RAIRO-Oper. Res. 54 (2020) 1269–1289
https://doi.org/10.1051/ro/2019043

RAIRO Operations Research
www.rairo-ro.org

A NON-LINEAR-THRESHOLD-ACCEPTING FUNCTION BASED ALGORITHM FOR THE SOLUTION OF ECONOMIC DISPATCH PROBLEM

Nabil Nahas[1], Mohamed Noomane Darghouth[1,*] and Mohammed Abouheaf[2,3]

Abstract. This article introduces a novel heuristic algorithm based on Non-Linear Threshold Accepting Function to solve the challenging non-convex economic dispatch problem. Economic dispatch is a power system management tool; it is used to allocate the total power generation to the generating units to meet the active load demand. The power systems are highly nonlinear due to the physical and operational constraints. The complexity of the resulting non-convex objective cost function led to inabilities to solve the problem by using analytical approaches, especially in the case of large-scale problems. Optimization techniques based on heuristics are used to overcome these difficulties. The Non-Linear Threshold Accepting Algorithm has demonstrated efficiency in solving various instances of static and dynamic allocation and scheduling problems but has never been applied to solve the economic dispatch problem. Existing benchmark systems are used to evaluate the performance of the proposed heuristic. Additional random instances with different sizes are generated to compare the adopted heuristic to the Harmony Search and the Whale Optimization Algorithms. The obtained results showed the superiority of the proposed algorithm in finding, for all considered instances, a high-quality solution in minimum computational time.

Mathematics Subject Classification. 49-02.

Received March 3, 2018. Accepted April 20, 2019.

1. Introduction

The decrease in the non-renewable energies reserves, along with the increase in the power consumption, has motivated researchers to focus on the Economic Dispatch (ED) problem. The ED problem ensures economical based allocation of the active load demand to a number of committed or online generating units while sustaining the units' physical and economic constraints [13,26]. The high non-linearity of the power systems and the massive amount of calculations needed to efficiently allocate the generating sources have made the ED problem hard to solve. The nonlinear physical constraints (spinning reserve, transmission losses, ramp rate limits, multiple fuel options, etc.) result in non-convex cost functions. The conventional approaches used to solve the ED problem are

Keywords. Economic dispatch, non-convex optimization, constraints optimization, meta-heuristics.

[1] Systems Engineering Department, King Fahd University of Petroleum & Minerals, 31261 Dhahran, Saudi Arabia.

[2] School of Electrical Engineering and Computer Science, Ottawa University, K1N 6N5 Ottawa, Canada.

[3] College of Energy Engineering, Aswan University, Aswan, Egypt.

*Corresponding author: darghouth@kfupm.edu.sa

based on Lagrangian frameworks. Kirchmayer [26] developed a solution for the ED problem based on Lagrangian and partial derivatives functions. Ramanathan [40] proposed a methodology involving emission constraints using a partially closed form technique. Madrigal and Quintana [32] proposed a mathematical approach based on duality theorems to solve the ED problem.

In practice, the ED problem is classified as a non-convex optimization problem. It has been shown that the ED problem is NP-hard even for small-size instances. Due to their robustness and feasibility, several heuristic and meta-heuristic methods are applied to solve different NP-hard optimization problems. Basically, they are inspired from biological, physical phenomena and laws of natural evaluation. The most popular algorithms are: Genetic Algorithm (GA) [20], Biogeography-Based Optimizer (BBO) [43], Simulated Annealing (SA) [27], Gravitational Local Search (GLSA) [48], Curved Space Optimization (CSO) [33], Harmony Search (HS) [12], Tabu Search (TS) [15,16], Bee Collecting Pollen Algorithm (BCPA) [30], Ant Colony Optimization [8], Particle swarm optimization (PSO) [22]. Their application to challenging engineering optimization problems becomes more and more popular, especially for large-scale instances. The majority of them showed promising solving capabilities by providing a timely manner high-quality solution.

The application of meta-heuristic algorithms to solve the ED problem was successful in terms of quality and effectiveness of the obtained solution [3,7,9,14,19,31,44]. Chen and Chang [6] suggested a lambda-based genetic algorithm to solve the ED problem for large-scale power systems. They developed a new encoding technique by considering the number of bots of each chromosome entirely independent of the number of units. Su and Lin [45] presented a new Hopfield-based approach to solve the ED problem, where the computational procedure involved a series of adjustments of the weighting factor associated with the transmission line losses. Lin *et al.* [29] developed an Improved Tabu Search (ITS) algorithm to solve the non-convex ED problem. They applied parallelism to reduce the effect of the initial conditions on the convergence rate. Gaing [10] used a Particle swarm optimization (PSO) method to solve the ED problem in power systems. Vasebi *et al.* [46] considered the combined heat and power ED problem and introduced a solving method based on a Harmony Search Algorithm (HSA). Abouheaf *et al.* [1] used an Adaptive Learning (Q-Learning) approach to solve the non-convex ED problem, where eligibility traces are used to speed up the convergence rate. Artificial Bee Colony (ABC) metaheuristic has been used in [2] to solve the ED problem with valve-point effects. The authors tested the performance of generalized forms of the algorithm on real-world industrial applications. The obtained results showed that the ability of the ABC algorithm to successfully solve the tackled problem. Khoa *et al.* [24] proposed an extension of the original single-particle mean-variance mapping optimization (MVMO) technique to solve the ED problem. The difference lies in the introduction of an initial search process with a set of particles characterized by two parameters: the scaling factor and the variable increment parameter in order to enhance the mapping procedure.

Recently, Nazari-Heris *et al.* [39] used a new meta-heuristic optimization technique called the "Whale Optimization Algorithm (WOA)" to solve the non-convex combined heat and power economic dispatch (CHPED) problem. The authors evaluated the effectiveness of the implemented method to optimally solve the problem through a set of three test systems with different combinations of power and heat units. The obtained results demonstrated the capability of the WOA to provide better solutions compared to existing solving techniques such as real-coded genetic algorithm with improved Mühlenbein mutation (RCGAIMM) [17], Group Search Optimization (GSO) and improved GSO (IGSO) [18], Greedy Particle Swarm Optimization (GPSO) and particle swarm optimization with time-varying acceleration coefficients (TVAC-PSO) [34], gray wolf optimization (GWO), and teaching–learning-based optimization (TLBO) [41].

Herein, Non-Linear Threshold Accepting Algorithm (NTAA) is proposed to solve the challenging non-convex ED problem under different operation constraints. The NTAA was recently developed by Nahas and Nourelfath [37]. It has demonstrated a higher level of efficiency in solving different bus instances of NP-hard problems for large non-linear systems. The algorithm provided excellent performance in solving a variety of difficult combinational optimization problems such as the quadratic assignment problem, dynamic discrete facility layout problems, flow-shop scheduling, and berth allocation [37,38].

A preliminary work was developed by Nahas *et al.* [36] to solve the ED problem under power balance and generation limit (capacity) constraints. The results have been compared to the Hybrid Stochastic Search

Algorithm (HSS) [4], Tabu Search algorithm (TS) [23] and Hybrid PSO-SQP [47]. The proposed approach outperformed the considered algorithms in terms of computational time and solution quality. These findings motivated us to investigate the efficiency of the NTAA to solve the non-convex ED problem compared to competitive solving techniques, especially in challenging cases of randomly large-sized instances. This will be validated against the standard benchmark systems available in the literature.

In this article, three metaheuristics (NTAA, HSA, and WOA) are developed to solve the non-convex ED problem. The performance of NTAA has been evaluated by conducting new experiments alongside the one that was introduced in [36]. The proposed NTAA is compared to the best solutions obtained by the HSA and WOA through four sets of twenty examples randomly generated (a total of 80 cases). Each set corresponds to a different number of committed units (*i.e.*, $N = 13$, 20, 40 and 60). A two-phase procedure is developed to randomly generate feasible initial solutions. In addition, the proposed algorithm is applied to two sets of numerical experiments based on existing benchmarks and compared to some solving techniques used in the literature.

The article is organized as follows. Section 2, presents the formulation of the non-convex ED problem. Section 3 introduces the proposed NTAA. The mathematical setup of the considered competitive algorithms namely HSA and WOA is provided in Section 4. A complexity analysis on the NTAA as well as HSA and WOA is performed. Section 5, evaluates the performance of the proposed algorithms using standard benchmarks and randomly generated instances with different sizes. Concluding remarks with future research extensions are presented in Section 6.

2. Problem formulation

In this section, the non-convex ED optimization problem is briefly described. The objective of the ED problem is to minimize a total power generation related cost function under multiple constraints. Generally, these constraints are related to intrinsic specifications of each generating unit and the different operation constraints (capacity, line maximum power flow, generation ramp limits, prohibited operation zones, spinning reserve, etc). In the sequel, constraints like power balance, generation limit (capacity), and valve-point loading effect are considered. The objective of the ED problem is to minimize a generation cost function such that

$$\min C_T = \min \left(\sum_{k=1}^{N} C_k \left(P_k\right) \right), \tag{2.1}$$

where C_k is the generation cost of each unit k, C_T is the total cost function, P_k is the power generated by the kth unit, and N is the total number of committed units.

The non-convex cost function $C_k\left(P_k\right)$ which describes the valve point loading effect is given by

$$C_k\left(P_k\right) = a_k + b_k P_k + c_k P_k^2 + \left| e_k \sin(f_k \sin(P_{k(\min)} - P_k)) \right|, \tag{2.2}$$

where a_k, b_k, c_k, e_k and f_k are the fuel cost coefficients of each generation unit k and $P_{k(\min)}$ is the minimum power generation capacity for unit k.

(i) *Generation capacity constraint:*

Each unit k has an output varying within the maximum and minimum generation capacities ($P_k^{\max}$ and $P_k^{\min}$) such that

$$P_k^{\min} \leq P_k \leq P_k^{\max}, \quad k \in N. \tag{2.3}$$

(ii) *Generation-demand equality constraint:*

This constraint implies a balance between the total generated power $\sum_{k=1}^{N}\left(P_k\right)$ and the total active load demand P_A plus the transmission losses P_L such that

$$\sum_{k=1}^{N}\left(P_k\right) = P_A + P_L. \tag{2.4}$$

The transmission losses P_L is given in terms of the *Kron's loss formula* such that

$$P_L = \sum_{i=1}^{N}\sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{j=1}^{N} B_{oj} P_j + B_{oo}. \tag{2.5}$$

where B_{ij}, B_{oj}, and B_{oo} are the transmission loss coefficients.

3. The Non-Linear Threshold Accepting Algorithm (NTAA)

The Non-Linear Threshold Accepting Algorithm (NTAA) was initially developed by Nahas and Nourelfath [37]. The idea behind the algorithm is to apply an accepting mechanism based on the transfer function of a low-pass RC-filter. An RC-filter is a passive filter made up of a resistor R in series with a capacitor C (see Fig. 1). The low-pass filter has a cut-off frequency $f_o = 1/(2\pi RC)$. The transfer function of the RC filter is given by

$$G(\omega) = \frac{V_{\text{out}}(\omega)}{V_{\text{in}}(\omega)}, |G(\omega)| = 1/\left(1 + \left(\frac{\omega}{\omega_0}\right)^2\right)^{1/2}, \tag{3.1}$$

where V_{out} is the output signal, V_{in} is the input signal, $\omega = 2\pi f$, and $\omega_0 = 2\pi f_0$.

Thus a flowchart inspired by this concept is introduced in Figure 2.

In NTAA, the solution is represented by a vector V of N elements. Each element $V(k)$, represents the quantity of power generated by the kth unit. To obtain a feasible solution, two main phases are proposed. In the first phase, an initial solution is randomly generated. In the case of an infeasible solution, a second phase is applied to find a feasible solution.

Steps for generating feasible initial solutions:

Step 1. Calculate the difference *diff* between the total of the quantity of power generated by all units (*i.e.*, $\sum_{i=1}^{N} M(i)$) and the total load demand (*i.e.*, $P_A + P_L$).
Step 2. If *diff* is positive, distribute the quantity *diff* equally among all units N'(the units satisfy (2.3)).
Step 3. If $N' < N$, go to *Step 1*.
Step 4. If *diff* is negative, remove the quantity $|diff|$ equally from all units N'(the units satisfy (2.3)).
Step 5. If $N' < N$, go to *Step 1*.
Step 6. End.

Detailed steps for both cases (*i.e.* $diff > 0$ and $diff < 0$) are illustrated in Figures 3 and 4.

The neighboring solution is obtained by applying a single move toward the actual solution S, as illustrated by the following steps

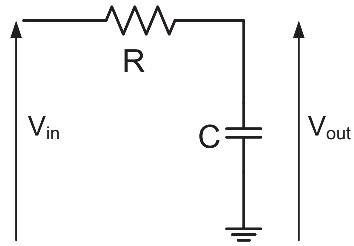

Figure 1. RC-filter.

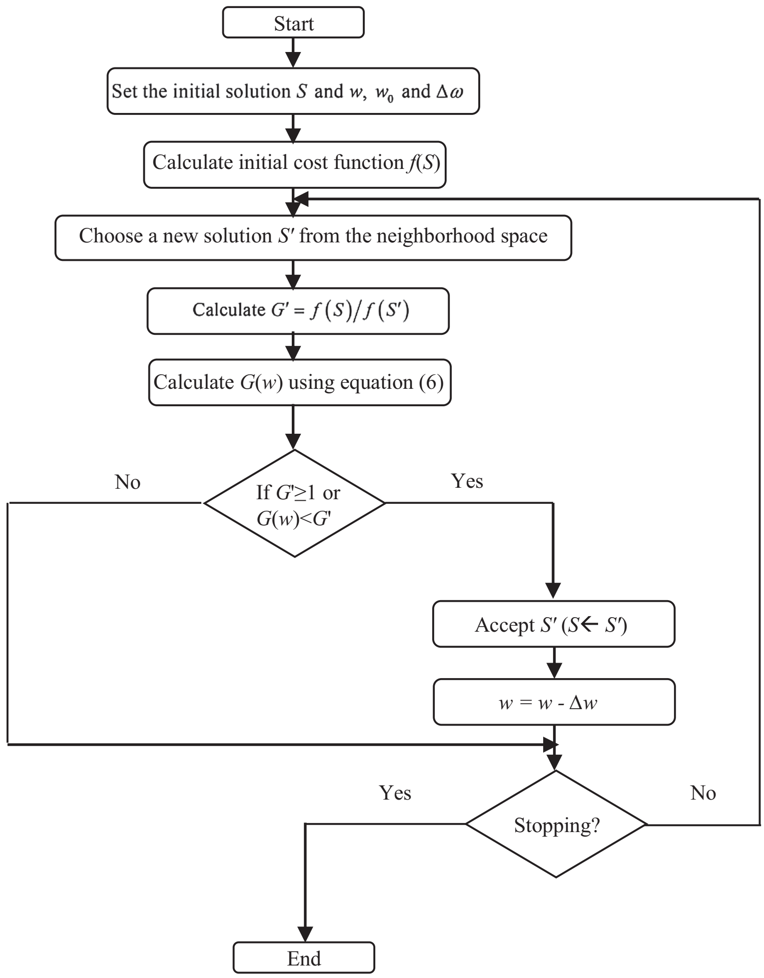

FIGURE 2. Non-linear Threshold algorithm for minimization problems.

Steps for finding the neighboring solution:

Step 1. Choose randomly two generating units g_1 and g_2.
Step 2. Calculate the difference between the upper limit of the generation capacity for unit g_1 and the actual quantity of power generated g_1 (*i.e.*, $\mathit{diff} = (P_i^{\max} - M\,(i))$).
Step 3. Set $A = \min(\mathit{diff}, M(g_2))$.
Step 4. Generate randomly a value G between 0 and A.
Step 5. Set $M(g_1) = M(g_1) + G$ and $M(g_2) = M(g_2) - G$.

4. Competitive algorithms

4.1. The Harmony Search Algorithm (HSA)

In the sequel, the Harmony Search Algorithm (HSA) is used to validate the performance of the proposed algorithm. The HSA has been developed by Geem *et al.* [12]. It was successfully used to solve the ED problem in the case of combined power and heat generating units by Vasebi *et al.* [46]. The obtained results reveal that

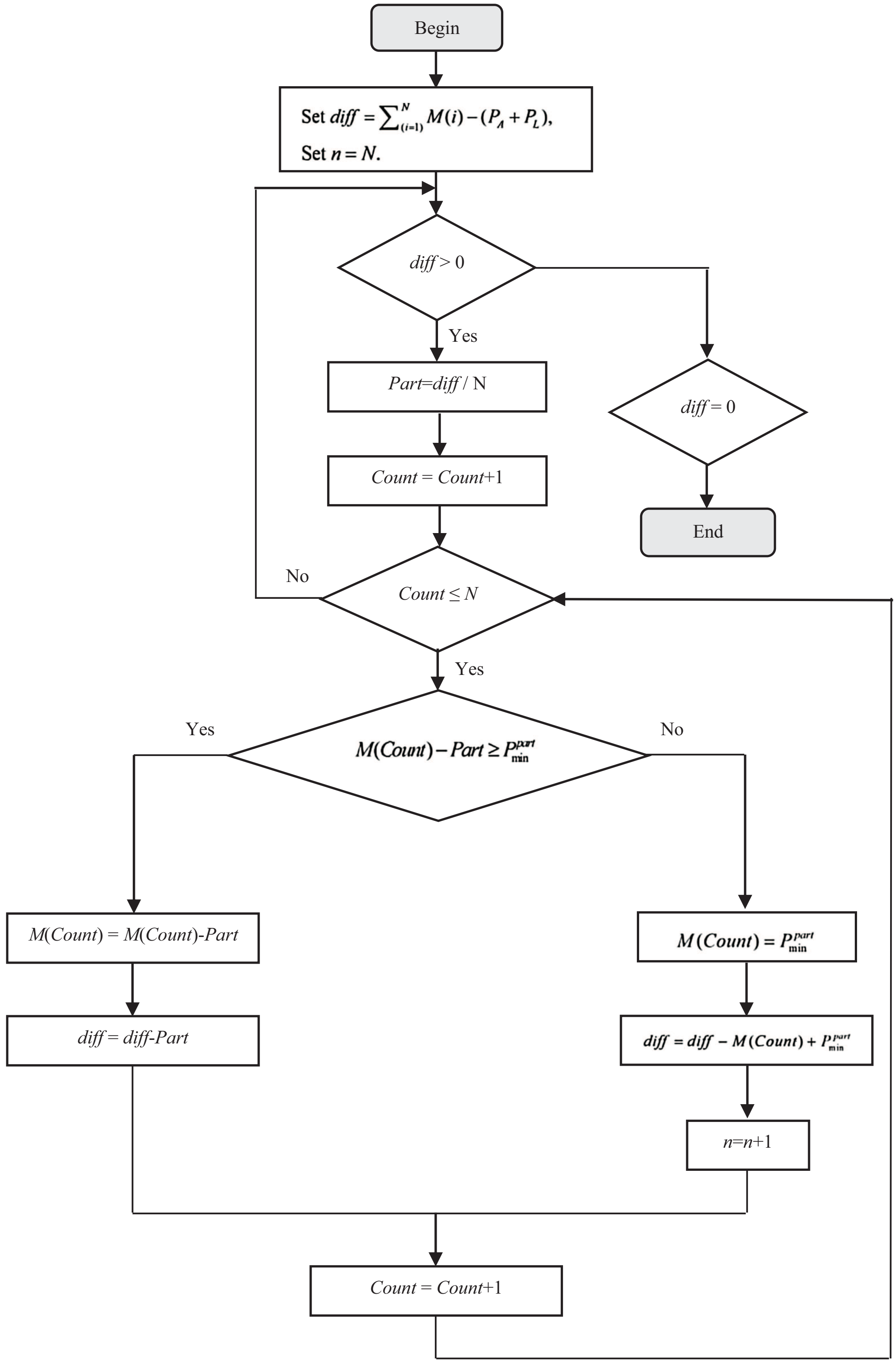

FIGURE 3. Generation of feasible solutions in case of $\sum_{(i=1)}^{N} M(i) - (P_A + P_L) > 0$.

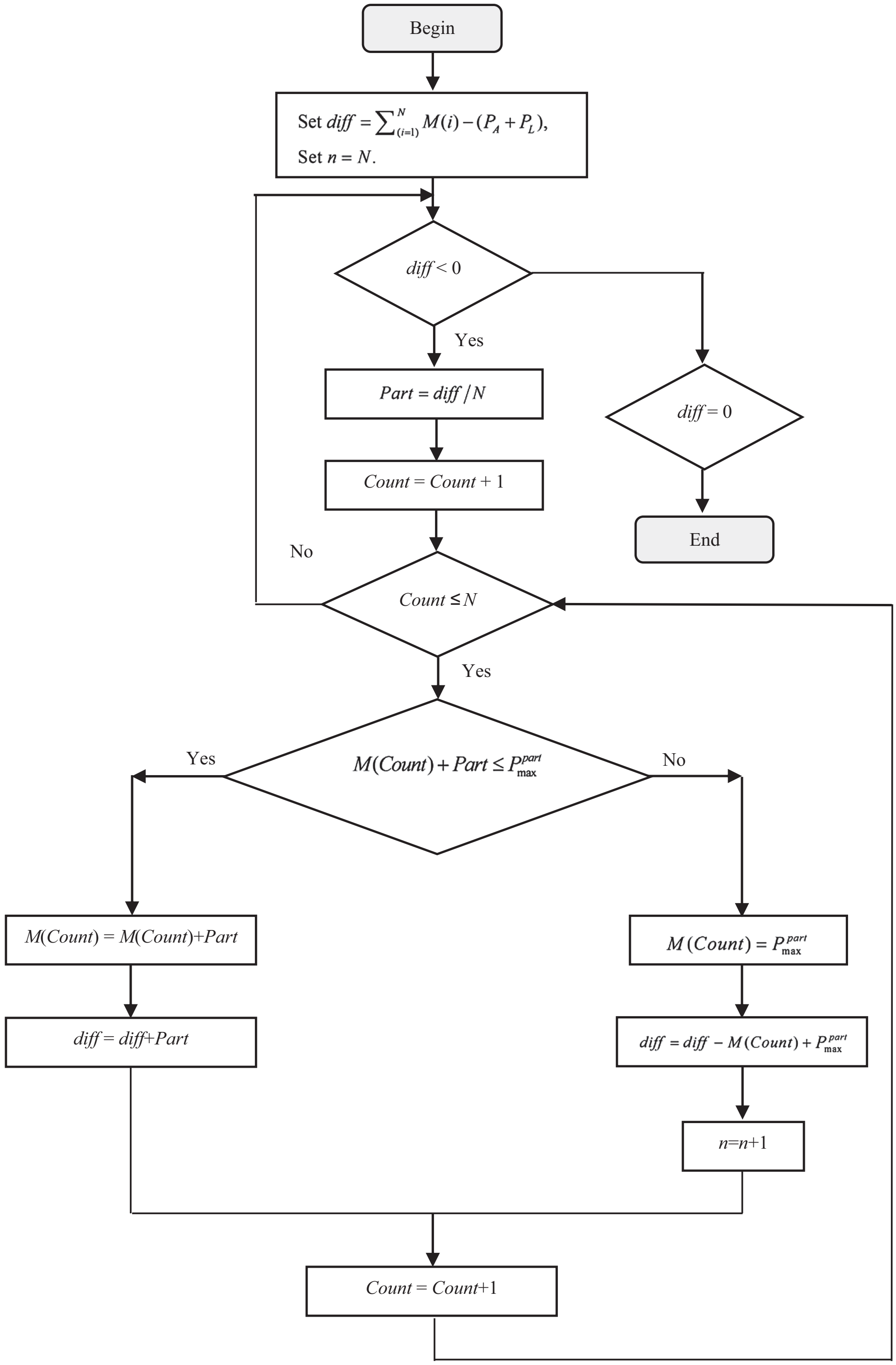

FIGURE 4. Generation of feasible solutions in case of $\sum_{(i=1)}^{N} M(i) - (P_A + P_L) < 0$.

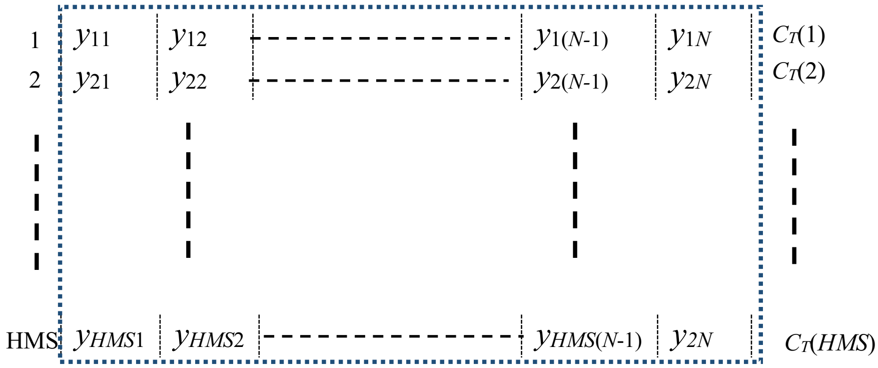

FIGURE 5. General structure of the Harmony memory (HM).

the HSA is able to produce better solutions compared to conventional methods. This makes it a good reference candidate to evaluate the efficiency of a newly proposed algorithm.

Harmony Search Algorithm:

Step 1. Initialization of input parameters: Number of generated solutions (HMS), a probability (HMCR), adjusting rate (PAR), and the maximum number of iterations ($\text{Iter}_{\max}$).

Step 2. Initialization of HMS solutions: HMS solutions are randomly generated, and for each solution $s(s = 1,\ldots,\text{HMS})$, the objective function f_i is evaluated. Figure 5 shows the general structure of the matrix HM containing a set of solutions.

Step 3. Construction of a new solution from the population: A new solution $s = (s_1,\ldots,s_n)$ is generated from the population with a probability HMCR. Each variable s_i $(i = 1,\ldots,n)$ is randomly selected from the vector $(y_{1i},\ldots,y_{\text{HMS}i})$ of HM.

Step 4. If the new solution is feasible and better than the worst solution, include the new solution in HM and exclude the worst solution from HM.

Step 5. If stopping criteria are not satisfied, go to *step 2*.

To escape local optima and improve the solutions obtained, the value of each element is modified, with a probability of a control parameter (PAR).

4.1.1. Solution representation

In the proposed algorithm, each row of the matrix HM is composed of N variables. The value of a variable $k(k = 1,\ldots,N)$ corresponds to the power generated by the kth unit. Figure 5 shows the structure of the HM matrix for the ED problem. Each variable y_{ij} $(i = 1,\ldots,\text{HMS}; j = 1,\ldots,N)$ can take any value between $p_j^{\min}$ and $p_j^{\max}$.

4.1.2. New solution construction

A new solution $S = \{s_1, s_2,\ldots,s_N\}$ is generated as follows:

- The power s_i generated by unit i $(i = 1,\ldots,N)$, is selected from the column i of HM with a probability HMCR and randomly with a probability of (1-HMCR):
- The new obtained solution is examined to determine whether it should be pitch-adjusted. This action is performed with the PAR rate using the following steps:

  Step 1. Choose randomly two generating units g_1 and g_2.

  Step 2. Calculate the difference between the upper limit of the generation capacity for unit g_1 and the actual quantity of power generated g_1 $(i.e., \mathit{diff} = (P_i^{\max} - M\,(i)))$.

Step 3. Set $A = \min(\mathit{diff}, M(g_2))$.
Step 4. Generate randomly a value G between 0 and A.
Step 5. Set $M(g_1) = M(g_1) + G$ and $M(g_2) = M(g_2) - G$.

– If the obtained solution is infeasible, the same procedure proposed for the NTAA (Sect. 3), is applied to overcome this issue.

4.2. The Whale Optimization Algorithm (WOA)

Initially, the Whale Optimization Algorithm (WOA) has been developed by Mirjalili and Lewis [35]. The proposed meta-heuristic optimization technique is inspired by the social behavior of humpback whales based on the bubble-net hunting strategy. The demonstrated efficiency of the WOA in solving large size non-convex mathematical optimization problems compared to existing state-of-the-art meta-heuristic methods encouraged several researchers to apply this algorithm to different optimization problems. The tackled optimization problems considered standard benchmark functions like unimodal, multimodal, fixed-dimension multimodal, and composite functions, as well as classical constrained engineering problems. This ensured the competitiveness behavior of the WOA against the other meta-heuristic methods.

Whale Optimization Algorithm:

The algorithm uses three operators to simulate the search for prey, encircle of the prey, and bubble-net foraging behavior of humpback whales. The procedure begins with the initialization of a set of random solutions. The positions of each search agent toward a randomly chosen search agent or the best solution obtained are updated at each iteration. This behavior is depicted by the following equations:

$$\vec{D} = \left|\vec{C} \cdot \vec{X}^*(t) - \vec{X}(t)\right|, \tag{4.1}$$

$$\vec{X}(t+1) = \vec{X}^*(t) - \vec{A} \cdot \vec{D}, \tag{4.2}$$

where t represents the current iteration. $\vec{A}$ and $\vec{C}$ stand for the coefficient vectors, $\vec{X}$ represents the position vector, and $\vec{X}^*$ its optimal value obtained so far.

The vectors $\vec{A}$ and $\vec{C}$ are obtained as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r} - \vec{a}, \tag{4.3}$$

$$\vec{C} = 2 \cdot \vec{r}, \tag{4.4}$$

where $\vec{a}$ is linearly decreased from 2 to 0 throughout the different phases of exploration and exploitation. $\vec{r}$ represents a random vector belonging to the interval $[0, 1]$.

The bubble-net behavior of the humpback whales which represents the exploitation phase is modeled using two approaches, namely shrinking encircling mechanism and the spiral updating position mechanism. The behavior of the first mechanism is achieved by decreasing the value of $\vec{a}$ introduced in equation (4.3). For a 2D space, the update of the spiral position is obtained by computing the distance between the whale located at (X, Y) and the prey located at (X^*, Y^*). Thus, a spiral equation is then created between the position of whale and prey to mimic the helix-shaped movement of humpback whales as follows:

$$\vec{X}(t+1) = \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t), \tag{4.5}$$

where $\vec{D} = \left|\vec{X}^*(t) - \vec{X}(t)\right|$ represents the distance between the current and the best solution obtained so far, b stands for a constant introduced to define the shape of the logarithmic spiral, and l is a random variable in $[-1, 1]$.

The search for the prey in the exploration phase is based on the same approach. Random values of $\vec{A}$ greater than 1 or less than -1 are considered to expand the searching space by forcing the search agent to move far away from a reference whale to perform a global search. Thus, the position of the search agent is updated according to a randomly chosen search agent instead of the best search agent found so far. This mechanism is described by the following equations:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_{rand} - \vec{X} \right|, \tag{4.6}$$

$$\vec{X}(t+1) = \vec{X}_{rand} - \vec{A} \cdot \vec{D}. \tag{4.7}$$

It's assumed that there is a random probability $p\,(p \in [0,1])$ which governs the choice between the shrinking encircling mechanism and the spiral model mechanism to update the position of whales during the optimization stage. Therefore, the vector $\vec{X}(t+1)$ is expressed as follows:

$$\vec{X}(t+1) = \begin{cases} \vec{X}^*(t) - \vec{A} \cdot \vec{D} & \text{if } p < 0.5, \\ \vec{D}' \cdot e^{bl} \cdot \cos(2\pi l) + \vec{X}^*(t) & \text{if } p \geq 0.5. \end{cases} \tag{4.8}$$

The algorithm can be summarized according to the following pseudo-code [35]:

Step 1. *Initialize the whales population* $X_i (i = 1, 2, \ldots, n)$
Step 2. *Calculate the fitness of each search agent*
Step 3. *Set* X^* *=the best search agent*
Step 4. ***while*** $(t < Iter_{\max}$ *maximum number of iterations*$)$
for *each search agent*
Update a, A, C, l, and p
if $(p < 0.5)$
if $(|A| < 1)$
Update the position of the current search agent using the equation (4.1)
else if $(|A| \geq 1)$
Select a random search agent (X_{rand})
Update the position of the current search agent by the equation (4.7)
end if
else if $(p \geq 0.5)$ *Update the position of the current search by the equation* (4.5)
end if
end for
Check if any search agent goes beyond the search space and amend it
Calculate the fitness of each search agent
Update X^* *if there is a better solution*
Set t=t+1
end while
Step 5. *return* X^*

A detailed description of the generic version of the WOA and the expression of all sub-components is provided in [35].

4.2.1. Solution representation

As in HSA, the solutions of the n agents are represented by a matrix M. Each row of the matrix M is composed of N variables and each variable y_{ij} $(i = 1, \ldots, n; j = 1, \ldots, N)$ can take any value between $p_j^{\min}$ and $p_j^{\max}$.

4.2.2. Dealing with non-feasible solutions

After updating the position of the current search, the obtained solution may become infeasible. To deal with the infeasibility issue, the procedure previously proposed in NTAA is applied (Figs. 3 and 4).

5. Complexity analysis

Each solving algorithm can be said to exhibit a *growth rate* on the order of a mathematical function relating the input size to the number of fundamental steps represented by the basic operations executed in order to reach an optimal solution. Big O notation is a commonly used measure able to characterize an algorithm complexity in the asymptotic sense. It helps in to estimate the complexity function for randomly large input. In what follows, a complexity analysis is performed to estimate the computational complexity of the proposed algorithm (NTAA) compared to other competitive algorithms namely: WOA and SA.

5.1. Non-Linear Threshold Accepting Algorithm

According to the proposed NTAA, three basic operations are executed to reach an optimal solution:

(1) Solution initialization: $O(N)$
(2) Procedure to generate a neighbor solution: $O(1)$
(3) Procedure to deal with the infeasibility issue: $O(N)$

Procedures 2 and 3 are executed $\text{Iter}_{\max}$ iterations. Therefore, the total complexity of NTAA is $(N+\text{Iter}_{\max}\times(1+N))$ or $O(N\times\text{Iter}_{\max})$.

5.2. Harmony Search Algorithm

In Harmony Search Algorithm (HSA), the basic operations are:

(1) Harmony population initialization: $O(\text{HMS}\times N)$.
(2) Procedure to generate a neighbor solution: $O(N)$.
(3) Procedure to deal with the infeasibility issue: $O(N)$.

Procedures 2 and 3 are executed $\text{Iter}_{\max}$ iterations. Therefore, the total complexity of HSA is $(\text{HMS}\times N+\text{Iter}_{\max}\times(N+N))$ or $O(N(\text{HMS}+\text{Iter}_{\max}))$ which can be simplified to $O(N\times\text{Iter}_{\max})$ because $\text{Iter}_{\max}$ is much greater than HMS.

5.3. Whale Optimization Algorithm

Four basic operations are mainly used to reach an optimal solution in Whale Optimization Algorithm (WOA):

(1) Whale population initialization: $O(n\times N)$.
(2) Procedure to update the whale parameters and the position of the agents: $O(n)$.
(3) Procedure to deal with the infeasibility issue: $O(n\times N)$.
(4) Procedure to calculate the objective function: $O(n\times N)$.

Procedures 2–4 are executed $\text{Iter}_{\max}$ iterations. Therefore, the total complexity of WOA is $(n\times N+\text{Iter}_{\max}(n+n\times N+n\times N))=(n\times N+n\times\text{Iter}_{\max}(1+2N))$ or $O(n\times N\times\text{Iter}_{\max})$.

According to the performed complexity analysis, we can conclude that both NTAA and HSA are much faster algorithms in term of executed iteration to reach an optimal solution than the WOA.

6. Numerical experiments and discussion

The algorithms were implemented using MATLAB software on a PC with a 2 GHz processor. In all the numerical experiments, the maximum number of solutions evaluated is fixed to 10^6.

6.1. Standard case studies

The performance of the proposed algorithms (NTAA, HS, and WOA) is evaluated using two benchmark cases. Preliminary numerical experiments have been conducted to tune the NTAA, HAS, and WOA parameters. After parameters fine-tuning, we found that the most appropriate values are: $1/\omega_0=0.25$, $\omega=800$ and $\Delta\omega=4\times10^{-6}$ for NTAA, HMS $=100$, HMCR $=0.97$, PAR $=0.3$ for HAS, and $n=10$ for the WOA.

TABLE 1. Parameters for the 13-unit system.

Unit	$P_{\min}$	$P_{\max}$	a	b	c	e	f
1	0	680	0.00028	8.1	550	300	0.035
2	0	360	0.00056	8.1	309	200	0.042
3	0	360	0.00056	8.1	307	200	0.042
4	60	180	0.00324	7.74	240	150	0.063
5	60	180	0.00324	7.74	240	150	0.063
6	60	180	0.00324	7.74	240	150	0.063
7	60	180	0.00324	7.74	240	150	0.063
8	60	180	0.00324	7.74	240	150	0.063
9	60	180	0.00324	7.74	240	150	0.063
10	40	120	0.00284	8.6	126	100	0.084
11	40	120	0.00284	8.6	126	100	0.084
12	55	120	0.00284	8.6	126	100	0.084
13	55	120	0.00284	8.6	126	100	0.084

TABLE 2. Comparison of NTAA and HS algorithm with the solutions found in the literature (Case 1).

Unit	HSS	TS	PSO-SQP	WOA	HS	NTAA
1	628.23	628.319	628.3205	628.318	628.318	628.3184
2	299.22	299.1993	299.0524	299.185	299.188	299.1978
3	299.17	331.8975	298.9681	299.184	299.199	299.1991
4	159.12	159.7305	159.4680	159.713	159.733	159.7330
5	159.95	159.7331	159.1429	159.720	159.733	159.7328
6	158.85	159.7306	159.2724	159.730	159.733	159.7328
7	157.26	159.7334	159.5371	159.729	159.733	159.7330
8	159.93	159.7308	158.8522	159.730	159.733	159.7329
9	159.86	159.7316	159.7845	159.726	159.733	159.7331
10	110.78	40.0028	110.9618	77.393	77.353	77.3991
11	75	77.3994	75	77.085	77.304	77.3994
12	60	92.3932	60	90.678	92.304	87.6911
13	92.62	92.3986	91.6401	89.802	97.933	92.3969
Minimum	24 275.71	24 313	24 261.05	24 170.79	24 170.08	**24 169.92**
Average	–	–	–	24 222.89	24 175.30	24 179.65
Std. Dev.	–	–	–	55.12	13.77	18.95

Case 1: 13-Unit System

In this example, 13 thermal generating units are considered, where the total load demand is 2520 MW. Table 1 shows the parameters of all considered units. The performance of the proposed algorithms (*i.e.* NTAA, HSA and WOA) is compared to those of the Hybrid Stochastic Search Algorithm (HSS) [4], the Tabu Search algorithm (TS) [23], and Hybrid PSO-SQP [47]. Table 2 shows the obtained results after 100 trials. The comparison between the results obtained by HSS, TS, and PSO-SQP and NTAA, HSA and WOA, clearly shows that HS, NTAA, and WOA outperform HSS, TS, and PSO-SQP. We can also notice that the results obtained by WOA and HSA are very close. However, the best solution has been obtained by the proposed NTAA. The convergence curves in Figure 6 show that the NTAA performs better than the HSA. According to the considered example, the convergence is faster, and the quality of the obtained solution is better.

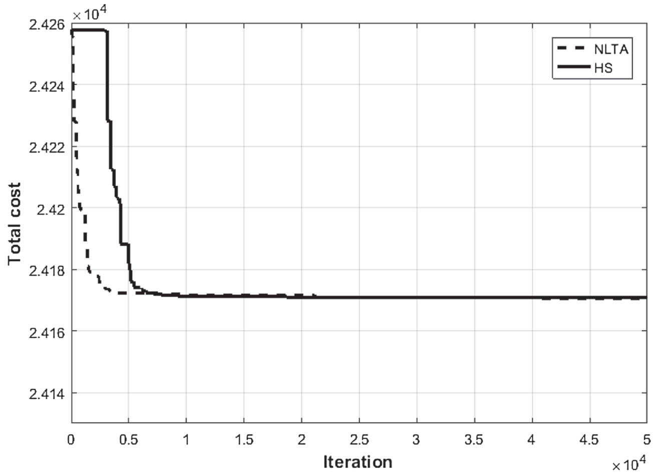

FIGURE 6. Evolution of the solution with NTAA and HS algorithms (Case 1).

TABLE 3. Comparison of NTAA and HS algorithm with the solutions found in literature (Case 2).

Algorithm	Best solution	Average	Std. Dev.
PSO	12 1664.43	12 2209.31	–
BFA	12 1415.65	–	–
BOO	12 1426.95	12 1503.33	–
NTAA	**12 1413.75**	**12 1469.36**	70.4
HS	12 1414.88	12 1483.80	33.9
WOA	12 1571.83	12 2044.29	240.3

Case 2: 40-Unit System

The second study case considers 40 generating units [42]. The total active load demand for this system is 10 500 MW. The proposed algorithms (*i.e.*, NTAA, HSA, and WOA) are compared to the best solutions found by Particle Swarm Optimization (PSO) [42], the Bacterial Foraging Algorithm (BFA) [21] and the Biogeography-Based Optimization algorithm (BBO) [5]. Table 3 shows the obtained results.

From Table 3, it can be seen that the NTAA produced a better solution compared to the other considered algorithms, followed by the HSA. The best result obtained by NTAA is summarized in Table 4. The average computation time, for this case, did not exceed 50 s. Figure 7 shows the convergence curves of the proposed algorithms. We can notice that the quality of the solution obtained by NTAA is better than that obtained by HSA.

6.2. NTAA robustness

This experiment is performed to illustrate the robustness of the proposed NTAA. The proposed algorithm is compared to HSA as well as WOA through four sets of 20 examples (a total of 80 cases). Each set corresponds to a different number of committed units ($N = 13$, 20, 40 and 60). All parameters ($P_{\min}$, $P_{\max}, a, b, c, e, f$, and

TABLE 4. Best solution obtained by NTAA over 10 runs (Case 2).

Unit	NTAA	Unit	NTAA
1	110.7998	21	523.2790
2	110.7995	22	523.2788
3	97.3990	23	523.2793
4	179.7331	24	523.2793
5	87.7998	25	523.2798
6	140	26	523.2793
7	259.6017	27	10
8	284.5995	28	10
9	284.6013	29	10
10	130	30	87.8004
11	94	31	190
12	94	32	190.000
13	214.7584	33	190.000
14	394.2779	34	164.7997
15	394.2784	35	200
16	394.2768	36	194.4996
17	489.2793	37	110
18	489.2793	38	110.0000
19	511.2792	39	110.0000
20	511.2783	40	511.2789
Total cost	**121 413.75**		

TABLE 5. Comparison results for problems with 13 committed units.

	HS			WOA			NTAA		
Instance	Min	Av.	St.d	Min	Av.	St.d	Min	Av.	St.d
1	32 216.062	**32 240.861**	20.48	32 222.214	32 318.744	69.01	**32 215.932**	32 280.036	53.89
2	34 085.599	**34 105.600**	26.04	34 126.305	34 218.205	69.73	**34 085.454**	34 106.520	16.54
3	37 032.205	**37 055.417**	21.07	37 036.440	37 152.315	68.60	**37 031.971**	37 098.185	55.16
4	31 507.342	31 576.875	45.01	31 534.506	31 717.227	86.25	**31 506.965**	**31 570.919**	34.11
5	33 885.891	33 937.916	29.87	33 920.459	34 067.576	77.57	**33 882.671**	**33 930.294**	42.72
6	33 585.326	33 725.930	54.66	33 566.050	33 747.124	107.56	**33 563.734**	**33 691.745**	70.33
7	35 653.719	**35 733.491**	47.06	35 664.163	35 782.651	94.42	**35 653.701**	35 746.374	65.00
8	33 771.904	**33 792.004**	14.86	33 773.446	338 870.902	60.03	**33 770.253**	33 810.096	33.49
9	**29 692.912**	**29 700.657**	18.35	29 700.234	29 762.663	60.68	29 692.925	29 720.941	34.21
10	**37 142.930**	37 174.506	22.72	37 175.518	37 253.228	69.88	37 142.940	**37 171.454**	23.18
11	38 052.169	38 141.256	48.86	38 109.711	38 256.039	79.91	**38 051.841**	**38 105.273**	32.79
12	35 741.037	35 786.237	12.64	35 742.77	35 872.422	64.03	**35 724.320**	**35 764.920**	32.95
13	34 699.252	34 749.099	24.67	34 701.168	34 769.335	54.37	**34 699.251**	**34 739.126**	42.43
14	42 053.991	42 204.351	47.42	42 034.707	42 171.083	83.50	**42 028.734**	**42 142.381**	86.03
15	40 750.728	40 792.093	29.14	40 753.368	40 802.524	48.85	**40 750.724**	**40 789.428**	26.31
16	32 538.823	32 544.599	6.23	32 546.223	32 615.121	61.88	**32 538.812**	**32 543.562**	12.22
17	35 403.220	**35 449.594**	41.43	35 404.636	35 458.187	62.69	**35 403.199**	35 457.131	44.87
18	34 979.423	**35 043.818**	33.37	34 989.152	35 092.073	60.62	**34 977.694**	35 064.520	44.06
19	33 676.422	33 735.926	34.35	337 347.397	33 844.468	69.22	**33 676.315**	**33 731.791**	38.44
20	34 553.319	34 637.113	75.41	34 557.813	34 723.575	82.43	**34 553.185**	**34 611.916**	52.10

Table 6. Comparison results for problems with 20 committed units.

Instance	HS			WOA			NTAA		
	Min	Av.	St.d	Min	Av.	St.d	Min	Av.	St.d
1	**45 473.011**	45 503.337	26.56	45 499.608	45 754.211	154.44	45 472.836	**45 543.077**	65.72
2	49 511.680	**49 543.244**	31.17	49 553.955	49 801.125	146.86	**49 511.572**	49 573.065	37.60
3	57 282.831	57 345.406	39.75	57 346.864	57 759.515	198.40	**57 252.888**	**57 329.119**	37.50
4	43 277.430	**43 310.682**	27.03	43 360.282	43 644.268	166.10	**43 277.204**	43 361.473	58.59
5	49 131.112	**49 191.592**	41.09	49 203.960	49 450.225	128.33	**49 120.941**	49 218.672	38.98
6	**49 417.372**	49 530.158	61.67	49 479.994	49 755.209	154.85	49 431.689	**49 518.966**	67.22
7	54 193.538	54 235.769	30.14	54 240.489	54 437.684	123.48	54 198.412	54 279.546	44.38
8	51 232.627	51 318.536	45.82	51 427.483	51 682.483	157.67	**51 230.563**	51 336.025	76.90
9	52 110.510	**52 152.694**	26.49	52 180.063	52 424.941	126.96	**52 110.233**	52 185.188	49.79
10	**54 781.641**	**54 836.466**	45.39	54 816.188	55 068.671	133.21	54 789.448	54 868.559	52.02
11	53 305.322	53 348.746	33.96	53 351.579	53 604.357	131.85	**53 304.523**	**53 332.752**	29.39
12	**62 171.398**	**62 213.611**	27.09	62 200.047	62 356.244	94.41	62 177.109	62 238.499	34.66
13	50 835.607	50 891.855	40.78	50 855.701	51 118.152	141.61	**50 818.873**	**50 872.010**	43.05
14	53 533.481	53 629.965	56.31	53 646.199	53 965.166	158.70	**53 533.273**	**53 629.845**	61.80
15	52 383.154	**52 423.492**	25.61	52 402.618	52 631.916	120.59	**52 382.855**	52 445.461	44.52
16	51 553.786	**51 603.981**	24.75	51 615.427	51 749.652	83.10	**51 548.332**	51 610.167	29.28
17	51 158.069	51 200.629	26.50	51 355.482	51 588.751	122.14	**51 153.623**	**51 196.123**	31.44
18	48 954.456	49 045.394	45.46	49 072.248	49 299.351	117.59	**48 954.144**	**49 042.548**	65.46
19	44 669.189	44 769.800	56.04	44 797.844	45 086.064	134.70	**44 650.508**	**44 739.364**	55.93
20	62 327.838	62 373.095	24.19	62 369.485	62 726.552	156.14	**62 319.761**	**62 371.124**	27.92

Table 7. Comparison results for problems with 40 committed units.

Instance	HS			WOA			NTAA		
	Min	Av.	St.d	Min	Av.	St.d	Min	Av.	St.d
1	103 038.128	**103 097.892**	36.8	103 554.713	104 240.923	313.23	**103 038.029**	103 098.660	44.60
2	**107 434.866**	**107 485.488**	41.8	107 815.070	108 452.458	261.11	107 435.029	107 505.950	65.92
3	108 059.740	108 112.711	45.0	108 535.722	109 156.526	245.32	**108 040.848**	**108 079.570**	23.71
4	**101 351.642**	**101 434.918**	43.2	102 178.417	102 743.000	289.54	101 353.142	101 455.018	52.88
5	108 403.825	108 453.166	32.0	108 842.095	109 442.285	257.23	**108 370.841**	**108 420.091**	27.37
6	**105 492.029**	**105 581.421**	47.6	105 978.722	106 482.030	224.26	105 504.982	105 587.389	54.75
7	**98 667.364**	**98 763.686**	45.2	98 998.880	99 530.833	227.23	98 682.492	98 765.851	45.11
8	**95 597.641**	**95 639.505**	31.2	96 315.384	96 845.372	248.05	95 604.106	95 653.439	29.30
9	**111 884.242**	**111 963.128**	56.0	112 288.193	112 867.094	291.26	111 894.751	112 076.946	87.98
10	98 752.567	**98 802.292**	29.9	99 341.181	99 979.992	275.03	**98 751.708**	98 812.583	35.32
11	104 198.503	104 244.967	27.9	104 571.692	105 301.470	340.46	**104 191.976**	**104 241.968**	24.55
12	**96 638.322**	**96 689.019**	25.9	97 024.417	97 714.530	267.22	96 644.666	96 696.078	34.54
13	96 670.137	96 783.109	53.2	97 227.180	97 775.241	255.76	**96 668.997**	**96 768.486**	58.40
14	95 636.854	95 775.240	67.9	96 406.231	96 974.135	323.54	**95 621.900**	**95 746.512**	75.21
15	**99 873.937**	**99 936.546**	38.3	100 513.994	101 291.268	344.05	99 881.778	99 942.525	40.17
16	106 547.428	106 629.736	51.2	107 249.980	107 720.768	273.47	**106 546.276**	**106 621.348**	47.35
17	112 189.605	112 327.625	62.0	112 954.597	113 479.016	287.13	**112 155.060**	**112 261.186**	68.27
18	91 438.024	91 575.935	63.2	92 424.343	92 935.283	311.69	**91 394.134**	**91 526.157**	72.32
19	101 367.193	101 430.029	39.3	101 803.939	102 357.738	242.52	**101 356.928**	**101 407.572**	31.43
20	115 587.727	115 683.169	58.9	115 837.214	116 463.020	275.06	**115 579.941**	**115 651.357**	41.63

Table 8. Comparison results for problems with 60 committed units.

	HS			WOA			NTAA		
Instance	Min	Av.	St.d	Min	Av.	St.d	Min	Av.	St.d
1	159 058.040	159 258.443	103.8	160 376.053	161 320.086	395.77	**159 030.017**	**159 085.579**	34.47
2	166 988.657	167 072.368	56.0	168 340.770	169 332.803	450.16	**166 983.588**	**167 048.189**	33.44
3	**154 704.071**	**154 805.385**	55.2	156 314.358	157 270.183	411.11	154 737.260	154 831.514	48.09
4	173 181.800	173 361.950	97.8	174 601.453	175 277.589	481.27	**173 165.442**	**173 289.964**	81.80
5	**138 419.302**	138 490.622	37.1	139 794.882	140 738.966	416.72	138 419.852	**138 480.948**	33.17
6	**146 340.777**	146 461.146	59.3	147 973.624	149 121.424	472.51	146 345.366	**146 452.595**	58.66
7	156 784.185	156 882.159	47.2	158 075.421	159 049.754	354.69	**156 783.195**	**156 845.344**	33.23
8	142 266.232	142 337.105	53.9	143 541.652	144 484.856	393.75	**142 262.558**	**142 303.730**	35.42
9	160 069.099	160 174.159	53.1	161 850.923	162 704.157	385.91	**160 034.278**	**160 092.416**	28.48
10	149 718.719	149 822.871	61.8	151 457.356	152 652.161	493.25	**149 714.841**	**149 780.530**	44.28
11	161 155.818	161 308.461	77.0	163 201.473	164 199.963	448.25	**161 145.313**	**161 207.535**	39.92
12	134 983.959	135 103.838	58.2	136 898.690	137 698.847	373.65	**134 933.866**	**134 995.472**	46.98
13	143 140.223	143 214.556	44.5	144 718.086	145 788.325	429.92	**143 130.905**	**143 196.634**	56.57
14	150 249.160	150 342.229	47.1	151 950.494	152 811.099	347.24	**150 233.278**	**150 287.898**	30.44
15	159 129.653	159 112.606	53.7	160 088.783	160 847.371	330.58	**158 988.123**	**159 054.601**	42.87
16	163 343.860	163 490.457	70.9	165 129.987	165 926.348	397.62	**163 332.185**	**163 427.034**	58.73
17	**155 807.637**	155 917.182	44.0	157 543.975	158 293.804	372.98	155 813.182	**155 860.161**	21.75
18	161 094.474	161 325.405	88.2	162 386.425	163 110.532	350.98	**161 038.494**	**161 224.784**	99.24
19	172 400.835	172 562.651	82.3	173 650.873	174 373.452	310.61	**172 395.237**	**172 460.358**	60.85
20	163 167.248	163 270.749	64.8	164 327.945	165 204.733	348.47	**163 125.292**	**163 205.741**	48.23

L) of these instances are randomly generated as follows:

$$\begin{aligned}
&P^i_{\min} \in [0, 60], && \forall i = 1, \ldots, N\\
&P^i_{\max} \in [120, 600], && \forall i = 1, \ldots, N\\
&a_i = 0.0002 + rand_3 \times 0.0002, && \forall i = 1, \ldots, N\\
&b_i = 6 + \text{round}\left[rand_2 * 3\right], && \forall i = 1, \ldots, N\\
&c_i = \text{round}\left[0.80 \times P^i_{\max}\right] + \text{round}\left[rand_1 \times \left(0.3 \times P^i_{\max}\right)\right], && \forall i = 1, \ldots, N\\
&e_i = 100 + \text{round}\left[rand_3 \times 200\right], && \forall i = 1, \ldots, N\\
&f_i = 0.03 + rand_4 \times 0.06, && \forall i = 1, \ldots, N\\
&L \in \left[0.75 * \sum_{i=1}^{N} P^i_{\max}, 0.95 * \sum_{i=1}^{N} P^i_{\max}\right]
\end{aligned}$$

where round (.) returns the nearest integer to the real number and $rand_1$, $rand_2$, $rand_3$, and $rand_4$ are random numbers generated from the uniform distribution on the interval [0, 1].

We used the same parameters of test instances used in case 1. By running the algorithms without further tuning on the 80 test instances, we avoid any parameters over-fitting. The proposed algorithms are evaluated in terms of solution quality. For each instance, 100 trials are performed.

Tables 5–8 show the best, the average, and the standard deviation results obtained by the NTAA, the HSA, and the WOA. It can be observed that:

- For the 13-units system, the NTAA achieved the best results in 19 out of the 20 test cases.
- For the 20-units system, the NTAA achieved the best results in 15 out of the 20 test cases.
- For systems composed of 40 and 60 units, the NTAA outperformed the HSA and the WOA in 12 and 15 cases respectively.
- The computational time of performing the NTAA is low compared to that of performing the HSA. It did not exceed 15 s for large instances.

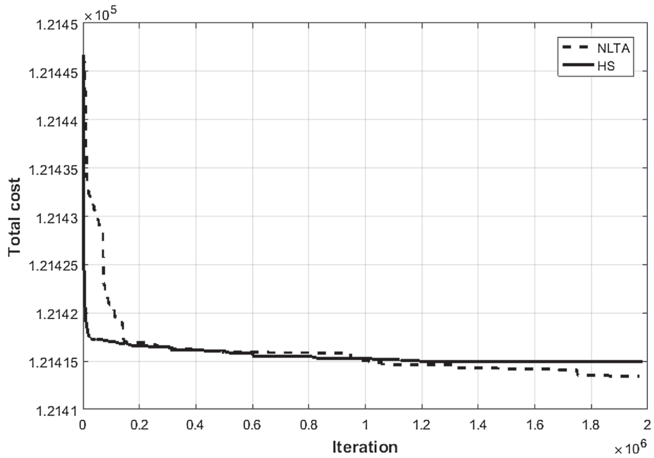

FIGURE 7. Evolution of the solution with NTAA and HS algorithms (Case 2).

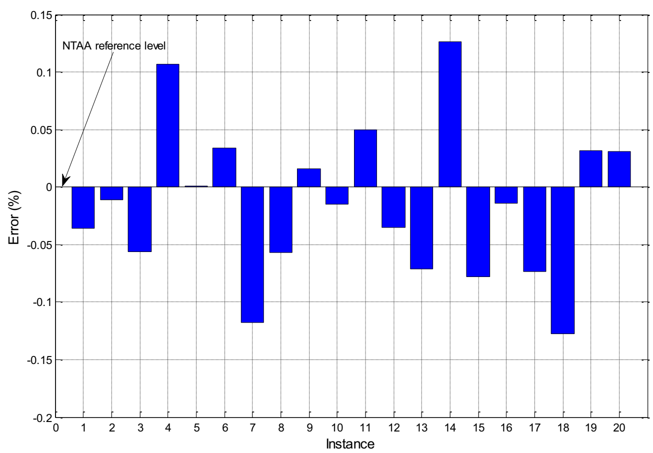

FIGURE 8. Comparing HS and NTAA algorithms for $N = 13$.

– For all the considered instances with different sizes, the WOA produced low-quality solutions compared to the NTAA and HSA.
– The standard deviation over 100 runs of the NTAA and HSA, is relatively low compared to WOA for all the test instances as shown in Tables 5–8. The average objective function values for each instance are very close to the best solutions, especially for the NTAA. This result demonstrates the robustness of the proposed NTAA.

It has been observed for these randomly generated instances that the NTAA outperforms the Harmony Search Algorithm (HSA) as well as the Whale Optimization Algorithm (WOA) in term of the mean value of the total cost. The HSA provided competitive results when compared to WOA.

Therefore, let's consider NTAA as a reference level and let us define the average percentage errors. The average percentage error is based on the relative difference between the reference level (NTAA) and the most

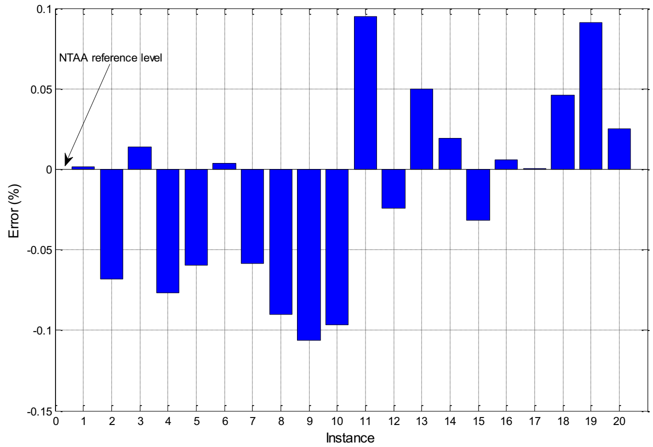

FIGURE 9. Comparing HS and NTAA algorithms for $N = 20$.

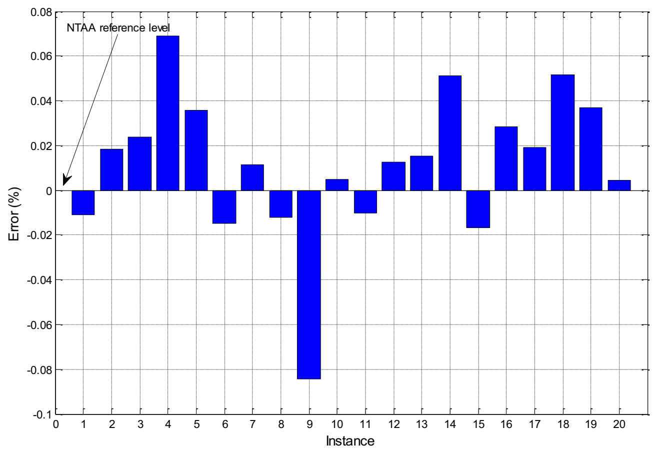

FIGURE 10. Comparing HS and NTAA algorithms for $N = 40$.

promising algorithm (HSA) and can be expressed as follows:

$$\text{Average error (\%)} = \frac{\text{Mean (Total cost HSA)} - \text{Mean (Total cost NTAA)}}{\text{Mean (Total cost NTAA)}} \times 100, \tag{6.1}$$

where Mean (Total cost NTAA) is the mean total cost value obtained by the NTAA and Mean (Total cost HSA) is the mean total cost value obtained by the HSA.

Figures 8–11 show the average errors obtained by the HSA relative to NTAA. We can notice that the NTAA outperforms the HSA, especially for large size instances ($N = 20$, $N = 40$, and $N = 60$).

7. Conclusion

In this article, a Non-Linear-Threshold-Accepting function based algorithm has been introduced to solve the non-convex economic dispatch problem subject to generation and active load demand constraints. The considered acceptance rule is inspired by the low-pass RC-filter frequently used to reduce the amplitude of signals with

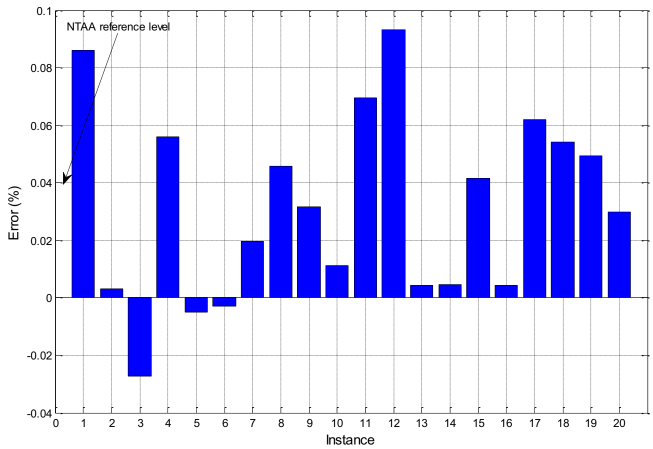

FIGURE 11. Comparing HS and NTAA algorithms for $N = 60$.

higher frequencies. A two-phase procedure is developed to randomly generate feasible initial solutions. The introduced Non-Linear Threshold Accepting Algorithm is applied to two sets of numerical experiments based on existing standard benchmarks. The obtained results are compared with several efficient solving techniques, namely Hybrid Stochastic Search, Tabu Search, and Hybrid PSO-SQP algorithms. Also, the proposed procedure is compared to the Harmony Search Algorithm and the Whale Optimization Algorithm for various sets of randomly generated instances with different numbers of committed units. The obtained results revealed the superiority of the proposed NTAA in finding, for all considered instances, a high-quality solution in lower computational time, outperforming other algorithms in the majority of the cases, even for large-sized instances problem.

Several future research extensions can be investigated. One extension is to explore the efficiency of the NTAA in solving the non-convex ED problem where multiple-fuel options are considered. Another extension is to investigate the robustness of the proposed algorithm in the case where the reliability of each power generation unit is an optimization factor.

Acknowledgements. This work was supported by the Deanship of Scientific Research (DSR) at King Fahd University of Petroleum & Minerals (KFUPM) under Grant No. IN141043.

REFERENCES

[1] M. Abouheaf, S. Haesart, W. Lee and F. Lewis, Q-Learning with eligibility traces to solve non-convex economic dispatch problems. *Int. J. Electr. Sci. Eng.* **7** (2012) 1390–1396.

[2] D. Aydin, G. Yavuz, S. Ozyön, C. Yaşar, and T. Stützle, Artificial bee colony framework to non-convex economic dispatch problem with valve-point effects: a case study. In: *GECCO 2017 – Proceedings of the Genetic and Evolutionary Computation Conference Companion* (2017) 1311–1318.

[3] M. Basu, Modified particle swarm optimization for nonconvex economic dispatch problems. *Int. J. Electr. Power Energy Syst.* **69** (2015) 304–310.

[4] D. Bhagwan and C. Patvardhan, Solution of economic load dispatch using real coded hybrid stochastic search. *Electr. Power Energy Syst.* **21** (1999) 165–170.

[5] A. Bhattacharya and P. Chattopadhyay, Biogeography-based optimization for different economic load dispatch problems. *IEEE Trans. Power Syst.* **25** (2010) 1064–77.

[6] P.-H. Chen and H.-C. Chang, Large-scale economic dispatch by genetic algorithm. *IEEE Trans. Power Syst.* **10** (1995) 1919–1926.

[7] S. Devi and M. Geethanjali, Optimum location and sizing of distribution static synchronous series compensator using particle swarm optimization. *Int. J. Electr. Power Energy Syst.* **62** (2014) 646–53.

[8] M. Dorigo, M. Birattari and T. Stutzle, Ant colony optimization. *IEEE Comput. Intell.* **1** (2006) 28–39.

[9] E. Elattar, A hybrid genetic algorithm and bacterial foraging approach for dynamic economic dispatch problem. *Int. J. Electr. Power Energy Syst.* **69** (2015) 18–26.

[10] Z.L. Gaing, Particle swarm optimization to solving the economic dispatch considering the generator constraints. *IEEE Trans. Power Syst.* **18** (2003) 1187–1195.

[11] Z.W. Geem, Harmony search optimisation to the pump-included water distribution network design. *Civil Eng. Environ. Syst.* **26** (2009) 211–221.

[12] Z.W. Geem, J.H. Kim and G.V. Loganathan, A new heuristic optimization algorithm: harmony search. *SIMULATION* **76** (2001) 60–68.

[13] E.E. George, Intrasystem transmission losses. *AIEE Trans.* **62** (1943) 153–158.

[14] A. Ghasemi, M. Gheydi, M.J. Golkar and M. Eslami, Modeling of Wind/Environment/Economic Dispatch in power system and solving via an online learning meta-heuristic method. *Appl. Soft Comput.* **43** (2016) 454–468.

[15] F. Glover, Tabu search – Part I. *ORSA J. Comput.* **1** (1989) 190–206.

[16] F. Glover, Tabu search – Part II. *ORSA J. Comput.* **2** (1990) 4–32.

[17] A. Haghrah, M. Nazari-Heris and B. Mohammadi-ivatloo, Solving combined heat and power economic dispatch problem using real coded genetic algorithm with improved Mühlenbein mutation. *Appl. Therm. Eng.* **99** (2016) 465–475.

[18] M.T. Hagh, S. Teimourzadeh, M. Alipour and P. Aliasghary, Improved group search optimization method for solving CHPED in large scale power systems. *Energy Convers. Manage.* **80** (2014) 446–456.

[19] H. Hamedi, Solving the combined economic load and emission dispatch problems using new heuristic algorithm. *Int. J. Electr. Power Energy Syst.* **46** (2013) 10–16.

[20] J.H. Holland, Genetic algorithms. *Sci. Am.* **267** (1992) 66–72.

[21] P. Hota, A. Barisal and R. Chakrabarti, Economic emission load dispatch through fuzzy based bacterial foraging algorithm. *Int. J. Electr. Power Energy Syst.* **32** (2010) 794–803.

[22] J. Kennedy and R. Eberhart, Particle swarm optimization. In: *Proceedings of the 1995 IEEE International Conference on Neural Networks* (1995) 1942–1948.

[23] S. Khamsawang, C. Boonseng and S. Pothiya, Solving the economic dispatch problem with Tabu search algorithm. *Proc. IEEE Int. Conf. Ind. Technol.* **1** (2002) 274–278.

[24] T.H. Khoa, P.M. Vasant, M.S.B. Singh and V.N. Dieu, Swarm based mean-variance mapping optimization for convex and non-convex economic dispatch problems. *Memetic Comput.* **9** (2017) 91–108.

[25] L.K. Kirchmayer, Economic Control of Interconnected Systems. Wiley, New York, NY (1959).

[26] L.K. Kirchmayer, Economic Operation of Power Systems. Wiley, New York, NY (1993).

[27] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi, Optimization by simmulated annealing. *Science* **220** (1983) 671–680.

[28] K.S. Lee and Z.W. Geem, A new structural optimization method based on the harmony search algorithm. *Comput. Struct.* **82** (2004) 781–798.

[29] W.-M. Lin, F.-S. Cheng and M.-T. Tsay, An improved tabu search for economic dispatch with multiple minima. *IEEE Trans. Power Syst.* **17** (2002) 108–112.

[30] X. Lu and Y. Zhou, A novel global convergence algorithm: bee collecting Pollen algorithm. In: Advanced Intelligent Computing Theories and Applications. With Aspects of Articial Intelligence. ICIC 2008. Edited by D.S. Huang, D.C. Wunsch, D.S. Levine and K.H. Jo. In Vol. 5227 of *Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg (2008)

[31] P. Lu, J. Zhou, H. Zhang, R. Zhang and C. Wang, Chaotic differential bee colony optimization algorithm for dynamic economic dispatch problem with valve-point effects. *Int. J. Electr. Power Energy Syst.* **62** (2014) 130–143.

[32] M. Madrigal and V.H. Quintana, An analytical solution to the economic dispatch problem. *IEEE Power Eng. Rev.* **20** (2000) 52e5.

[33] F.F. Moghaddam, R.F. Moghaddam and M. Cheriet, Curved space optimization: a random search based on general relativity theory. Preprint arXiv:1208.2214 (2012).

[34] B. Mohammadi-Ivatloo, M. Moradi-Dalvand and A. Rabiee, Combined heat and power economic dispatch problem solution using particle swarm optimization with time varying acceleration coefficients. *Electr. Power Syst. Res.* **95** (2013) 9–18.

[35] S. Mirjalili and A. Lewis, The Whale optimization algorithm. *Adv. Eng. Softw.* **95** (2016) 51–67.

[36] N. Nahas and M. Abouheaf, Novel heuristic solution for the non-convex economic dispatch problem. In: *13th International Multi-Conference on Systems Signals & Devices (SSD)* (2016).

[37] N. Nahas and M. Nourelfath, Non-linear threshold accepting meta-heuristic for combinatorial optimization problems. *Int. J. Metaheuristics* **3** (2014) 265–290.

[38] N. Nahas and M. Nourelfath, Joint optimization of maintenance, buffers and machines in manufacturing lines. *Eng. Optim.* **50** (2018) 37–54.

[39] M. Nazari-Heris, M. Mehdinejad, B. Mohammadi-Ivatloo and G. Babamalek-Gharehpetian, Combined heat and power economic dispatch problem solution by implementation of whale optimization method. *Neural Comput. App.* **31** (2019) 421–436.

[40] R. Ramanathan, Emission constrained economic dispatch. *IEEE Trans. Power Syst.* **9** (1994) 1994–2000.

[41] P.K. Roy, C. Paul and S. Sultana, Oppositional teaching learning-based optimization approach for combined heat and power dispatch. *Int. J. Electr. Power Energy Syst.* **57** (2014) 392–403.

[42] A. Selvakumar and K. Thanushkodi, A new particle swarm optimization solution to nonconvex economic dispatch problems. *IEEE Trans. Power Syst.* **22** (2007) 42–51.

[43] D. Simon, Biogeography-based optimization. *IEEE Trans. Evol. Comput.* **12** (2008) 702–13.

[44] M. Singh and J.S. Dhillon, Multiobjective thermal power dispatch using opposition-based greedy heuristic search. *Int. J. Electr. Power Energy Syst.* **82** (2016) 339–353.

[45] C. Su and C. Lin, New approach with a Hopfield modeling framework to economic dispatch. *IEEE Trans. Power Syst.* **15** (2000) 541–545.

[46] A. Vasebi, M. Fesanghary and S.M.T. Bathaee, Combined heat and power economic dispatch by harmony search algorithm. *Int. J. Electr. Power Energy Syst.* **29** (2007) 713–719.

[47] T. Victoire and A. Jeyakumar, Hybrid PSO-SQP for economic dispatch with valve-point effect. *Electr. Power Syst. Res.* **71** (2004) 51–59.

[48] B. Webster and P.J. Bernhard, A local search optimization algorithm based on natural principles of gravitation. In: *Proceedings of the 2003 International Conference on Information and Knowledge Engineering (IKE'03)* (2003) 255–261.