2016), Issue 6, pp.1-10.

[8] Dexuan Zou, Steven Li, Xiangyong Kong, Haibin Ouyang and Zongyan Li. "Solving the combined heat and power economic dispatch problems by an improved genetic algorithm and a new constraint handling strategy," *Applied Energy, ELSEVIER*, 237(2019), pp.646-670.

[9] Roberto Ponciroli, Nicolas E. Stauff, Jackson Ramsey, Francesco Ganda and Richard B. Vilim, "An Improved Genetic Algorithm Approach to the Unit Commitment/Economic Dispatch Problem," *IEEE TRANSACTIONS ON POWER SYSTEMS*, 35(2020), No. 5, pp. 4005-4013.

[10] G. Mehta, R. P. Singh, and V. K. Yada, "Optimization of Combined Economic Emission Dispatch Problem using Artificial Bee Colony Method," *International Journal on Cybernetics & Informatics (IJCI)*, 6(2017), no. 1/2, pp.107-118.

[11] Rabiee A, Jamadi M, Mohammadi-Ivatloo B and Ahmadian A. "Optimal Non-Convex Combined Heat and Power Economic Dispatch via Improved Artificial Bee Colony Algorithm," *Processes*, 8(2020), Issue 9, pp. 1-22.

[12] Apinan Aurasopon and Chiraphon Takeang. "Multiple Hybrid of Lambda Iteration and Bee Colony Optimization Method for Solving Economic Dispatch Problem," *International Journal on Electrical Engineering and Informatics*, 13(2021), Number 1, pp. 57-72.

[13] Ziane Ismail, Benhamida Farid and Graa Amel, "Simulated Annealing Optimization for Generation Scheduling with Cubic Fuel Cost Function", *WSEAS Transactions on Information Science and Applications*, 14(2017), pp. 64-69.

[14] Riyadh BOUDDOU, Farid BENHAMIDA, Amine ZEGGAI, Nanda Kishore Ray, Binod Kumar Prusty and Subhransu Sekhar Dash. "The Dynamic Economic Dispatch in an Integrated Wind-Thermal Electricity Market Using Simulated Annealing Algorithm," *PRZEGLĄD ELEKTROTECHNICZNY*, 96(2020), no. 11, pp. 55-60.

[15] Vikram Kumar Kamboj, Ashutosh Bhadoria and S. K. Bath, "Solution of non-convex economic load dispatch problem for small-scale power systems using ant lion optimizer," *Neural Comput & Applic*, 28(2017), pp. 2181-2192.

[16] Faisal Z. Alazemi and Ahmed Y. Hatata, "Ant Lion Optimizer for Optimum Economic Dispatch Considering Demand Response as a Visual Power Plant," *Electric Power Components and Systems*, 47(2019), Issue 6-7, pp. 629-643.

[17] Hardiansyah Hardiansyah, "Dynamic economic emission dispatch using ant lion optimization," *Bulletin of Electrical Engineering and Informatics*, 9(2020), No. 1, pp. 12-20.

[18] Hardiansyah Hardiansyah and Junaidi Junaidi, "Multi-Objective Ant Lion Optimizer for Solving Environmental/Economic Dispatch," *PRZEGLĄD ELEKTROTECHNICZNY*, 97(2021), Issue 2, pp. 153-158.

[19] K. Selvakumar, K. Vijayakumar, D. Sattianadan, and C. S. Boopathi, "Shuffled Frog Leaping Algorithm (SFLA) for Short Term Optimal Scheduling of Thermal Units with Emission Limitation and Prohibited Operational Zone (POZ) Constraints," *Indian Journal of Science and Technology*, 9(2016), No. 42, pp. 1-6.

[20] Majid Khalili, Javad Nikoukar and Mostafa Sedighizadeh, "Combined Heat and Power Economic Dispatch using Improved Shuffled Frog Leaping Algorithm," *Journal of Advances in Computer Research*, 12(2021), No. 1, pp.1-11.

[21] M. Mohammadian, A. LorestaniM and M. Ardehali, "Optimization of single and multi-areas economic dispatch problems based on evolutionary particle swarm optimization algorithm", *Energy, Elsevier*, 161(2018), pp. 710-724.

[22] Soudamini Behera, Sasmita Behera, Ajit Kumar Barisal and Pratikhya Sahu, "Dynamic economic emission dispatch of thermal-wind-solar system with constriction factor-based particle swarm optimization algorithm", *World Journal of Engineering*, 18(2020), No. 2, pp. 217-227.

[23] Gong Wenjie, Yu Litao, Han Aoyang and Li Zhengjie. "Optimal Dispatch Model of Active Distribution Network Based on Particle Swarm Optimization Algorithm with Random Weight,"

International Conference on Big Data, Artificial Intelligence and Internet of Things Engineering (ICBAIE 2021), IEEE Xplore, April(2021), pp. 482-485.
[24] Seyedali Mirjalili, "The Ant Lion Optimizer," *Advances in Engineering Software*, 83(2015), pp. 80-98.
[25] Kennedy J, Eberhart R., "Particle swarm optimization," *IEEE International Conference Neural Networks*, 4(1995), pp. 1942–1948.
[26] Taher Niknam, "A new fuzzy adaptive hybrid particle swarm optimization algorithm for non-linear, non-smooth and non-convex economic dispatch problem," *Applied Energy*, 87(2010), pp.327–339.
[27] Taher Niknam, Hasan Doagou Mojarrad and Hamed Zeinoddini Meymand' "A novel hybrid particle swarm optimization for economic dispatch with valve-point loading effects," *Energy Conversion and Management*, 52(2011). pp. 1800–1809.
[28] Shanhe Jiang, Zhicheng Ji and Yanxia Shen, "A novel hybrid particle swarm optimization and gravitational search algorithm for solving economic emission load dispatch problems with various practical constraints," *Electrical Power and Energy Systems*, 55(2014), pp. 628–644.
[29] Mehdi Mehdinejad, Behnam Mohammadi-Ivatloo, Reza Dadashzadeh-Bonab and Kazem Zare, "Solution of optimal reactive power dispatch of power systems using hybrid particle swarm optimization and imperialist competitive algorithms," *Electrical Power and Energy Systems*, 83(2016), pp. 104–116.
[30] R. Rahmani, M. F. Othman, R. Yusof, and M. Khalid, "Solving Economic Dispatch Problem using Particle Swarm Optimization by an Evolutionary Technique for Initializing Particles", *Journal of Theoretical and Applied Information Technology*, 46(2012), No. 2, pp. 526-536.
[31] E. Bijami and M. M. Farsangi, "An Improved Adaptive Shuffled Frog Leaping Algorithm to Solve Various Non-smooth Economic Dispatch Problems in Power Systems," *Iranian Conference on Intelligent Systems (ICIS)*, (2014), pp. 1-6.
[32] Venkatakrishnan GR., Rengaraj R. and Salivahanan S., "Grey wolf optimizer to real power dispatch with non-linear constraints," *Comput. Model Eng. Sci.*, 115(2018), No.1, pp. 25–45.
[33] Jun Chen and Hashem Imani Marrani, An Efcient New Hybrid ICA‑PSO Approach for Solving Large Scale Non‑convex Multi Area Economic Dispatch Problems, "An Efcient New Hybrid ICA‑PSO Approach for Solving Large Scale Non‑convex Multi Area Economic Dispatch Problems,"
[34] S. Nagaraju, et al., "Economic Load Dispatch Considering Valve Point Loading using Cuckoo Search Algorithm," *International journal of Science & Engineering Development Research*, (2016), vol. 1, pp. 225-229.
[35] Mohammad Reza Gholami Dehbalaee, Gholam hossein Shaeisi and majid valizadeh, "A novel exclusive binary search algorithm to solve the nonlinear economic dispatch problem," *Journal of Energy Management and Technology (JEMT)*, 4(2020), Issue 3, pp. 48-56.
[36] Wanchai Khamsen, Chiraphon Takeang and Patiphat Aunban, "Hybrid method for solving the non smooth cost function economic dispatch problem," *International Journal of Electrical and Computer Engineering (IJECE)*, 10(2020), No.1, pp. 609-616.