A cuckoo bird-inspired meta-heuristic algorithm for optimal short-term hydrothermal generation cooperation

Thang Trung Nguyen*, Dieu Ngoc Vo and Bach Hoang Dinh

Abstract: Optimal short-term hydrothermal generation cooperation (OSTHTGC) is used to determine the optimal operation strategy for cascaded hydropower plants and thermal plants. In this problem, equality constraints consisting of water time delay, reservoir volume, continuity water and power balance, as well as inequality constraints related to the limits of thermal and hydro generations are taken into account. In this paper, a Cuckoo Bird-Inspired Meta-Heuristic Algorithm (CBIA), which is advanced by using only few control parameters and high success rate, has been proposed to solve the OSTHTGC problem. The proposed method is applied for four test cases where the valve point loading effects of thermal units and the combination operation of a four-cascaded reservoir system are considered. The comparison of obtained results has revealed that that CBIA method is an efficient method with high quality solution and fast convergence speed compared to other relevant approaches.

Subjects: Algorithms & Complexity; Artificial Intelligence; Power & Energy

Keywords: Cuckoo bird-inspired algorithm; optimal hydrothermal generation cooperation; valve point loading effect

1. Introduction

Optimal short-term hydrothermal generation cooperation (OSTHTGC) problem can be classified into two different problems, fixed head and variable head short-term hydrothermal scheduling. The former considers the water head of the reservoir as a constant while the latter treats the water head as

ABOUT THE AUTHOR

Thang Trung Nguyen received his BE and ME degrees in Electrical Engineering from University of Technical Education Ho Chi Minh City, Ho Chi Minh City, Vietnam in 2008 and 2010, respectively. Now, he is teaching at department of electrical and electronics engineering, Ton Duc Thang University, Ho Chi Minh city, Vietnam. His research interests include optimization of power system, power system operation and control and renewable energy. He has co-authored research papers and articles in national journals, international journals, conference proceedings, conference paper presentation and book chapters.

PUBLIC INTEREST STATEMENT

Cuckoo Bird-Inspired Meta-Heuristic Algorithm (CBIA) is an effective and robust technique for solving optimization problems where complicated objectives and complex constraints are considered. It comprised two phases of generating new solutions at each iteration, Lévy flights random walk to explore prospective domains of the search space and selective random walk to rapidly converge to a global optimal solution. Optimal short-term hydrothermal generation cooperation (OSTHTGC) minimizes the electricity generation fuel cost of thermal units while using the natural water of a river effectively with complicated hydro reservoir constraints. Applying CBIA approach to solve the OSTHTGC problem is essential and very promising.
a variable. The head of the reservoir is fixed if the reservoir volume within the entire scheduling ho-
riso� is constant. This assumption is reasonable based on the fact that the hydropower plant has a
large reservoir and the volume difference between inflow and discharge (via turbine) is very small.
On the contrary, when the reservoir capacity is quite small or the difference between the inflow and
discharge is high enough, the head of reservoir will be regarded as a variable. Obviously, the variable
head short-term scheduling is more complex than the fixed head short-term scheduling because the
power generation of hydro plants involves a function with respect to water discharge and reservoir
volume, which are varying during the optimal operation process (Wood & Wollenberg, 1984).
Furthermore, in this paper the hydro plants called cascaded reservoirs are considered where the
discharge water of the upper reservoirs is the inflow water of the lower reservoirs.

In recent years, there have been many algorithms developed for solving the cascaded hydrother-
mal scheduling problem, such as the decomposition and coordination techniques (Soares, Lyra, &
Tavares, 1980; Wang & Shahidehpour, 1993), the Evolutionary Programming (Sinha, Chakrabarti, &
Chattopadhyay, 2003; Yang, Yang, & Huang, 1996), the Genetic Algorithm (Basu, 2010; Orero &
Irving, 1998; Kumar & Naresh, 2007), the Two-phase Neural Network (Naresh & Sharma, 1999), the
Differential Evolution algorithm (Lokshminarasimman & Subramanian, 2006, 2008; Youlin,
Jianzhong, Hui, Ying, & Yongchuan, 2010) or the Hybrid algorithm of Differential Evolution and
Sequential Quadratic Programming (Sivasubramani & Shanti Swarup, 2011), the Particle Swarm
Optimization algorithm (Binghui, Xiaohui, & Jinwen, 2007; Hota, Barisol, & Chakrabarti, 2009; Mandal,
Casu, & Chakroborty, 2008; Nima & Hasson, 2010; Ying et al., 2012; Yuan, Wang, & Yuan, 2008), the
Clonal Selection Algorithm (Swain, Barisol, Hota, & Chakrabarti, 2011), the Adaptive Chaotic Artificial
Bee Colony algorithm (Xing, Jianzhong, Shuo, Rui, & Yongchuan, 2013), Teaching Learning Based
Optimization (Roy, 2013); the Krill Herd Algorithm (Roy, Pradhan, & Paul, in press), the Symbiotic
Organisms Search Algorithm (Das & Bhattacharya, in press), Quasi-Oppositional Group Search
Optimization (Basu, 2016), and Ant Lion Optimization (Dubey, Pandit, & Panigrahi, 2016).

Among these methods, Decomposition and Coordination techniques (SC) are the earliest methods
employed to deal with the complex OSTHTGC problem. The methods have used Lagrange optimiza-
tion function and divided a large problem into two sub-problems such as thermal sub-problem and
thermal sub-problem (Soares et al., 1980; Wang & Shahidehpour, 1993). A principal difficulty which
is considered over the optimal interval in the studies is the effect of stochastic load demand.
Therefore, the stochastic sub-problem must first be solved to determine the fixed input data for the
two other sub-problems. Then, based on the Lagrange function, the thermal sub-problem is solved
for lambda value which is used as input data in the hydro sub-problem. Finally, the solution for the
hydrothermal system scheduling including thermal and hydro generations is obtained. In spite of
obtaining accurate solutions and insignificant constraint violations, the mentioned methods still suf-
fer from the basic drawback of the Lagrange optimal approach, which is unable to deal with the
complex power systems with the non-convex fuel cost function of thermal units. In Yang et al.
(1996), the conventional evolutionary (CEP) has been applied to solve an OSTHTGC problem with the
non-convex objective function. Other methods based evolutionary programming, such as the fast EP
(FEP) and improved fast EP (IFEP) have been developed to solve the more complex OSTHTGC problem
considering the non-convex objective function and prohibited operating zone of hydro units (Sinha
et al., 2003). The CEP was stated more efficient and robust than simulated annealing via compari-
sions of result obtained from two different systems. However, there was no comparison among these
modified EP methods with other research to verify the ability of the methods to deal with large scale
and complex hydrothermal systems. The first classical GA (CGA) and its improved version, real-coded
genetic algorithm (RCGA) applied for the OSTHTGC problem are presented in Orero and Irving (1998),
Kumar and Naresh (2007), Basu (2010), respectively. The study did not aim to demonstrate any ad-
vantages of the CGA over other methods but testing the ability of CGA to deal with constraint viola-
tions on a power system of four cascaded hydropower plants and one thermal plant with quadratic
fuel cost function of thermal units and without transmission losses. Dissimilarly, the RCGA has been
tested on a larger power system of four cascaded hydropower plants and three thermal plants with
non-convex fuel cost function. However, if only paying attention on the computational time, the GA
based methods are relatively weak because they have taken much time to reach optimal solutions. Another method based TPNN taking the scheduled water discharge of hydro reservoirs as the states of analogue neurons (Naresh & Sharma, 1999) was developed to deal with the OSTHTGC problem and compared to the standard augmented Lagrange method (ALM). Although TPNN could obtain better solutions than ALM, but the method have the same disadvantage as applied to the non-differential problem. The modified DE (Lakshminarasimman & Subramanian, 2006) has proposed several modifications to deal with equality constraints like the load balance, especially the reservoir’s volume at the final subinterval. Moreover, a new approach modified from hybrid DE (Lakshminarasimman & Subramanian, 2008) was developed to deal with equality constraints and to reduce computational time. The hybrid DE was built as two extra operations, acceleration and migration, where the former has improved the fitness quality and the later has exploited the search space. As a result, this method could converge to global optimal solutions very fast compared to conventional DE, MDE, HDE and several other methods.

An adaptive chaotic differential evolution (ACDE) integrating an adaptive dynamic control mechanism for the crossover factor (Youlin et al., 2010) has been proposed to control the recombination and chaotic local search operation to avoid the effect of premature convergence (local trap). It is one of the best modified DE versions with the high quality solution and fast computational time. Similarly, the PSO method has been applied for a large scale hydrothermal system consisting of four hydro plants and three thermal plants with non-convex fuel cost function (Mandal et al., 2008). Several improved versions of PSO have also been developed by recommending new various factors, such as the inertia weight and constriction factor, and a selection strategy of best solutions from either all particles or just several specific particles. However, despite many attempts, the modified versions of PSO could not get the better solution than the improved versions of DE in the same test systems.

New modifications of DE, such as Clonal Selection Algorithm (Swain et al., 2011) and Hybrid Differential Evolution and Sequential Quadratic Programming (HDE–SQP) (Sivasubramani & Shanti Swarup, 2011), have been developed to improve the solution quality and convergence speed for hydrothermal systems with fixed head and variable head. Several test cases have been performed to verify the efficiency of the proposed methods in the complex power systems considering the non-convex objective function and the prohibited zone of hydro units. An adaptive chaotic artificial bee colony (ACABC) (Xiang et al., 2013) has been implemented for searching the optimal solution of the short-term hydrothermal scheduling problem considering nonlinear constraints and nonlinear objective functions. The method could avoid the effect of premature convergence thank to the chaotic search and adaptive coordinating mechanism. A novel teaching learning based optimization (TLBO) has been applied to the problem with the non-convex fuel cost function and the prohibited zone of hydro units (Roy, 2013). The TLBO is mainly based on teaching phase and learning phase, and does not need any algorithm to determine the control parameters. Similarly, some recent state-of-the-art algorithms have been presented in Roy et al. (in press), Das and Bhattacharya (in press), Basu (2016), Dubey et al. (2016) to solve the OSTHTGC problem with promising results as the lower fuel costs and acceptable simulation time. However, the studies have verified their performance by the convergence time only, but there was no any comparison of computer processors which used in the performances. This issue will solved and presented in detail in our paper.

In this paper, a Cuckoo Bird-Inspired Meta-Heuristic Algorithm (CBIA), which is advanced by using only few control parameters and high success rates, has been proposed to solve the OSTHTGC problem. The CBIA, inspired from the intelligent reproduction behaviour of cuckoo birds (Yang & Deb, 2009), has several advantages over PSO and GA for benchmark functions such as obtaining better solution quality, higher success rate, and fewer easily selected control parameters. Consequently, CBIA has been widely and successfully applied for various engineering optimization problems such as the economic load dispatch (Dieu, Schegner, & Ongsakul, 2013), the hydrothermal scheduling (Thang & Dieu, 2015; Thang, Dieu, & Anh, 2015) and the distribution network reconfiguration. The OSTHTGC problem considered the non-convex fuel cost function of thermal units and the cascaded reservoirs of hydro plants. The four test systems have been applied to verify the performance of the
proposed CBIA and the simulation results in terms of total costs as well as run time are used to compare to those of other reported methods. The comparisons have shown that the proposed CBIA is a very strong method for solving the short-term cascaded hydrothermal scheduling problem.

2. Problem formulation

2.1. Fuel cost objective

The main objective of the OSTHTGC problem is to minimize total generation fuel cost while satisfying the constraints of hydraulic, load power balance, and generation capacity limits. The OSTHTGC problem having $N_1$ thermal units and $N_2$ hydro units scheduled in $M$ sub-intervals is formulated as follows:

$$\text{Min } C_T = \sum_{m=1}^{M} \sum_{i=1}^{N_1} t_m F_{im}$$

where $F_{im}$ is the fuel cost of the $i$th thermal unit for one hour at the $m$th subinterval. Traditionally, the fuel cost of thermal units is approximately represented as a quadratic function, such as

$$F_{im} = \left[ a_{si} + b_{si} P_{sl,m} + c_{si} P_{sl,m}^2 \right]$$

Recently, the fuel cost of thermal units with valve-point loading effects has been widely used in the power system optimization problems. This curve contains higher order nonlinearity and discontinuity due to the valve-point loading effects as follows:

$$F_{im} = \left[ a_{si} + b_{si} P_{sl,m} + c_{si} P_{sl,m}^2 + d_{si} \times \sin \left( e_{si} \times \left( p_{min}^{sl} - p_{sl,m} \right) \right) \right]$$

2.2. Operational constraints

The proposed system has to satisfy following constraints:

- Equality constraint of power balance

$$\sum_{i=1}^{N_1} P_{si,m} + \sum_{j=1}^{N_2} P_{hj,m} - P_{t,m} - P_{d,m} = 0$$

where $P_{t,m}$ and $P_{d,m}$ are load demand and transmission loss at the $m$th subinterval. $P_{hj,m}$ is the power output of hydro plant $j$ at the $m$th subinterval which is related to the water discharge and the reservoir volume as described by the function below:

$$P_{hj,m} = C_{1hj}(V_{j,m})^2 + C_{2hj}(Q_{j,m})^2 + C_{3hj}Q_{j,m}V_{j,m} + C_{4hj}V_{j,m} + C_{5hj}Q_{j,m} + C_{6hj}$$

where $C_{1hj}, C_{2hj}, C_{3hj}, C_{4hj}, C_{5hj}, C_{6hj}$ are the six coefficients of the $j$th hydropower plant.

- Hydraulic continuity equation

The water flow of hydro plants must satisfy the hydraulic constraints as

$$V_{j,m-1} - V_{j,m} + I_{j,m} - Q_{j,m} - S_{j,m} + \sum_{i=1}^{Nu} \sum_{m=1}^{M} (Q_{i,m-\tau_j} + S_{i,m-\tau_j}) = 0$$
where $V_{j,m}, I_{j,m}$ and $S_{j,m}$ are reservoir volume, water inflow and spillage discharge rate of $j$th hydro-power plant in $m$th interval. $\tau_j$ is the water delay time between reservoir $j$ and its up-stream $i$ at the $m$th subinterval and $N_u$ is the number of up-stream units above hydro-plant $j$.

- Initial and final reservoir volumes

$$V_{j,0} = V_{j,\text{initial}}; \quad V_{j,M} = V_{j,\text{end}}$$ (7)

- Reservoir storage and water discharge Limits

$$V_{j,\text{min}} \leq V_{j,m} \leq V_{j,\text{max}}; \quad j = 1, 2, \ldots, N_2; \quad m = 1, 2, \ldots, M$$ (8)

$$Q_{j,\text{min}} \leq Q_{j,m} \leq Q_{j,\text{max}}; \quad j = 1, 2, \ldots, N_2; \quad m = 1, 2, \ldots, M$$ (9)

where $V_{j,\text{max}}$ and $V_{j,\text{min}}$ are the maximum and minimum reservoir storage of the hydro plant $j$, respectively; $Q_{j,\text{max}}$ and $Q_{j,\text{min}}$ are the maximum and minimum water discharge of the hydro plant $j$.

- Generator operating limits

$$p_{\text{si,min}} \leq p_{\text{i,m}} \leq p_{\text{si,max}}; \quad i = 1, 2, \ldots, N_1; \quad m = 1, 2, \ldots, M$$ (10)

$$p_{\text{hj,min}} \leq p_{\text{h,m}} \leq p_{\text{hj,max}}; \quad j = 1, 2, \ldots, N_2; \quad m = 1, 2, \ldots, M$$ (11)

where $p_{\text{si,max}}, p_{\text{si,min}}, p_{\text{hj,max}}, p_{\text{hj,min}}$ are maximum, minimum power output of thermal plant $i$ and hydro plant $j$, respectively.

**2.3. The slack water discharge and the slack power output**

In the optimization methods based meta-heuristic, to keep the balance constraints in power systems, most operational variables are first determined except the slack values. The slack water discharge of $j$th reservoir at the $m$th subinterval, $Q_{j,M,d}$ and the slack power output of thermal unit 1 at the $m$th subinterval, $P_{s1,m}$ the $m$th used to exactly keep power balance constraint (4) and volume constraint (7), respectively, are obtained by

$$Q_{j,M,d} = V_{j,0} - V_{j,M} + \sum_{m=1}^{M} I_{j,m} - \sum_{m=1}^{M-1} Q_{j,m} - \sum_{m=1}^{M} S_{j,m} + \sum_{i=1}^{N_u} \sum_{m=1}^{M} \left( Q_{i,m-\tau_i} + S_{i,m-\tau_i} \right) = 0$$ (12)

$$P_{s1,m} = P_{D,m} + P_{I,m} - \sum_{i=2}^{N_2} p_{\text{si,m}} - \sum_{j=1}^{N_2} p_{\text{hj,m}}$$ (13)

**3. The proposed method based Cuckoo Bird-Inspired Algorithm for the OSTHTGC problem**

**3.1. Cuckoo Bird-Inspired Algorithm**

Cuckoo bird is one of brood parasite species so it does not build its own nest and female cuckoo will lay her own eggs to other host bird nests. The cuckoos are very intelligent to choose the host bird, whose eggs with the same color as Cuckoos eggs. The strategy allows the Cuckoo egg to trick the host bird since the host bird cannot identify any alien eggs in their own nests. The fact demonstrates why there are more than 120 species of other birds can be cheated and continue to incubate the Cuckoo eggs until they are hatched.

Not every host bird is totally tricked, however, about 20% of Cuckoo eggs will be recognized as alien eggs and thrown away out of the nests or the host bird forsakes them and the host nest. Each female Cuckoo can lay between 12 and 22 eggs per season and lays each one in each nest. On the other hand, before laying Cuckoo eggs into other nests, the Cuckoos carefully observe the routine...
and the behavior of the other species to select the specie which has longer timing of hatching than them. Thanks to this strategy, Cuckoo chicks are very aggressive toward the host chicks; therefore, the first instinct action that Cuckoo chicks will do is to propel the host eggs out of the nest, increasing the food host bird provide the Cuckoo chicks.

Therefore, the Cuckoo optimisation algorithm (Yang & Deb, 2009), inspired from the cuckoo’s behavior in the real life, consists of three idealized rules below:

**Rule 1**: Each cuckoo lays eggs and put each egg in a nest of other species.

**Rule 2**: The best nest with the highest quality of cuckoo egg will be carried over to the next generation.

**Rule 3**: A fraction of the initial cuckoo eggs may be discovered as alien eggs by the host bird. The probability of the discovery is in range of \([0, 1]\). In this case, the host bird either propels the alien egg out of its nest or forsakes its nest to build a new one elsewhere. To simplify it is supposed that a fraction \(P_a\) of the number of nests is replaced by new nests in this rule.

Therefore, the method based CBIA is developed corresponding to the three important stages below:

- **Initialization**: A population of \(N_p\) host nests is randomly initialized by using Rule 1.
- The first phase of generating new solutions: The new solutions will be randomly generated via Lévy Flights corresponding to Rule 2.
- The second phase of generating new solutions: The new solutions will be generated via the discovery of alien eggs corresponding to Rule 3.

### 3.2. Implementation of Cuckoo Bird-Inspired Algorithm

Based on the three rules in Section 3.1, the method based CBIA for solving OSTHTGC problems is described as follows.

#### 3.2.1. Initialization

Similar to other meta-heuristic algorithms, each cuckoo nest (solution) in \(N_p\) nests is represented by a vector \(X_d = [P_{s1,m,d}, Q_{j,m,d}] (d = 1, ..., N_p)\). Certainly, the valid solution has to satisfy the upper and lower limits shown in Equations (8)÷(11). Consequently, each solution \(X_d\) (cuckoo nest) should be randomly initialized within the limits \(P_{s1,m} \leq P_{s1,m,d} \leq P_{s1,\text{max}}\) \((i = 2, ..., N_1; m = 1, ..., M)\) and \(Q_{j,m} \leq Q_{j,m,d} \leq Q_{j,\text{max}}\) \((j = 2, ..., N_2; m = 1, ..., M - 1)\).

From Equation (6), the reservoir volume at the \(m\)th subinterval is obtained by:

\[
V_{j,m} = V_{j,m-1} + I_{j,m} - Q_{j,m} - S_{j,m} + \sum_{i=1}^{N_1} \left( Q_{j,m-i} + S_{j,m-i} \right) \tag{14}
\]

The values of \(Q_{j,\text{lim}}\) is obtained by (12) and hydro generations can be then calculated using Equation (5). The slack thermal unit is obtained using Equation (13).

Based on the initial population of nests, the fitness function to verify the quality of optimization solutions for the considered problem is calculated by

\[
FT_d = \left( \sum_{m=1}^{M} \sum_{i=1}^{N_1} F(P_{s1,m,d}) + K_s \sum_{m=1}^{M} (P_{s1,m,d} - P_{\text{lim}})^2 + K_v \sum_{j=1}^{M} \sum_{m=1}^{M-1} (V_{j,m,d} - V_{\text{lim},j}^2) \right) \tag{15}
\]

+ \(K_Q \sum_{j=1}^{N_2} \left( Q_{j,M,d} - Q_{j,\text{lim}} \right)^2 + K_h \sum_{j=1}^{N_2} \sum_{m=1}^{M} (P_{hj,m,d} - P_{hj,\text{lim}})^2)\)
where $K_s$ and $K_h$ are respectively penalty factors for the slack thermal unit 1 and all hydro units; $K_v$ and $K_q$ are respectively penalty factors for reservoir volume over $M - 1$ subintervals and water discharge at the subinterval $M$. Furthermore, the limits of variables in Equation (15) can be calculated as below.

$$\begin{align*}
p_{\text{lim}}_s &= \begin{cases} 
    p_{1,m}^{\text{max}} & \text{if } p_{1,m} > p_{1,m}^{\text{max}}, \quad m = 1, \ldots, M, \\
    p_{1,m}^{\text{min}} & \text{if } p_{1,m} < p_{1,m}^{\text{min}}, \\
    p_{1,m} & \text{otherwise}
\end{cases} 
\end{align*}$$  (16)

$$\begin{align*}
v_{\text{lim}}_j &= \begin{cases} 
    v_{j,m}^{\text{max}} & \text{if } v_{j,m} > v_{j,m}^{\text{max}}, \\
    v_{j,m}^{\text{min}} & \text{if } v_{j,m} < v_{j,m}^{\text{min}}, \\
    v_{j,m} & \text{otherwise}
\end{cases} 
\end{align*}$$  (17)

$$\begin{align*}
q_{\text{lim}}_j &= \begin{cases} 
    q_{j,M}^{\text{max}} & \text{if } q_{j,M} > q_{j,M}^{\text{max}}, \\
    q_{j,M}^{\text{min}} & \text{if } q_{j,M} < q_{j,M}^{\text{min}}, \\
    q_{j,M} & \text{otherwise}
\end{cases} 
\end{align*}$$  (18)

$$\begin{align*}
p_{\text{lim}}_{hj} &= \begin{cases} 
    p_{hj}^{\text{max}} & \text{if } p_{hj,m} > p_{hj}^{\text{max}}, \\
    p_{hj}^{\text{min}} & \text{if } p_{hj,m} < p_{hj}^{\text{min}}, \\
    p_{hj,m} & \text{otherwise}
\end{cases} 
\end{align*}$$  (19)

3.2.2. The first phase of generating new solutions

In this section, the new solutions generated by Lévy flights (Thang & Dieu, 2015; Thang et al., 2015) as shown by

$$\begin{align*}
X_{\text{new}}_d &= X_d + \alpha (X_{\text{best}} - X_d) \left( r_1 \times \frac{\sigma(X)}{\sqrt{r(X)}} \right) 
\end{align*}$$  (20)

where $X_{\text{best}}$ and $X_d$ are the best egg (the current best solution) and the $d$th egg in the population ($N_p$ eggs); $\alpha > 0$ is an updated step size.

The value of $\alpha$ has a significant influence on the searching process because different new solutions would be produced according to specific values of $\alpha$. If this parameter is set to a high value, there is a huge difference between the old and new solutions and the searching process is either obtained a chance of fast convergence or left outside of the feasible zone. On the contrary, if the value is set too small the location for the new solution is very close to the previous and the optimal search strategy is also not effective due to long computational time. Similar to other meta-heuristic methods, there is no criterion to appropriately select the parameter $\alpha$, but it should be adjusted via experiment work. The new solutions generated from (20) must satisfy the required limits and need to be redefined if violated according to the equations below

$$\begin{align*}
X_{\text{new}}_d &= \begin{cases} 
    X_{d,\text{max}} & \text{if } X_{\text{new}}_d > X_{d,\text{max}} \\
    X_{d,\text{min}} & \text{if } X_{\text{new}}_d < X_{d,\text{min}}
\end{cases} 
\end{align*}$$  (21)

3.2.3. The second phase of generating new solutions

In this section, the second phase of solution generation is used to improve quality of the previously obtained solution. This mechanism differs from other meta-heuristic methods, leading to a better optimization solution and faster computational time. In the cuckoo bird’s behavior, there is a possibility that an alien egg may be identified by the host bird. Thus, it could either throw the exposed egg out of its nest or forsake its own nest with all eggs and build a new one. Like the Lévy flights, the
discovery action of alien eggs in the nests is randomly implemented with a probability of $P_a$ to generate a new solution for an optimization problem. This new solution is created by:

$$X_d^{\text{dis}} = \begin{cases} X_d + \text{rand}(X_{r1} - X_{r2}) & \text{if } \text{rand} < P_a \\ X_d & \text{otherwise} \end{cases}$$

The newly obtained solutions also need to be redefined using Equation (21) above in case they violate the upper or lower values.

3.2.4. Stopping criteria

The searching process mentioned above will stop when the maximum number of iterations is reached.

3.3. Overall procedure of solving the OSTHTGC problem using the proposed CBIA

Step 1: Select CBIA parameters including the number of host nests, $N_p$, the probability of a host bird discovering an alien egg in its nest, $P_a$, and the maximum number of iterations, $G_{\text{max}}$.

Step 2: Initialize the population of $N_p$ host nests as mentioned in Section 3.3.1.

- Calculate the slack water discharge and the slack thermal unit 1 using Equations (12) and (13).
- Calculate all hydro generation powers using Equation (5).

Step 3: Evaluate the fitness functions using Equation (15) to choose the best nest (current best solution) corresponding to the lowest fitness function value, $X_{\text{best}}$.

- Set the initial iteration $G = 1$.

Step 4: Generate a new set of solutions via Lévy flights as described in Section 3.3.2 and repair any violated solutions by using Equation (21).

Step 5: Calculate the slack water discharge and the slack thermal unit 1 corresponding to the new set of solutions using Equations (12) and (13), respectively.

- Calculate new values of all hydro generations using e.g. (5).

Step 6: Calculate the functions of the newly obtained solutions using Equation (15).

- Compare the fitness of the new solution and that of the old solution (at the same nest) to retain a better one at each nest.

Step 7: Generate a new set of solutions based on the discovery action of alien egg as described in Section 3.3.3 and repair any violated solution using Equation (21).

Step 8: Calculate the slack water discharge and the slack thermal unit 1 corresponding to the new set of solutions using (12) and (13), respectively.

- Calculate new values of all hydro generations using (5).

Step 9: Calculate the fitness functions for the newer set of solutions using (15).

- Compare the fitness of the new solution and that of the current solution (at the same nest) to retain a better one at each nest.
Step 10: Evaluate all updated current solutions to choose the best current solution, $G_{best}$.

Step 11: Check if $G < G_{max}$, $G = G + 1$ and return to Step 4. Otherwise, stop.

4. Numerical results
In this paper, the performance of the proposed CBIA has been verified by using four test cases of the OSTHTGC problem. The proposed CBIA is coded in MATLAB platform and implemented 50 independent trials for each specific value of $P_a$ which run on the PC configuration of 1.8 GHz processor and 4 GB RAM.

4.1. Two test cases with the quadratic fuel cost function of thermal plants
In this section, two test cases comprising four cascaded hydropower plants and one thermal plant with the quadratic fuel cost function scheduled in 24 one-hour sub-intervals are considered. The transportation delay times in hour considered in test case 1 are $\tau_{13} = 2$, $\tau_{34} = 4$, and in test case 2 are $\tau_{13} = 1$, $\tau_{34} = 2$. The full data for the test case 1 and test case 2 are taken from Lakshminarasimman and Subramanian (2008), Youlin et al. (2010), respectively.

The simulation has been performed via various values of $P_a$ in a range of $[0.1, 0.9]$ with a step of 0.1 to analyze the effect of $P_a$ for the CBIA method. The number of nests and the maximum number of iterations are respectively set to 100 and 15,000. Tables 1 and 2 present the statistic data of obtained results for the test case 1 and 2, respectively. As observed from Table 1, the best minimum and average costs are respectively obtained at $P_a = 0.3$ and 0.1 whereas the best standard deviation costs are gotten at $P_a = 0.4$. Table 2 shows that the best minimum cost and average cost are respectively $154590.1657$ and $154595.5666$ obtained at the same value $P_a = 0.6$ meanwhile the standard deviation costs is reached at $P_a = 0.3$ for system 2. Moreover, Tables 3 and 4 show the comparison results of CBIA and other methods for test cases 1 and 2, respectively. As shown in these tables, CBIA can obtain the optimal solution better than all other relevant methods (Kumar & Naresh, 2007) and the computation time faster than most of mentioned methods except BCGA and and RCGA for only test case 1. To analyze in more details about the CBIA performance, the fitness convergence characteristic for the test system 1 is depicted in Figure 1 as well as the optimal solutions in terms of water discharge, thermal and hydro generations for the two test systems are shown in Figures 2–5. All of these results demonstrate the advanced efficiency of CBIA compared to other relevant methods. Furthermore, to fairly compare the computational time among the considered methods for solving the same problem but using different programming languages and/or different computer processors, we convert the execution time corresponding to CPU speed (GHz) of the mentioned methods into a common base unit which is defined by CPU time of CBIA with 1.8 GHz processor. Thus, the adjusted CPU time in PU is determined by:

| $P_a$ | Min total cost ($) | Average total cost ($) | Max total cost ($) | Std. dev. ($) | Avg. CPU (s) |
|-------|--------------------|------------------------|-------------------|--------------|--------------|
| 0.1   | 922,505.37         | 922,671.91             | 923,097.54        | 113.52       | 78.8         |
| 0.2   | 922,532.38         | 922,712.76             | 923,089.14        | 115.63       | 79.4         |
| 0.3   | 922,487.68         | 922,706.58             | 923,000.68        | 102.85       | 78.6         |
| 0.4   | 922,555.43         | 922,687.45             | 922,933.15        | 94.73        | 80.5         |
| 0.5   | 922,550.46         | 922,716.58             | 923,074.87        | 105.52       | 78.4         |
| 0.6   | 922,545.09         | 922,740.90             | 923,071.94        | 121.77       | 79.6         |
| 0.7   | 922,562.51         | 922,750.78             | 923,054.95        | 130.46       | 80.3         |
| 0.8   | 922,548.25         | 922,723.96             | 923,068.80        | 120.52       | 80.6         |
| 0.9   | 922,553.64         | 922,713.39             | 923,120.62        | 125.89       | 79.7         |
It is noted that the value of 1.8 (GHz) is the processor of the CPU chip used to run CBIA and the CPU time obtained by CBIA is used as the common base time. Therefore, the adjusted CPU time determined for other methods is a number of relative times compared to CPU time of the proposed CBIA. Of course, according to Equation (23), the adjusted CPU time of the CBIA is absolutely 1.0 pu.

$$\text{Adjusted CPU time} = \frac{\text{Given CPU speed (GHz)}}{1.8 \text{ (GHz)}} \times \frac{\text{Given CPU time (second)}}{\text{CPU time from CBIA (second)}}$$  \hspace{1cm} (23)
Figure 1. The cost convergence characteristic for test case 1 (convex system).

Figure 2. Optimal water discharge for test case 1 (convex system).

Figure 3. Hydro and thermal generations for test case 1 (convex system).
Consequently, if the adjusted CPU time of any method is 2.0 pu, it means that this method executes two times slower than the CBIA in the same problem. On the contrary, if that value is only 0.5 pu, it indicates that the method runs two times faster than CBIA. Table 5 presents the adjusted CPU times in pu for the test case 1 corresponding to the CPU chip used to run these methods. However, it is unable to do the same for the test system 2 because the relevant papers of mentioned methods did not report their execution time. As observed from Table 5, the adjusted CPU time for test system 1, the proposed CBIA method is the relatively fastest, which is from 1.17 to 13.77 times faster than other mentioned methods.
### 4.2. Two test cases of complex power systems with the non-convex fuel cost function of thermal plants

In this section, the test case 3 has applied for a power system comprising four cascaded hydropower plants and one thermal plant with the non-convex fuel cost function and test case 4 has comprised four cascaded hydropower plants and three thermal plants with the non-convex fuel cost function. The optimization period is scheduled in 24 one-hour sub-intervals. The full data for the test case 3 and test case 4 are taken from Kumar & Naresh (2007) and Basu (2004), respectively. Similar to the investigation of the CBIA method for the test systems with the quadratic fuel cost function of thermal plants, as described in Section 4.1, this simulation has been performed via 9 values of $P_a$ in a range of $[0.1, 0.9]$ with a step of 0.1. The number of nests and the maximum number of iterations are respectively set to 100 and 15,000. Tables 6 and 7 present the statistic data of obtained results for the test case 3 and 4, respectively. As observed from these tables, the best minimum cost of test systems 3 is $947056.12$ at $P_a = 0.4$ and that of test system 4 is $41064.897$ at $P_a = 0.5$. To analyze in

| Method                          | Program language and processor used (GHz) | CPU speed (pu) | Given CPU time (s) | Given CPU time (pu) | Adjusted CPU time (pu) |
|--------------------------------|------------------------------------------|----------------|-------------------|---------------------|------------------------|
| CEP Mandal et al. (2008)       | Matlab, 0.85                             | 0.47           | 2.292.10          | 29.16               | 13.77                  |
| FEP Mandal et al. (2008)       | Matlab, 0.85                             | 0.47           | 1.911.20          | 24.32               | 11.48                  |
| IFEP Mandal et al. (2008)      | Matlab, 0.85                             | 0.47           | 1.033.20          | 13.15               | 6.21                   |
| GA Lakshminarasimman and Subramanian (2006) | NA                                       | NA             | 1.920.00          | 24.43               | NA                     |
| BCGA Lakshminarasimman and Subramanian (2008) | Matlab, 3.0                             | 1.67           | 64.51             | 0.82                | 1.37                   |
| RCGA Lakshminarasimman and Subramanian (2008) | Matlab, 3.0                             | 1.67           | 57.52             | 0.73                | 1.22                   |
| MDE Basu (2004)                | NA                                       | NA             | NA                | NA                  | NA                     |
| GCPPO Thang et al. (2015)      | C++, 2.0                                 | 1.11           | 182.40            | 2.32                | 2.58                   |
| GWPSO Thang et al. (2015)      | C++, 2.0                                 | 1.11           | 129.10            | 1.64                | 1.82                   |
| LCPSO Thang et al. (2015)      | C++, 2.0                                 | 1.11           | 103.50            | 1.32                | 1.46                   |
| LWPSO Thang et al. (2015)      | C++, 2.0                                 | 1.11           | 82.90             | 1.05                | 1.17                   |
| EGA Thang and Dieu (2015)      | Matlab, 2.0                              | 1.11           | NA                | NA                  | NA                     |
| PSO Thang and Dieu (2015)      | Matlab, 2.0                              | 1.11           | NA                | NA                  | NA                     |
| EPSO Thang and Dieu (2015)     | Matlab, 2.0                              | 1.11           | NA                | NA                  | NA                     |
| IPSOSinha et al. (2003)        | Matlab, 3.06                             | 1.70           | NA                | NA                  | NA                     |
| CBIA                          | Matlab, 1.8                              | 1.00           | 78.60             | 1.00                | 1.00                   |
more details about the CBIA performance, the fitness convergence characteristic for test system 4 is depicted in Figure 6 as well as the optimal solutions in terms of water discharge, thermal and hydro generations for the two non-convex test systems are shown in Figures 7–10. Moreover, Tables 8 and 9 show the comparison results of CBIA and other relevant methods for test case 3 and 4, respectively. As shown in Table 8, the CBIA method can obtain the optimal solution better than other methods but taking more time for convergence. As seen from the Table 9, the fuel cost of CBIA is the

| \( P_a \) | Min total cost ($) | Average total cost ($) | Max total cost ($) | Std. dev. ($) | Avg. CPU (s) |
|-------|------------------|------------------------|-------------------|--------------|------------|
| 0.1   | 947,095.6296     | 947,392.965            | 948,110.5547      | 194.09125    | 100.1      |
| 0.2   | 947,170.6835     | 947,469.651            | 947,867.3684      | 165.7907     | 99.8       |
| 0.3   | 947,362.4929     | 947,450.932            | 947,878.1209      | 165.65557    | 102.3      |
| 0.4   | 947,056.12       | 947,461.61             | 948,049.804       | 191.9116     | 101.6      |
| 0.5   | 947,131.15       | 947,472.155            | 947,939.308       | 205.11005    | 99.2       |
| 0.6   | 947,081.1536     | 947,487.074            | 948,074.0174      | 216.89415    | 102.34     |
| 0.7   | 947,097.6764     | 947,471.911            | 947,862.1274      | 198.47885    | 103.4      |
| 0.8   | 947,142.6228     | 947,498.684            | 948,012.9466      | 236.07314    | 105.6      |
| 0.9   | 947,192.5979     | 947,497.052            | 948,108.7591      | 225.02246    | 103.4      |

| \( P_a \) | Min total cost ($) | Average total cost ($) | Max total cost ($) | Std. dev. ($) | Avg. CPU (s) |
|-------|------------------|------------------------|-------------------|--------------|------------|
| 0.1   | 42,528.59        | 44,973.76              | 46,371.89         | 943.67       | 96.7       |
| 0.2   | 41,672.16        | 43,710.50              | 45,800.02         | 1,075.76     | 97.5       |
| 0.3   | 41,475.74        | 43,023.10              | 45,600.13         | 945.11       | 98.8       |
| 0.4   | 41,840.11        | 42,936.28              | 44,563.48         | 671.28       | 92.6       |
| 0.5   | 41,064.897       | 42,787.139             | 45,101.07         | 705.069      | 94.4       |
| 0.6   | 41,780.89        | 42,974.18              | 44,689.71         | 578.65       | 95.1       |
| 0.7   | 41,969.76        | 43,113.58              | 44,827.05         | 647.33       | 97.7       |
| 0.8   | 41,770.26        | 42,945.50              | 44,217.00         | 532.37       | 98.6       |
| 0.9   | 41,688.42        | 43,182.07              | 44,712.98         | 670.48       | 99.2       |
smallest one while the second best fuel cost from MHDE (Lakshminarasimman & Subramanian, 2008) is $41,856.50 and the poorest from SA (Mandal et al., 2008) is $47,306.00. It implies that the saving cost, thanks to using CBIA, is approximately 2 and 15% compared to using SA and MHDE, respectively. Table 10 reports the adjusted CPU time in pu of relevant methods to fairly compare among their execution time. As seen from test system 3, the CBIA and two versions of GA including BCGA and RCGA (Kumar & Naresh, 2007) have nearly the same values of the adjusted CPU time. On the contrary, at test system 4 the adjusted CPU time of CBIA is higher than that from most other
methods except DE and MDE (Lakshminarasimman & Subramanian, 2008) and Clonal selection algorithm (Swain et al., 2011). However, the mentioned methods with faster computation time have implemented in the C++ and JAVA programming which are from two to three times faster than Matlab.
Table 8. Performance comparison of relevant methods for test case 3 (non-convex system)

| Method                                           | Min cost ($) | Avg. time (s) |
|-------------------------------------------------|--------------|---------------|
| BCGA Lakshminarasimman and Subramanian (2008)   | 952,618.00   | 66.3          |
| RCGA Lakshminarasimman and Subramanian (2008)   | 951,559.24   | 57.32         |
| DE Swain et al. (2011)                          | 947,497.85   | NA            |
| CBIA                                            | 947,056.89   | 97            |

Table 9. Performance comparison of relevant methods for test case 4 (non-convex system)

| Method                                           | Cost ($) | CPU (s) |
|-------------------------------------------------|----------|---------|
| EP-IFS Yang and Deb (2009)                      | 45,063.00| NA      |
| SA Thuan and Anh (2015)                         | 47,306.00| NA      |
| EP Thuan and Anh (2015)                         | 45,466.00| NA      |
| PSO Thuan and Anh (2015)                        | 44,740.00| NA      |
| DE Basu (2010)                                  | 44,526.10| 200     |
| MDE Basu (2010)                                 | 42,611.14| 125     |
| HDE Basu (2010)                                 | 42,337.30| 48      |
| MHDE Basu (2010)                                | 41,856.50| 31      |
| Clonal selection Roy (2013)                     | 42,440.574| 109   |
| KHA Das and Bhattacharya (in press)             | 41,926.00| NA      |
| QOGSO Sivasubramani and Shanti Swarup (2011)    | 42,120.02| 625.07  |
| CBIA                                            | 41,064.897| 94.4   |

Table 10. Comparison of adjusted computational time for test case 3 and 4 (non-convex system)

| Non-convex system | Method                                           | Program language & processor used (GHz) | CPU speed (pu) | Given CPU time (s) | Given CPU time (pu) | Adjusted CPU time (pu) |
|-------------------|-------------------------------------------------|----------------------------------------|---------------|--------------------|--------------------|------------------------|
| 3                 | BCGA Lakshminarasimman and Subramanian (2008)   | Matlab, 3.0                            | 3.0           | 66.3               | 0.68               | 1.14                   |
|                   | RCGA Lakshminarasimman and Subramanian (2008)   | Matlab, 3.0                            | 3.0           | 57.32              | 0.59               | 0.99                   |
|                   | CBIA                                            | Matlab, 1.8                            | 1.8           | 97                 | 1.00               | 1.00                   |
| 4                 | DE Basu (2010)                                  | C++, 2.4                               | 1.33          | 200                | 2.12               | 2.82                   |
|                   | MDE Basu (2010)                                 | C++, 2.4                               | 1.33          | 125                | 1.32               | 1.77                   |
|                   | HDE Basu (2010)                                 | C++, 2.4                               | 1.33          | 48                 | 0.51               | 0.68                   |
|                   | MHDE Basu (2010)                                | C++, 2.4                               | 1.33          | 31                 | 0.33               | 0.44                   |
|                   | Clonal selection Roy (2013)                     | Matlab, 3.06                           | 1.70          | 109                | 1.15               | 1.96                   |
|                   | AABC Naresh and Sharma (1999)                   | JAVA, 2.53                             | 1.41          | 15                 | 0.16               | 0.22                   |
|                   | CABC Naresh and Sharma (1999)                   | JAVA, 2.53                             | 1.41          | 21                 | 0.22               | 0.31                   |
|                   | ACABC Naresh and Sharma (1999)                  | JAVA, 2.53                             | 1.41          | 16                 | 0.17               | 0.24                   |
|                   | KHA Das and Bhattacharya (in press)             | Matlab, 3.0                            | 1.67          | NA                 | NA                 | NA                     |
|                   | QOGSO Sivasubramani and Shanti Swarup (2011)    | Matlab, 3.0                            | 1.67          | 625.07             | 6.62               | 11.1                   |
|                   | CBIA                                            | Matlab, 1.8                            | 1.00          | 94.4               | 1.00               | 1.00                   |
5. Conclusions

The paper has presented the application of the Cuckoo bird-inspired algorithm for solving OSTHTGC problem in which cascaded reservoirs with a set of complicated hydraulic constraints and non-convex fuel cost function of thermal units are taken into consideration. In order to verify the powerful search process of the proposed CBIA method, four test systems including two with the quadratic fuel cost function and two with the non-convex fuel cost function of thermal units have been considered. Moreover, the evaluation is based on not only the optimal solutions of the mentioned methods but also the adjusted CPU time in pu. The simulation results has revealed that the proposed CBIA method is very efficient for solving the optimal short-term hydrothermal scheduling problem.

Funding
The authors received no direct funding for this research.

Author details
Thang Trung Nguyen
E-mail: nyguentrungthang@tdt.edu.vn
Dieu Ngoc Vo
E-mail: vndieu@gmail.com
ORCID ID: http://orcid.org/0000-0002-1237-0071

Bach Hoang Dinh
E-mail: dinhhhoangbach@tdt.edu.vn

1 Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Viet Nam.
2 Department of Power Systems, Ho Chi Minh City University of Technology, Ho Chi Minh City, Viet Nam.

Citation information
Cite this article as: A cuckoo bird-inspired meta-heuristic algorithm for optimal short-term hydrothermal generation cooperation, Thang Trung Nguyen, Dieu Ngoc Vo & Bach Hoang Dinh, Cogent Engineering (2016), 3: 1266863.

References
Basu, M. (2004). An interactive fuzzy satisfying method based on evolutionary programming technique for multiobjective short-term hydrothermal scheduling. Electric Power Systems Research, 69, 277–285. http://dx.doi.org/10.1016/j.epsr.2003.10.003
Basu, M. (2010). Economic environmental dispatch of hydrothermal power system. International Journal of Electrical Power & Energy Systems, 32, 711–720. http://dx.doi.org/10.1016/j.ijepes.2010.01.005
Basu, M. (2016). Quasi-oppositional group search optimization for hydrothermal power system. International Journal of Electrical Power & Energy Systems, 81, 324–335. http://dx.doi.org/10.1016/j.ijepes.2016.02.051
Binghui, Y., Xiaohui, Y., & Jinwen, W. (2007). Short-term hydrothermal scheduling using particle swarm optimization method. Energy Conversion and Management, 48, 1902–1908.
Das, S., & Bhattacharyya, A. (in press). Symbiotic organisms search algorithm for short-term hydrothermal scheduling. Ain Shams Eng Journal. doi:10.1016/j.asej.2016.06.002
Dieu, V. N., Schegner, P., & Ongsakul, W. (2013). Cuckoo search algorithm for non-convex economic dispatch. IET Generation, Transmission & Distribution, 7, 643–654.
Dubey, H. M., Pandit, M., & Peninigrahi, B. K. (2016). Ant lion optimization for short-term wind integrated hydrothermal power generation scheduling. International Journal of Electrical Power & Energy Systems, 83, 158–174. http://dx.doi.org/10.1016/j.ijepes.2016.03.057
Hota, P. K., Barisol, A. K., & Chakraborti, R. (2009). An improved PSO technique for short-term optimal hydrothermal scheduling. Electric Power Systems Research, 79, 1047–1053. http://dx.doi.org/10.1016/j.epsr.2009.01.001
Kumar, S., & Naresh, R. (2007). Efficient real coded genetic algorithm to solve the non-convex hydrothermal scheduling problem. International Journal of Electrical Power & Energy Systems, 29, 738–747. http://dx.doi.org/10.1016/j.ijepes.2007.06.001
Lakshminarasimman, L., & Subramanian, S. (2006). Short-term scheduling of hydrothermal power system with cascaded reservoirs by using modified differential evolution. IEEE Proceedings - Generation, Transmission and Distribution, 153, 693–700. http://dx.doi.org/10.1049/iptg-20050407
Lakshminarasimman, L., & Subramanian, S. (2008). A modified hybrid differential evolution for short-term scheduling of hydrothermal power systems with cascaded reservoirs. Energy Conversion and Management, 49, 2513–2521. http://dx.doi.org/10.1016/j.enconman.2008.05.021
Mandal, K. K., Basu, M., & Chakraborty, N. (2008). Particle swarm optimization technique based short-term hydrothermal scheduling. Applied Soft Computing, 8, 1392–1399. http://dx.doi.org/10.1016/j.asoc.2007.10.006
Naresh, R., & Sharma, J. (1999). Two-phase neural network based solution technique for short term hydrothermal scheduling. IEEE Proceedings - Generation, Transmission and Distribution, 146, 657–663. http://dx.doi.org/10.1049/ip-gtd:19990855
Nima, A., & Hassan, R. S. (2010). Daily hydrothermal generation scheduling by a new modified adaptive particle swarm optimization technique. Electric Power Systems Research, 80, 723–732. doi:10.1016/j.epsr.2009.11.004
Otero, S. O., & Irving, M. R. (1998). A genetic algorithm modelling framework and solution technique for short term optimal hydrothermal scheduling. IEEE Transactions on Power Systems, 13, 501–518. http://dx.doi.org/10.1109/59.667375
Roy, P. K. (2013). Teaching learning based optimization for short-term hydrothermal scheduling problem considering valve point effect and prohibited discharge constraint. International Journal of Electrical Power & Energy Systems, 53, 10–19. http://dx.doi.org/10.1016/j.ijepes.2013.03.024
Roy, P. K., Pradhan, M., & Paul, T. (in press). Krill herd algorithm applied to short-term hydrothermal scheduling problem. Ain Shams Eng Journal. doi:10.1016/j.asej.2015.09.003
Sinha, N., Chakraborti, R., & Chattopadhyay, P. K. (2003). Fast evolutionary programming techniques for short-term hydrothermal scheduling. IEEE Transactions on Power Systems, 18, 214–220. http://dx.doi.org/10.1109/TPWRS.2002.807053
Sivasubramani, S., & Shanti Swarup, K. (2011). Hybrid DE-SQP algorithm for non-convex short term hydrothermal scheduling problem. Energy Conversion and Management, 52, 757–761. http://dx.doi.org/10.1016/j.enconman.2010.07.056
Soares, S., Lyra, C., & Tavares, H. (1980). Optimal generation scheduling of hydrothermal power systems. IEEE Transactions on Power Apparatus and Systems, PAS-99, 1107–1118. http://dx.doi.org/10.1109/TPAS.1980.319741
Swain, R. K., Barisol, A. K., Hota, P. K., & Chakraborti, R. (2011). Short-term hydrothermal scheduling using clonal selection algorithm. International Journal of Electrical Power & Energy Systems, 33, 647–656. http://dx.doi.org/10.1016/j.ijepes.2010.11.016
Thuan, N. T., & Anh, T. V. (2015). Distribution network reconfiguration for power loss minimization and voltage profile improvement using cuckoo search algorithm. *International Journal of Electrical Power & Energy Systems*, 68, 233–242.

Thang, N. T., & Dieu, V. N. (2015). Modified cuckoo search algorithm for short-term hydrothermal Scheduling. *Electrical Power and Energy Systems*, 5, 271–281.

Thang, N. T., Dieu, V. N., & Anh, T. V. (2015). Cuckoo search algorithm for short-term hydrothermal scheduling. In *Cogent Engineering*. Cogent OA, 3(1), 1266863. http://dx.doi.org/10.1080/23311916.2016.1266863

© 2016 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

Cogent Engineering (ISSN: 2331-1916) is published by Cogent OA, part of Taylor & Francis Group.

Publishing with Cogent OA ensures:
- Immediate, universal access to your article on publication
- High visibility and discoverability via the Cogent OA website as well as Taylor & Francis Online
- Download and citation statistics for your article
- Rapid online publication
- Input from, and dialog with, expert editors and editorial boards
- Retention of full copyright of your article
- Guaranteed legacy preservation of your article
- Discounts and waivers for authors in developing regions

Submit your manuscript to a Cogent OA journal at www.CogentOA.com