Alexandria Engineering Journal (2021) 60, 5431–5461

HOSTED BY

ELSEVIER

Alexandria University

Alexandria Engineering Journal

www.elsevier.com/locate/aej
www.sciencedirect.com

Optimal operation of wind-hydrothermal systems considering certainty and uncertainty of wind

Ly Huu Pham [a], Bach Hoang Dinh [a], Thang Trung Nguyen [b], Van-Duc Phan [c,*]

[a] *Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Viet Nam*
[b] *Power System Optimization Research Group, Faculty of Electrical and Electronics Engineering, Ton Duc Thang University, Ho Chi Minh City, Viet Nam*
[c] *Faculty of Automobile Technology, Van Lang University, Ho Chi Minh City, Viet Nam*

Received 9 December 2020; revised 5 March 2021; accepted 3 April 2021

KEYWORDS

Cuckoo search algorithm;
Wind turbines;
Hydroelectric plants;
Thermal power plants;
Fuel cost;
Hydrothermal system

Abstract This paper proposes a High Performance Cuckoo Search Algorithm (HPCSA) for determining suitable operation parameters of the optimal wind-hydro-thermal system scheduling (OWHTSS) problem. The objective of the problem is to reach the lowest electricity generation cost of thermal power plants (TPPs) and wind power plants (WPPs) while exactly meeting all constraints of TPPs, WPPs and hydroelectric plants (HEPs). HPCSA is formed by applying improvements on the two main techniques of original Cuckoo Search Algorithm (CSA) to cover CSA' drawbacks such as searching random solution spaces, always using two random solutions for getting a jumping step and suffering from slow convergence. HPCSA accompany with CSA, Adaptive CSA (ACSA), Snap-Drift CSA (SDCSA) and Water Cycle Algorithm (WCA) are run for solving four test systems in which the largest and complicated system is comprised of four TPPs, four HEPs and two WPPs with the uncertain wind feature. The result comparisons indicate that HPCSA is superior to applied and previous methods, and other modified versions of CSA in the literature in terms of better cost, higher stability, faster search ability and higher success rate. As a result, it leads to a conclusion that HPCSA is a strong metaheuristic algorithm for solving OWHTSS problem.

1. Introduction

Optimal HTS operation in a short-term period (ST-OHTSO) is one of the most valuable problems in power systems as TPPs and HEPs are integrated in a common system to produce and supply electricity to loads over multi-time periods, 24 single hours or some days within one week. The problem aims to reduce the electricity generation cost of all TPPs and exactly meet all constraints of TPPs, HEPs and the system [1]. ST-OHTSO problem is classified into variable head ST-OHTSO

* Corresponding author.
E-mail addresses: phamhuuly.st@tdtu.edu.vn (L. Huu Pham), dinhhoangbach@tdtu.edu.vn (B. Hoang Dinh), nguyentrungthang@tdtu.edu.vn (T. Trung Nguyen), duc.pv@vlu.edu.vn (V.-D. Phan).

Peer review under responsibility of Faculty of Engineering, Alexandria University.

https://doi.org/10.1016/j.aej.2021.04.025
1110-0168

Nomenclature

a_{si}, b_{si}, c_{si}, α_{si}, β_{si} The i^{th} TPP's cost function coeefficent
a_{hj}, b_{hj}, c_{hj} The j^{th} HEP's discharges coefficients
∂ Randomly produced number in [0, 1]
$\Delta So_s^{Smallest}$ The smallest step size of the s^{th} solution
ΔSo_s^{Medium} The medium step size of the s^{th} solution
$\Delta So_s^{Largest}$ The largest step size of the s^{th} solution
$\Delta Ft_{s1\text{-}s2}$ Fitness deviation of the solution s_1 and solution s_2
e_{wf} Direct price of the wf^{th} WPP ($/MWh)
Ft_{mean} Mean fitness of all solutions
Ft_{min} The lowest fitness of population
Ft_{s1}, Ft_{s2} Fitness value of the two solutions
Ft_s, Ft_s^{new} Fitness value of the s^{th} solution and the s^{th} new solution
g_{wf} Underestimation price of the wf^{th} WPP ($/MWh)
h_{wf} Overestimation price of the wf^{th} WPP ($/MWh)
$I_{hj,m}$, $q_{hj,m}$ Inflow and discharge of HEP j over the m^{th} period
$Iter^{max}$ Maximum number of iterations
m,M Time period index and number of periods
MN_{sp} Maximum number of solution pairs
N_{tp} Number of TPPs
N_{hp} Number of HEPs
N_{csp} Number of potential solution pairs
$P_{si,\max}$, $P_{si,\min}$ Maximum power and minimum power of the i^{th} TPP
$P_{si,m}$ Power of the i^{th} TPP at the interval m
$P_{wwf,m}$ Generated power from of the wf^{th} WPP at the m^{th} period
$P_{swf,m}$ Scheduled power from the wf^{th} WPP at the m^{th} period
$P_{Load,m}$, $P_{Loss,m}$ Power demand and loss over the m^{th} period
$P_{hj,\min}$, $P_{hj,\max}$ Minimum and maximum power of the HEP j
$P_{wwf,\min}$, $P_{wwf,\max}$ Minimum and maximum power of the wf^{th} WPP
$P_{wwf,rate}$ The wf^{th} WPP's rated power
P_s Population size
$q_{hj,\max}$, $q_{hj,\min}$ Maximum and minimum discharge limits of the HEP j
So_s, So_s^{new} The s^{th} old and new solutions
Tol^{low} Lower bound of tolerance
Tol^{up} Upper bound of tolerance
$V_{hj,\min}$, $V_{hj,\max}$ Minimum and maximum reservoir volumes of the HEP j
$V_{hj,m}$ Reservoir volume of the HEP j at the end of the period m
$V_{hj,Available}$ Available water in the HEP j
$V_{hj,Require}$ The j^{th} HEP reservoir volume requested after one operation day
$V_{wwf,m}$ The wf^{th} WPP's wind speed over the m^{th} period
$V_{wwf,ci}$ Lower bound of wind speed for power generation in the wf^{th} WPP
$V_{wwf,co}$ Upper bound of wind speed for power generation in the wf^{th} WPP
$V_{wwf,rate}$ The wf^{th} WPP's rated wind speed

(VH-ST-OHTSO) problem and fixed head ST-OHTSO (FH-ST-OHTSO) problem due to the different models of water head during considered periods. The comer considers the change of head after each interval due to the deviation between inflows and discharge while the latter supposes that the deviation is very small and the head remains unchanged. The unlike head models lead to the different functions of generation, a volume and discharge function for VH-ST-OHTSO and a discharge function for FH-ST-OHTSO problem [2]. In the FH-ST-OHTSO problem, there are two study cases in which the first case ignores reservoir volume constraints but considers the constraint of total available water [1–13] while the second case takes the reservoir volume constraints into account [14–36]. Both limits of reservoirs and reservoir volume balance are the difficult constraints of the second case. So, the second case is more complicated in comparison with the first case and it is selected to be a study case in the paper together with power generation from WPPs with the consideration of wind speed uncertainty. In recent years, clean energy resources, especially wind energy received a huge number of attentions from almost all countries around the world. A general policy of all nations is to reduce the use of power energy from conventional power plants that damage healthy and life of people, produce polluted emissions and increase the green house effects. Instead, renewable energy resources (RESs) such as solar and wind are being encouraged and arising as a highly effective power. Nowadays, more and more applications of RESs are integrated in many engineering fields and operation problems such as the combination of heat, cooling and power system with wind turbines [37], the use of a hybrid wind and photovoltaic system for supplying power energy to buildings [38], optimal flow power problem [39] and economic load dispatch problem [40–41].

In the paper, TPPs, HEPs and WPPs are integrated to produce and transmit electricity to customers for one working day with twenty-four periods in which fixed head model is considered for HEPs. The problem can be called optimal scheduling of WHTS (OWHTSS). The key task of OWHTSS problem is to find the best operation parameters for all plants that lead to the smallest electric generation cost and the exact satisfaction of all constraints from TPPs, HEPs, WPPs and power systems. In the fixed-head model, discharge is a function of hydroelectric generation and other given coefficients. Both discharge and generation are limited within operating boundaries such as the minimum and maximum discharge, and the minimum and maximum generation. In addition, the minimum and maximum volume of reservoirs are also constrained in the fixed head model. To reach good solutions of the OWHTSS problem, we propose a HPCSA method and apply four other popular and effective methods such as Water Cycle algorithm (WCA) [41], Cuckoo Search Algorithm (CSA) [42], Adaptive Cuckoo Search Algorithm (ACSA) [30], and Snap drift cuckoo search algorithm (SDCSA) [43]. It is emphasized that all the five implemented methods are population-based algorithms with different features from evolution algorithms [2,18], and [44,45] and the latest algorithms [46–51]. CSA and WCA are original methods meanwhile the proposed HPCSA in the paper and two other previously developed methods including ACSA [30] and SDCSA [43] are the three modified versions of CSA. Evolution algorithms are comprised of Real Coded Genetic algorithm (RCGA) [44], Nondominated Sorting Genetic Algorithm-II (NSGA-II) [45], Differential Evolution (DE) [2], and Evolutionary Programming (EP) [18]. The latest algorithms consist of Salp Swarm Algorithm (SSA) [46], Sunflower Optimization Algorithm (SFO) [47], Heap-Based Optimizer (HBO) [48], Chimp Algorithm (CA) [49], Jellyfish Algorithm (JA) [50] and Equilibirium Optimizer (EO) [51]. These methods are not effective in solving the OWHTSS problem, especially for the largest systems of the OWHTSS problem with four TPPs, four HEPs and two WPPs. As stated in the study [52], these evolution algorithms had a slow convergence rate and a poor computation effectiveness. Meanwhile, the largest system has a high number of decision variables, a high number of dependent of variables, and a high number of equality constraints regarding reservoir volume balance and power balance. After finding these decision variables, two main issues are the violation of power balance and reservoir volume balance. Volume of reservoir will be corrected if the volume balance constraint is violated. Next, discharge and power generation of hydroelectric plants must be penalized as a result. Then, power balance constraint is checked and penalty term will be applied for the first thermal power plant's power output. The decision variables obtained by using the evolution operator of these evolutionary algorithms at each computation iteration are not optimal due to the constraint violation. The constraint handling effort makes difficulties in generating effective decision variables and it becomes a main shortcoming of these evolution algorithms [52]. The main drawback is also found as applying these latest algorithms and they also cope with the same difficulty as evolutionary algorithms. Disadvantages of the latest methods can be explained as follows:

(1) Use one solution update technique for each iteration and search around current old solutions by randomly using one step or two steps without certain basics. For each iteration, only one new solution generation is created and evaluated. Then, all the existing solutions are compared for retaining the most effective solution set and bring the retained set to the next generation. The diversity of search process is not exploited over the number of iterations. As new solutions are found around their old solutions by adding one or two steps randomly, these methods have to cope with two major shortcomings: 1) search around ineffective solutions many times but search around effective spaces only a few times even one time; 2) the distance from the old solutions to new solutions are calculated by using the same way, either the deviation of two random solutions (called one step) or the deviation of four random solutions (called two steps). In solving the studied problem, control and dependent variables of valid solutions may belong to different search spaces and even these search spaces are so far from each other. If solutions fall into local zones in some first iterations, the applied search mechanism only focus on the zones and there are no enough large steps for jumping out the zone and approaching other zones. So, the main shortcomings avoid finding different valid spaces for different control and dependent variables. As a result, the success rate of the methods for solving the problem is very low.
(2) Use the same method of generating optimal solutions for from the first to the last iteration. These methods easily fall into inefficient spaces with local optimal solutions and hardly escape the trapped spaces. Basically, large search spaces need to be explored in some first iterations and then ineffective spaces must be eliminated for giving more search chances to other effective spaces. There may be many solutions satisfying all constraints of the problem but the solutions with good quality and low cost function are few. So, these algorithms cannot achieve good results as expected.

On the contrary to these evolution and latest algorithms, CSA has two solution update techniques where Lévy flights distribution is employed in the first technique and mutation is the second technique. The major differences between CSA and other methods are Lévy flights distribution and two solution update processes in each iteration. Lévy flights distribution supports the first solution update process in expanding search zones while the mutation technique exploits the zones effectively. To reach the Lévy flights-based large steps, the deviation between the best solution and a considered solution is multiplied by the Lévy flights distribution. The Lévy flights distribution can generate a very high random number, resulting in a very large step. So, new solutions of the first technique can fall into new search zones, which are so far from the previous zones. In the mutation technique, a uniform distribution producing random number within 0 and 1 and the deviation between two random solutions are multiplied to form a smaller step. The new solutions of the second technique are nearby old solutions. If the new search zones found by the first technique have promising solutions, the second technique will seek these promising solutions in the zones more carefully. The combination of exploration and exploitation becomes a powerful tool for handling complex constraints and producing valid solutions with the high effectiveness. CSA was successfully and widely applied for different optimization problems in electrical engineering, such as optimal wind farm design [53], optimal HTS operation [54], fault analysis and detection for wind turbines [55], and efficient energy management in smart buildings and houses [56]. However, there have been a lot of improved versions of CSA so far due to its limitation of searching capability for complex problems containing many local zones and non-differential objective functions. In general, the main disadvantage of CSA is to select the best setting for mutation factor that can balance the performance of the global search and the local search. Thus, a high number of researches has aimed to the weak points of CSA [28–32,43,57–64]. Among the studies, SDCSA utilizes a high modification numbers on main mechanisms and mutation factor. The study pointed out the time-consuming disadvantage due to the different settings of the mutation factor and its solution was to use an adaptive mutation factor instead of setting the mutation factor within 0.0 and 1.0. However, the initial point of the adaptive mutation factor plays an important role and it should be set to three values including 0.0, 0.5, and 1.0. Hence, the adaptive factor may not be a suitable and robust solution for this issue. In addition, the study also proposed three different formulas for local search strategy instead of a sole method like conventional CSA [42]. The conditions for using the three formulas are based on randomizations. Consequently, the improvement level of SDCSA over CSA is not certain and the performance of SDCSA may be low for other problems excluding benchmark optimization functions [43].

In this paper, two modifications are proposed on CSA to form HPCSA with intent to get good solutions, a high success, and a quicker convergence and a high stability. As a result, two main mechanisms of local and global search phases are improved. For testing the performance of HPCSA and other ones, four study cases are performed and discussed. In the study, the novelties are stated as follows:

Develop Wind-Hydro-Thermal Systems optimally scheduled in a short-term period. The fixed water head of hydroelectric plants and valve point loading effects of thermal power plants are considered in the systems
Consider uncertainty of wind characteristic
Propose HPCSA method with powerful performance.

As applying HPCSA for OWHTSS problem, promising results are obtained and the contributions of the study are significant as follows;

1. Successfully solve OWHTSS problem. The suitable electricity generation cost is reached and all constraints are exactly handled
2. The proposed HPCSA can reach the highest success rate of 100% for all systems whereas that of WCA, CSA, ACSA and SDCSA is smaller
3. HPCSA can reach lower electricity generation costs, better stability, and faster speed than WCA, CSA, ACSA and SDCSA.

The organization of this paper are as follows. Literature review is presented in Section 2. The OWHTSS problem's formulation is presented in the Section 3. CSA and HPCSA with proposed mechanisms and modifications are explained in Section 4. The application of HPCSA for the OWHTSS problem is mathematically formulated and detailed in Section 5. Four considered systems, obtained results and analyzed are given in Section 6. Lastly, Section 7 conclude and confirm the value of the work.

2. Literature review

The FH-ST-OHTSO problem has attracted a huge number of researchers and there have been many potential algorithms including conventional and modern algorithms. Applied algorithms were used for the problem to be GSBA [14], NA [14], SAA [15], EPAVs [16–21], PSOVs [22–26] and [33], CSO [27], CSAVs [28–32], IHS [34], COS [35] and CGWO-DA [35]. Among the applied algorithms, only GSBA [14] and NA [14] are not meta-heuristic algorithms and mainly employ Lagrange optimization function. The two mentioned methods suffer from significant shortcomings such as not application for non-differential function and ineffectiveness for large-scale systems. Unlike GSBA and NA, other metaheuristics and their improved versions more successfully deal with the difficulties and reach much better results such as shorter time and higher accuracy. Among a group of four methods including SSA, EPA, CSA and CSO, CSO and CSA seem to be more powerful about the quality of solutions and the success rate but CSO's speed was not reported and concluded. CSA outperformed others in terms of fuel cost, the mean solution, population, and the number of iterations. On the contrary, SSA is the least powerful method with the highest fuel cost and the slowest speed even the applied test was a one TPP-one HEP system operated over six periods. By using different proposed mutation mechanisms, and Gaussian and Cauchy distributions, improved versions of EPA were developed like Improved EPA (IEPA) [19], Fast IEPA (FIEPA) [19], Enhanced FIEPA (EFIEPA) [19], Hybrid EPA (HEPA) [20] and Running Fast EPA (RFEPA) [21]. Finally, the improved EPA methods have found more potential results than EPA in [16–18]. Similarly, CSA was also improved by proposing effective modifications such as the use of one evaluation round, the use of Gaussian, Cauchy and Lévy distributions instead of a uniform distribution, and the applications of modified mutation techniques. Namely, these improved methods include CSA with one evaluation round and Cauchy distribution (OECSA-CD) [28], CSA with one evaluation round and Lévy distribution (OECSA-LD) [28], CSA with Gaussian distribution (CSA-GD) [29], CSA with Cauchy distribution (CSA-CD) [29], CSA with Lévy distribution (CSA-LD) [29], Adaptive CSA (ACSA) [30], Improved CSA (ICSA) [31], Modified CSA (MCSA) [32], and Adaptive Selective CSA (ASCSA) [32]. These CSA variants were developed for the applications to larger scale systems and a higher challenge of a non-differential function. The comparison of results showed Lévy distribution was superior to Cauchy and Gaussian distributions, and one evaluation round could be faster than two evaluation rounds. However, the real robustness of one evaluation was not demonstrated persuasively because only a very simple system with one TPP and one HEP system using differential functions was run. ASCSA has applied a modified mutation technique and it successfully solved three complex problems.

In recent years, WPPs have been integrated with TPPs, HEPs, or HTSs. The optimal power generation of WPPs and TPPs was successfully determined by applying ABCA [40] and WCA [41]. Then, the integration of WPPs and TPPs is expanded by considering HEPs to form optimal WHTS operation problem. Optimal solutions of this problem were successfully reached by NSGA-III [65], BCOA [66], NADA [67], MPSO [68], MIPA [69], and SCA [70]. Generally, all the applied metaheuristics have been run for the purpose of presenting the implementation process and proving a highly successful possibility of satisfying all constraints rather than proving a robustness and a fast search ability. Consequently, the real performance of these methods is still a question.

3. Formulation of optimal wind-hydro-thermal system scheduling problem

The Wind-Hydro-Thermal system is the combination of WPPs and conventional HTSs in order to supply power energy to loads for operation. The combined system is comprised of N_{wf} WPPs, N_{hp} HEPs and N_{tp} TPPs. Fig. 1 shows a typical Wind-Hydro-Thermal system supplying electricity to loads at

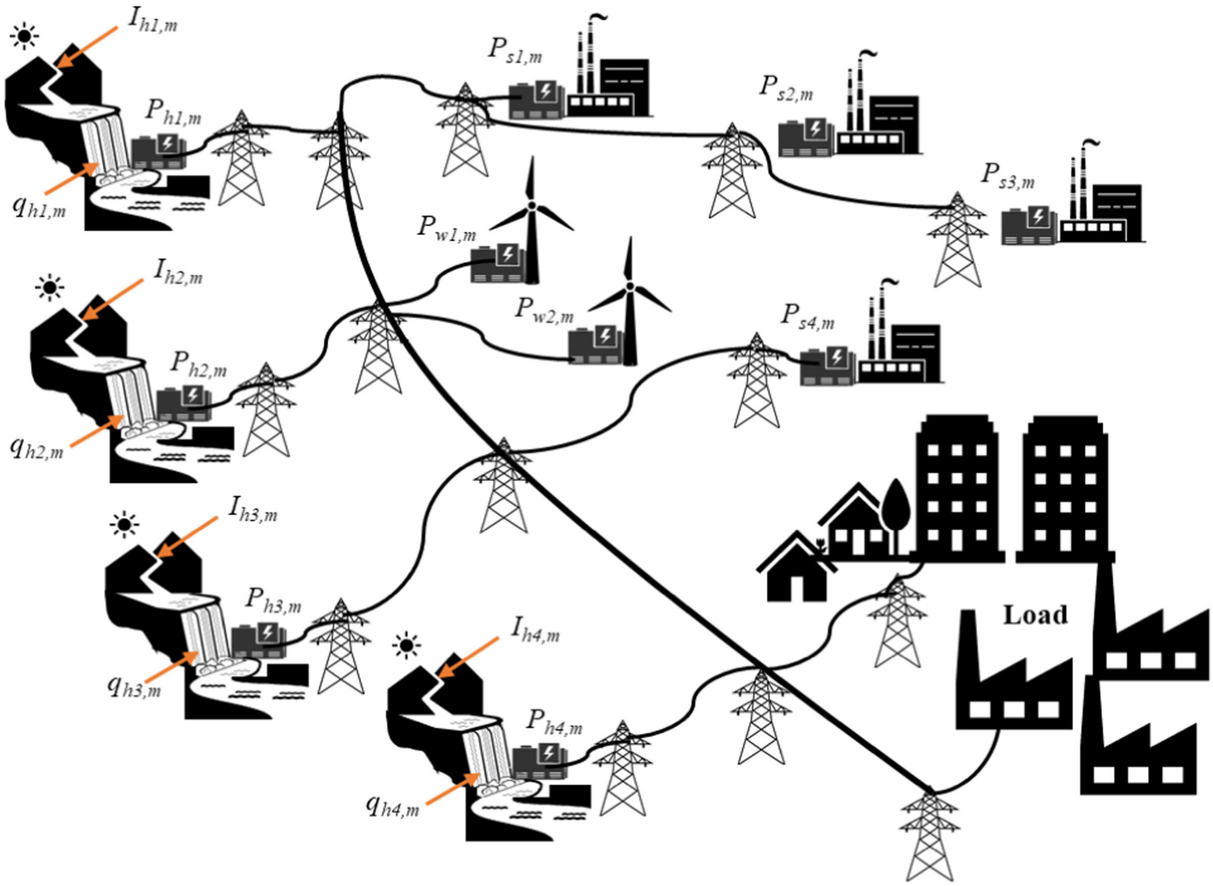

Fig. 1 An example of the wind-hydro-thermal system.

the m^{th} interval. The main objective and constraints of the OWHTSS problem are presented as follows:

3.1. Objective function

3.1.1. The cost modeling for thermal plant

As reported by the Department of Energy [71], the use of energy resources in the world for generating electricity will significantly grow. Especially, fossil fuels used in TPPs account for a big rate of the world electricity supply and they will be out of resources in the future. Furthermore, the fuel sources are very expensive for producing electricity power energy. So, the cost of generating electricity from purchasing the fuels such as coal, gas and oil in TPPs must be considered as a core objective that need to be minimized. The total fuel cost (*TFC*) of all N_{tp} thermal power plants operating in a day with twenty-four hours ($M = 24$) is expressed as a quadratic function below [1]:

$$TFC = \sum_{i=1}^{N_{tp}} \sum_{m=1}^{M} a_{s,i} + b_{s,i} P_{si,m} + c_{s,i} (P_{si,m})^2 \quad (1)$$

As taking the consideration of valve effects in power increase and decrease process, *TFC* in Eq. (1) is replaced with the following discrete form [32].

$$TFC = \sum_{i=1}^{N_{tp}} \sum_{m=1}^{M} \Big(a_{si} + b_{si} P_{si,m} + c_{si} (P_{si,m})^2 + \left| \alpha_{si} \times sin(\beta_{si} \times (P_{si,min} - P_{si,m})) \right| \Big) \quad (2)$$

3.1.2. The cost model for wind power

Wind stations can be owned by the state power companies (SPC) or private power producers (PPP). If wind stations belong to SPC, the electricity generation cost of wind stations is the cheapest and can be ignored owing to no fuel consumption. Otherwise, the contract signed by PPP and independent system operator for purchasing and selling scheduled power should take such cost into account [72]. One of the challenges of encompassing wind power in system grids is an instability of the wind speed, leading the power output of wind stations can be over or under the scheduled power [73]. This requires different costs for wind power to be direct cost, underestimation cost and overestimation cost [73]. These costs based upon the comparison between the real and scheduled outputs of wind power are regarded as a part of the objective function in the problem. The mentioned terms are explained in details as follows:

Direct cost of wind power

The wind power cost bought from WPPs is called as the direct cost. This cost only exists as the WPPs is owned by PPP and the contract of purchasing and selling scheduled power is signed by PPP and independent system operator. If the WPPs are owned by the system operator, this cost is neglected due to no fuel consumption [72,74]. Otherwise, the cost has to be included and mathematically modelled by [74]:

$$C_{dwf}(P_{swf}) = e_{wf}.P_{swf} \quad (3)$$

Underestimation cost of wind power

Normally, the scheduling of real power generation by all available power plants is first formed. The presence of the wind power in power systems can lead to an effect on the initial plan. The cause comes from the uncertainty of the wind speed characteristic, leading to the error forcast's operators. The operators may overestimate or underestimate the wind power availability as compared to scheduled generation power of WPPs. The underestimation appears when the predicated power is smaller than the actual power generated by the WPPs. Therefore, the surplus amount will be wasted and the operators have to pay an additional cost of the surplus wind power energy for PPP. The underestimation cost for the wf^{th} WPP is mathematically formulated by [74,75]:

$$C_{uwf}\left(P_{wwf}-P_{swf}\right)=g_{wf}\left[\begin{array}{c}\left(P_{wwf,rate}-P_{swf}\right)\left[exp\left(-\frac{V_{wwf,rate}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)-exp\left(-\frac{V_{wwf,co}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)\right]\\+\left(\frac{P_{wwf,rate}\cdot V_{wwf,ci}}{V_{wwf,rate}-V_{wwf,ci}}+P_{swf}\right)\left[exp\left(-\frac{V_{wwf,rate}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)-exp\left(-\frac{V_{1}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)\right]\\+\left(\frac{P_{wwf,rate}\cdot c_{wwf}}{V_{wwf,rate}-V_{wwf,ci}}\right)\left\{\begin{array}{c}\Gamma\left[\left(1+\frac{1}{k_{wwf}}\right),\left(\frac{V_{1}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)\right]\\-\Gamma\left[\left(1+\frac{1}{k_{wwf}}\right),\left(\frac{V_{wwf,rate}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)\right]\end{array}\right\}\end{array}\right]$$

where

$$V_1^{k_{wwf}}=V_{wwf,ci}+\frac{P_{wwf}}{P_{wwf,rate}}\left(V_{wwf,rate}-V_{wwf,ci}\right)\tag{4}$$

where P_{swf} and P_{wwf} are the scheduled power and generated power from the wf^{th} WPP, respectively.

Overestimation cost of wind power

In contrast to the underestimation cost, the overestimation cost happens as the predicted power is higer than the actual power produced from the WPPs. It will lead to a power shortage amount that wind stations cannot supply enough power energy to load as signed in the contract. For handling this problem, the best solution for the operator is to purchase some power from the conventional power plants such as HEPs and TPPs. The model of the overestimation cost of the wf^{th} WPP is given by [74,75]:

$$C_{owf}\left(P_{swf}-P_{wwf}\right)=h_{wf}\left[\begin{array}{c}P_{swf}\left[1-exp\left(-\frac{V_{wwf,ci}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)+exp\left(-\frac{V_{wwf,co}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)\right]\\+\left(\frac{P_{wwf,rate}\cdot V_{wwf,ci}}{V_{wwf,rate}-V_{wwf,ci}}+P_{swf}\right)\left[exp\left(-\frac{V_{wwf,ci}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)-exp\left(-\frac{V_{1}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)\right]\\+\left(\frac{P_{wwf,rate}\cdot c_{wwf}}{V_{wwf,rate}-V_{wwf,ci}}\right)\left\{\begin{array}{c}\Gamma\left[\left(1+\frac{1}{k_{wwf}}\right),\left(\frac{V_{1}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)\right]\\-\Gamma\left[\left(1+\frac{1}{k_{wwf}}\right),\left(\frac{V_{wwf,ci}^{k_{wwf}}}{c_{wwf}^{k_{wwf}}}\right)\right]\end{array}\right\}\end{array}\right]\tag{5}$$

3.2. The cost model for hydro plant

A huge difference between HEPs and TPPs is that HEPs must pay a huge initial investment cost but their electricity generation fuel cost is very low and can be neglected [76]. Derived from view point, HEPs will produce electricity to loads without the consideration of cost but their constraints regarding hydraulic issues and generators must be seriously supervised.

3.3. The objective of OWHTSS problem

The intention of the Wind-Hydro-Thermal system operation is to fully exploit the power from HEPs but reduce TFC of TPPs and WPPs as much as possible. As a result, the objective is formed as follows [68]:

$$\begin{aligned}ReduceTFC=&\sum_{i=1}^{N_{tp}}\sum_{m=1}^{M}\Big(a_{si}+b_{si}P_{si,m}+c_{si}(P_{si,m})^2\\&+\left|\alpha_{si}\times sin(\beta_{si}\times(P_{si,min}-P_{si,m}))\right|\Big)\\&+\sum_{wf=1}^{N_{wf}}\sum_{m=1}^{M}\left(\begin{array}{c}C_{dwf,m}\left(P_{swf,m}\right)+C_{uwf,m}\left(P_{wwf,m}-P_{swf,m}\right)\\+C_{owf,m}\left(P_{swf,m}-P_{wwf,m}\right)\end{array}\right)\end{aligned}\tag{6}$$

3.4. Analysis of the wind model

3.4.1. Power generation

Power generation of WPPs is mainly dependent on wind speed, air density and area of blades as shown in Eq. (7). In the equation, area of blades, which is a constant, is a simple factor meanwhile wind speed, which is a variable, is a complicated factor. So, previous studies mainly focused on the determination of wind speed rather than calculating the area [77]. For a general case dependent on wind speed, power generation can be plotted in Fig. 2 and expressed in Eq. (8) [77].

$$PW_{wwf,m}=\begin{cases}0\left(V_{wwf,m}<V_{wwf,ci},V_{wwf,m}>V_{wwf,co}\right)\\\frac{1}{2}\rho\times A\times V_{wwf}{}^{3},\left(V_{wwf,ci}\le V_{wwf,m}\le V_{wwf,rate}\right)\\PW_{wwf,rate},\left(V_{wwf,rate}\le V_{wwf,m}\le V_{wwf,co}\right)\end{cases}\tag{7}$$

$$P_{wwf,m}=\begin{cases}0,\left(V_{wwf,m}<V_{wwf,ci},V_{wwf,m}>V_{wwf,co}\right)\\P_{wwf,rate}\times\frac{\left(V_{wwf,m}-V_{wwf,ci}\right)}{\left(V_{wwf,rate}-V_{wwf,ci}\right)},\left(V_{wwf,ci}\le V_{wwf,m}\le V_{wwf,rate}\right)\\P_{wwf,rate},\left(V_{wwf,rate}\le V_{wwf,m}\le V_{wwf,co}\right)\end{cases}\tag{8}$$

3.4.2. Wind power probability function

The Weibull distribution function is the most popular function and the most widely used function to describe a wind speed frequency curve. As introduced, Cumulative Distribution Function (CDF) and probability density function (pdf) of wind speed have the following forms [78]:

$$CDF=1-exp\left[-\left(\frac{V_{wwf}}{c_{wwf}}\right)^{k_{wwf}}\right],V_{wwf}\ge 0\tag{9}$$

$$pdf=\left(\frac{k_{wwf}}{c_{wwf}}\right)\times\left(\frac{V_{wwf}}{c_{wwf}}\right)^{k_{wwf}-1}\times exp\left[-\left(\frac{V_{wwf}}{c_{wwf}}\right)^{k_{wwf}}\right],V_{wwf}>0\tag{10}$$

where c_{wwf} and k_{wwf} are the scale factor and shape factor at the wf^{th} wind farm, respectively.

Fig. 3 shows Weibull pdf curves with three values of shape factor k_w (k_w = 1, 2 and 3) and a fixed value of scale factor

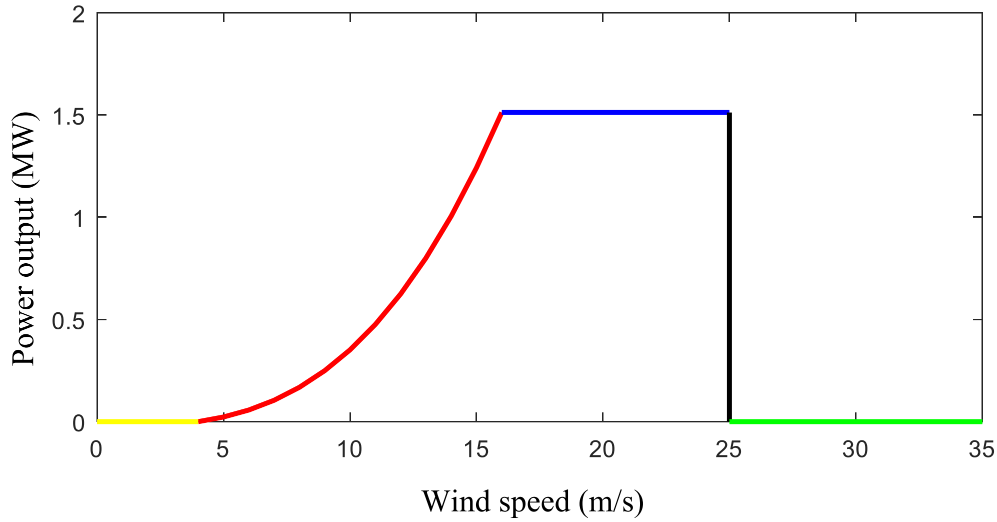

Fig. 2 A typical wind turbine characteristic.

value c_w ($c_w = 10$). The probability of distribution wind speed from 0 to 20 m/s is different for the same wind speed of the three curves. The shape of the curves with $k_w = 2$ and $k_w = 3$ is like and different from that of the curve with $k_w = 1$. In the $k_w = 1$ curve, the probability of lower wind speed is higher and the highest speed of 20 m/s has the lowest probability. In the $k_w = 2$ curve, the wind speeds nearby 7 m/s have the highest probability while the probability of other speeds far from the 7 m/s speed tends to be decreased gradually. The probability of wind speed in the $k_w = 3$ curve is approximately symmetrical through the mean speed of 10 m/s. Clearly, $k_w = 1$ seems to be not suitable for determining the wind speed distribution probability while $k_w = 3$ is more suitable than $k_w = 2$ for the locations with higher mean wind speed. So, if a location with mean wind speed around 7–8 m/s, $k_w = 2$ should be selected. For another case with mean wind speed around 9–10 m/s, $k_w = 3$ is more appropriate than $k_w = 2$.

In order to show the influence of the scale factor c_w on the wind speed probability, $k_w = 2$ is selected while different values of c_w are tried. Fig. 4 displays five curves of Weibull pdf with five values of c_w (c_w = 8, 10, 12, 14, 16). Observing the curves can be recognized that the curves with higher c_w have lower maximum probability but have higher mean probability for all wind speeds. In fact, the peak of the red curve with $c_w = 8$ is higher than 0.2 but the pink curve of $c_w = 16$ is about 0.09. The highest probability of other remaining curves is respectively about 0.16, 0.14 and 0.12. However, as wind speed increases, the curves with higher c_w tend to reach a higher probability than curves with lower c_w. The phenomenon implies that the higher values of c_w are more suitable for locations with high wind speed but lower values of c_w are more appropriate for locations with lower wind speed.

In general, the wind power is calculated by [78]:

$$\begin{aligned} PW_{wwf}(V_{wwf}=0) &= PW(V_{wwf} < V_{wwf,ci}) + P(V_{wwf} > V_{wwf,co}) \\ &= CDF(V_{wwf,ci}) + [1 - CDF(V_{wwf,co})] \\ &= 1 - exp\left[-\left(\frac{V_{wwf,ci}}{c_{wwf}}\right)^{k_{wwf}}\right] + exp\left[-\left(\frac{V_{wwf,co}}{c_{wwf}}\right)^{k_{wwf}}\right] \end{aligned} \tag{11}$$

$$\begin{aligned} PW_{wwf}(V_{wwf}=V_{wwf,rate}) &= PW(V_{wwf,rate} \le V_{wwf} \le V_{wwf,co}) \\ &= CDF(V_{wwf,co}) - CDF(V_{wwf,rate}) \\ &= exp\left[-\left(\frac{V_{wwf,rate}}{c_{wwf}}\right)^{k_{wwf}}\right] + exp\left[-\left(\frac{V_{wwf,co}}{c_{wwf}}\right)^{k_{wwf}}\right] \end{aligned} \tag{12}$$

3.5. The effect of Weibull paramters to wind power costs

To investigate the influence of Weibull pdf parameters on electricity generation costs of WPPs, two WPPs (WPP1 and WPP2) with rated power of 75 MW and 60 MW are selected for simulation [79]. Weibull shape and scale parameters are given by $k_{wf1} = 2$ and $c_{wf1} = 9$ for WPP1, and $k_{wf2} = 2$ and $c_{wf2} = 10$ for WPP2. Moreover, direct cost, underestimation cost and overestimation cost are $e_{wf1} = 1.6$, $e_{wf2} = 1.75$, $g_{wf1} = g_{wf2} = 1.5$ and $h_{wf1} = h_{wf2} = 3$, respectively. Figs. 5 and 6 are plotted to show the variations of direct, underestimation, overestimation and total costs for WPP1 and WPP2, respectively. The two figures have the same shape. Direct cost and overestimation cost are increased whereas underestimation cost is decreased as the power is increased. Direction cost and overestimation cost are zero but underestimation cost is the highest at the lowest power output, i.e. 0 MW. On the contrary, underestimation cost reaches the smallest value of $0 but direction cost and overestimation cost reach the highest at the rated power. However, total cost is always increased as the power output is increased nonlinearly. Consequently, the determination of optimal power output for WPPs with intent of reducing total cost has a significant meaning in power systems with existence of WPPs, TPPs and HEPs.

3.6. The set of constraints of OWHTSS problem

3.6.1. Power balance and generation limits

The first constraint that is seriously taken into account in the problem is power balance between generation side and consumption side. In this paper, power of generation side is the total power of WPPs, HEPs and TPPs while power of con-

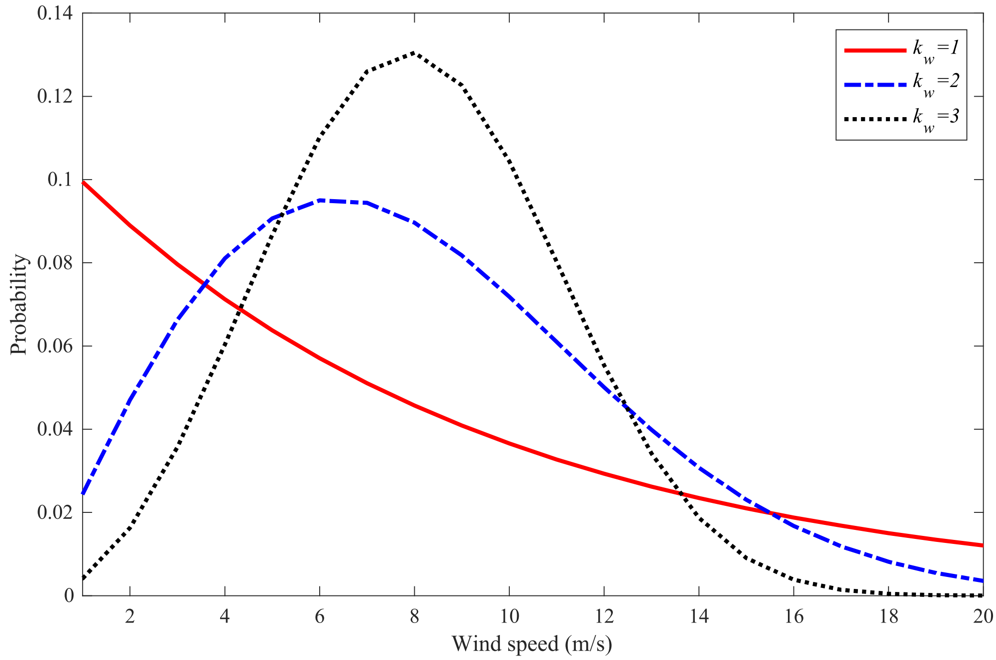

Fig. 3 Weibull pdf with different k_w values.

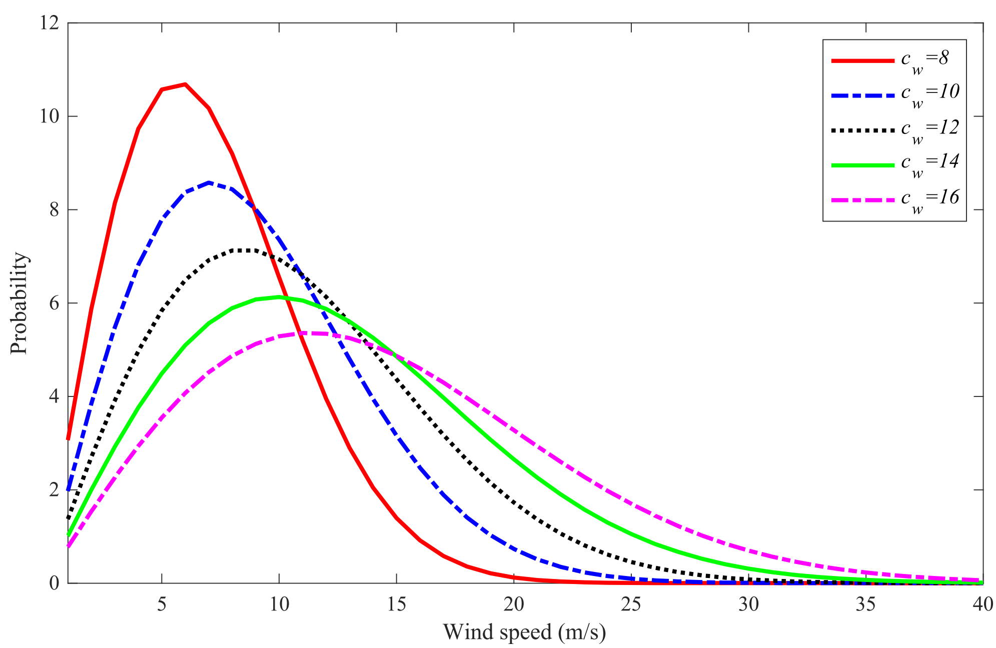

Fig. 4 Weibull PDF with different c_w values.

sumption side is the total of load demand and power loss in conductors. The constraint is mathematically expressed as follows [32]:

$$\sum_{i=1}^{N_{tp}} P_{si,m} + \sum_{j=1}^{N_{hp}} P_{hj,m} + \left(\sum_{wf=1}^{N_{wf}} P_{wwf,m} \right) - P_{Load,m} - P_{Loss,m} = 0 \quad (13)$$

In order to exactly meet the constraint, power generation of each power plant must be satisfied within a predetermined range as the limits below [16]:

$$P_{si,min} \le P_{si,m} \le P_{si,max} \quad (14)$$

$$P_{hj,min} \le P_{hj,m} \le P_{hj,max} \quad (15)$$

$$P_{wwf,min} \le P_{wwf,m} \le P_{wwf,max} \quad (16)$$

3.6.2. Hydraulic constraints

Reservoir volume limits: After each electricity generation hour, reservoir volume is calculated and it must be within a given range as shown in the following inequality [17]:

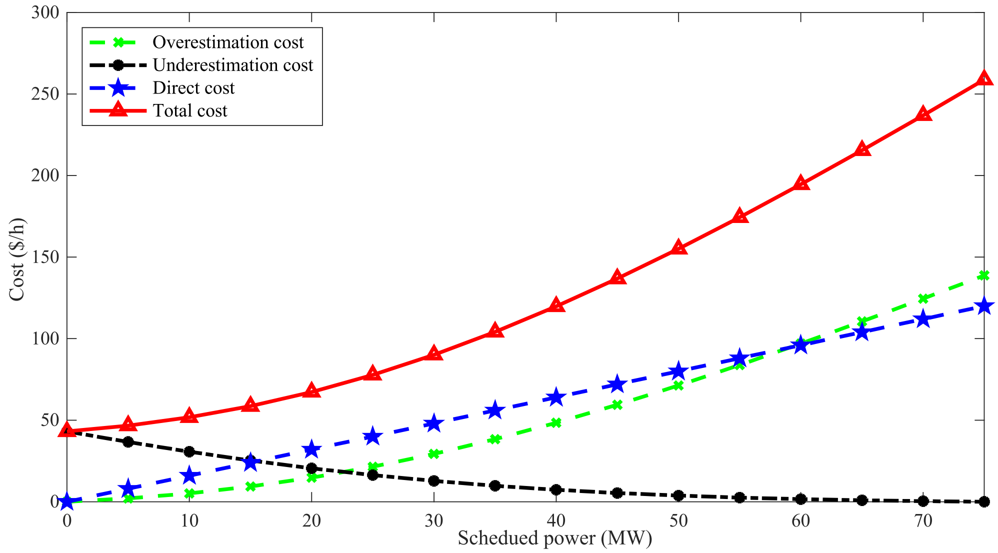

Fig. 5 Cost variation of WPP1.

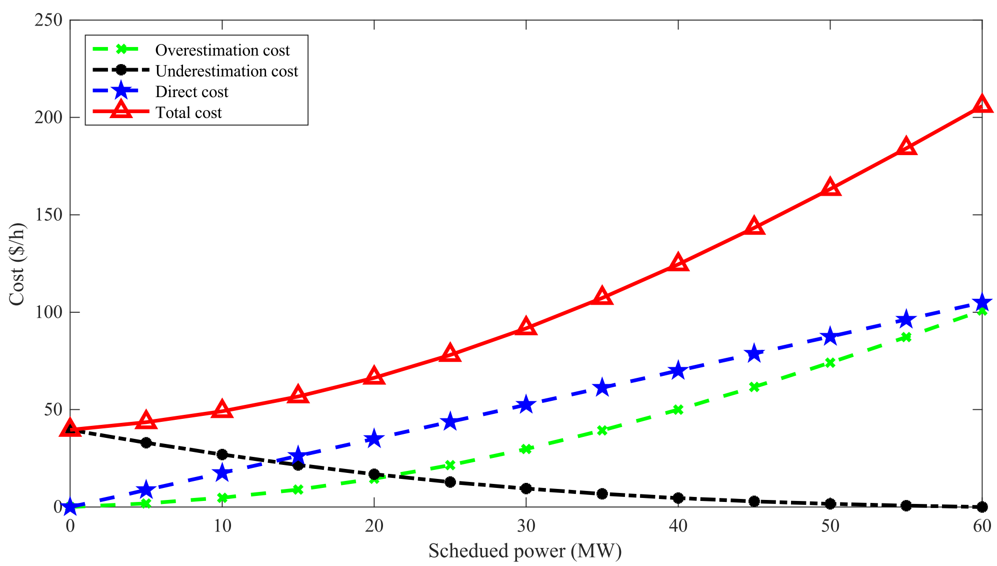

Fig. 6 Cost variation of WPP2.

$$V_{hj,min} \leq V_{hj,m} \leq V_{hj,max}, \tag{17}$$

Discharge limits: Discharge through turbines can drive generators and produce electricity; however, the discharge must be within a predetermined range satisfying physical limits of turbines and generators. So, the constraint is considered as follows [17]:

$$q_{hj,min} \leq q_{hj,m} \leq q_{hj,max} \tag{18}$$

where $q_{hj,m}$ is a function of hydroelectricity generation as follows [80]:

$$q_{hj,m} = a_{hj} + b_{hj}P_{hj,m} + c_{hj}\left(P_{hj,m}\right)^2 \tag{19}$$

Reservoir volume balance constraint: In addition to the restrictions of boundary, volume and discharge are also seriously constrained in reservoir all the time as follows [80]:

$$V_{hj,m-1} - V_{hj,m} + I_{hj,m} - q_{hj,m} = 0, j = 1, 2, \cdots, N_{hp}; m = 1, \cdots, M \tag{20}$$

In the constraint, two special values of the m^{th} interval that must be taken into consideration are the first and the last intervals. As $m = 1$, $V_{hj,m-1}$ will be equal to $V_{hj,0}$ and as $m = M$, $V_{hj,m}$ will be equal to $V_{hj,M}$. The two parts are related to other operation requirements consisting of available reservoir volume at the beginning and retained reservoir volume at the end of a day. The constraints are as follows [80]:

$$V_{hj,0} = V_{hj,Available} \tag{21}$$

$$V_{hj,M} = V_{hj,Require} \tag{22}$$

4. The proposed HPCSA method

4.1. Classical Cuckoo search Algorithm (CSA)

CSA is a metaheuristic algorithm with two different generation mechanisms for updating solutions and one selection mechanism for retaining promising solutions and abandoning ineffective solutions. The first generation mechanism is based on Lévy flights random walk around the current solution meanwhile the second generation mechanism is based on mutation technique similarly in DE. Lévy flights random walk can produce change steps with high size, expanding search zones whereas mutation mechanism focuses on local search with a smaller size step. The two mechanisms are formulated as follows [42]:

$$So_s^{new} = So_s + \alpha_0 \times L\overset{\hat{A}'}{e}vy(\xi) \otimes (So_s - Gbest) \tag{23}$$

$$So_s^{new} = \begin{cases} So_s if r_s \geq M_F \\ So_s + \partial \times (So_{r1} - So_{r2}) else \end{cases} \tag{24}$$

In Eq. (23), α_0 is the scaling factor within 0 and 1; Lévy (ξ) is the Lévy distribution [31], and *Gbest* is the best candidate solution in population. In Eq. (24), r_s is the random parameter in [0,1]; M_F is a selected factor within [0, 1], called mutation factor; and So_{r1} and So_{r2} are two randomly chosen candidate solutions from the population.

4.2. High performance Cuckoo search Algorithm (HPCSA)

As pointed out in [2,43], CSA can reach solutions close to the global optimum solution with low possibility, and CSA must be run by setting high values to population and the iteration number. This manner leads to the long simulation time and restricts the wide application for high dimension optimization problems. The exploration phase may be ineffective as employing the best solution for newly updating nearby each old solution [63] while the exploitation phase is stated to be easily fallen into local optimal zones due to the mutation technique like the DE algorithm [43]. In addition, M_F selection is one of the highest disadvantages of CSA within the range of 0 and 1.0 [64]. Thus, in the paper, making exploitation phase more efficient and simplifying the settings of M_F are determined to be the leading duty.

4.2.1. The first modification on two solution update techniques

CSA can solve discontinuous benchmark functions more successfully than PSO and GA [42] thank to the Lévy flights-based exploration mechanism and mutation-based exploitation mechanism. The comer mechanism is applied for exploiting very large search zones but the later mechanism is employed for focusing on smaller search zones. The jumping steps based on the deviation of So_{r1} and So_{r2} as shown in Eq. (24) are utilized to generate new solutions for all iterations of one dependent trial run. The fact that solutions tend to be improved better and move to nearby global optimum solutions as iteration reaches to its maximum value. In addition, solution searching space is normally expanded at the beginning, and it is narrowed at the last iteration. Hence, using the two solutions-based jumping steps for all iterations can be an ineffective approach. Consequently, this paper suggests using different sizes for jumping steps that can be very small, medium, relatively high or very large jumping steps. For the case that a high number of solutions is distributed far from each other, Eq. (24) is retained. But for another case with many solutions close each other, the largest step is employed instead of the smallest step in Eq. (24). For another remaining case with a medium number of solutions close each other, the medium step is recommended for utilization. For the three considered cases, the smallest, medium, and largest steps are defined as follows:

$$\Delta So_s^{Smallest} = (So_{r1} - So_s) \tag{25}$$

$$\Delta So_s^{Medium} = (So_{r1} - So_s) + (So_{r2} - So_s) \tag{26}$$

$$\Delta So_s^{Largest} = (So_{r1} - So_s) + (So_{r2} - So_s) + (So_{r3} - So_s) \tag{27}$$

Where So_{r3} is also a randomly selected solution from population similarly to So_{r1} and So_{r2}.

On the other hand, solution searching spaces should be chosen appropriately and update mechanisms should be exploited effectively. The different solution spaces can result in found solutions with different quality. Hence, selected solutions for update and width of solution spaces around these selected solutions must be determined exactly and effectively. Clearly, the three steps above are the diversity of solution space width including narrow width, relatively large width, and large width. In addition to the width, selected solution spaces must be also diversified by using the best solution *Gbest* and each candidate solution So_s. As a result, three formulas below are used for the exploration phase in the proposed HPCSA method.

$$So_s^{new} = So_s + \alpha_0 \times \Delta So_s^{Smallest} \otimes L\overset{\hat{A}'}{e}vy(\beta) \tag{28}$$

$$So_s^{new} = So_s + \alpha_0 \times \Delta So_s^{Medium} \otimes L\overset{\hat{A}'}{e}vy(\beta) \tag{29}$$

$$So_s^{new} = Gbest + \alpha_0 \times \Delta So_s^{Largest} \otimes L\overset{\hat{A}'}{e}vy(\beta) \tag{30}$$

The three equations above are also the three methods of selecting search space and search width. If all the three methods are applied for each old solution' update progress, the success of finding highly effective solution can be very high. But, the core disadvantage of the use of three methods is time consuming for simulation process and it can fall into premature convergence issue that need to be avoided seriously. Consequently, only one method is selected for update and selection condition of the sole method must be established for the exploration phase or exploitation phase. The selection condition of the sole method is described as follows:

In the first stage, the number of close solution pairs and the number of all solution pairs, N_{csp} and MN_{sp}, are obtained by Algorithm 1.

Algorithm 1. Pseudo Code for determining N_{csp} and MN_{sp}

```
Ncsp = 0; MNsp=0;
for s1= 1 to (PS − 1)
    for s2= (s1 + 1) to Ps
        Calculate average fitness function Ftmean of the whole population
        Calculate ΔFt_{s1-s2} by:
                    ΔFt_{s1-s2}=|Ft_{s1}-Ft_{s2}|

                MNsp= MNsp+1;
            If ΔFt_{s1-s2}≤Ftmean-Ftmin
                Increase Ncsp using Ncsp = Ncsp + 1
            Else
        Keep Ncsp using Ncsp = Ncsp + 0
        end
    end
end
```

In the second stage, calculate the ratio Tol as follows

$$Tol = \frac{N_{csp}}{MN_{sp}} \tag{31}$$

Finally, two tolerances consisting of lower tolerance Tol^{low} and upper tolerance Tol^{up} are selected for determining the update method as follows:

$$So_s^{new} = \begin{cases} So_s + \partial \times \Delta\Delta So_s^{Smallest}, Tol < Tol^{low} \\ Gbest + \partial \times \Delta So_s^{Largest}, Tol > Tol^{up} \\ So_s + \partial \times \Delta So_s^{Medium}, otherwise \end{cases} \tag{32}$$

It is clear that Tol^{low} and Tol^{up} directly influence determined solution spaces and search space width. As observing from Algorithm 1 and Eq. (31), Tol is within the range from higher than 0 to 1.0. As a result, Tol^{low} and Tol^{up} should be set to the range from 0 and 1.0 and Tol^{low} must be less than Tol^{up}.

4.2.2. The second modification

In the second modification, M_F is fixed at 1 for updating all old solutions. This modification seems to be simple, but its contributions to the improvement level of the proposed HPCSA is high. The contributions are summarized as follows:

1. Reduce simulation time to tune M_F.
2. Take the highest possibility of producing new solutions for all old solutions.
3. Increase the probability of producing promising solutions.

Nevertheless, the applied mechanism is not significantly effective if the next modification imposed on the exploitation phase is not applied. In fact, the second modification highly affects the searching performance, producing high quality solutions. Furthermore, if the update for all current new solutions becomes ineffective, the selection technique can retain better ones (which are current old solutions) and abandon worse ones (which are newly produced solutions). The selection technique is expressed as follows:

$$So_s = \begin{cases} So_s^{new}, Ft_s^{new} < Ft_s \\ So_s, otherwise. \end{cases} \tag{33}$$

As shown in Eq. (33), if all solutions are newly updated to produce So_s^{new} and the newly found solutions are worse than their old solutions So_s, the equation retains the old ones and abandons these new ones.

5. The implementation of the proposed HPCSA method for OSWHT problem

5.1. Selection of control and dependent variables

As shown in Problem Formulation section, the number of operation parameters regarding HEPs, TPPs and WPPs and the number of constraints associated with these parameters are high. However, not every parameter needs to be determined by using the proposed HPCSA method. All parameters can be classified into known parameters and unknown parameters. The known parameters are input data of the problem and they are used to determine the unknown parameters where the proposed HPCSA is in charge of finding these unknown parameters. The main unknown parameters that need to be determined in HEPs, TPPs and WPPs are as follows:

- $q_{hj,m}$, $V_{hj,m}$, $P_{hj,m}$ ($j = 1, \ldots, N_{hp}$; $m = 1, \ldots, M$)
- $P_{si,m}$ ($i = 1, \ldots, N_{hp}$; $m = 1, \ldots, M$)
- $P_{wwf,m}$ ($wf = 1, \ldots, N_{wf}$; $m = 1, \ldots, M$)

In oder to solve the problem successfully, the main operation parameters have to be divided into control variables and dependent variables. One of the successful factors for solving the problem is the selection of control variables and dependent variables. In the problem, inequality constraints are presented in Eqs. (14)-(18) while equality constraints are shown in Eqs. (13) and (20). These inequality constraints are considered to bring operation parameters to allowable operation ranges while the consideration of these equality constraints aim to satisfy the general requirements of the whole system and the whole hydraulic constraints of reservoirs. For dealing with the constraints, two variable types including control variables and dependent variables are defined and included in each solution. Control variables are directly added in solutions and their limitations are always warranted by using lower bound and upper bound. There are two ways to satisfy the bounds of the control variables such as either initializing the control variables within the lower bound and upper bound or setting them to the lower bound (if they are smaller the bound) and the upper bound (if they are higher than the bound). On the contrary, dependent variables are not direct variables in the solution and their bounds are not treated as the same way of the control variables. In general, some of dependent variables are used as slack variables and their main role is to handle equality constraints exactly. Thanks to their role, all equality constraints are always exact but the dependent variables may violate their limitations. Their violation cannot be repaired as the control variables but they are penalized in the fitness function. A successful selection of control varibles and dependent variables is expressed as follows:

- Control variables: $P_{si,m}$ (i = 2, …, N_{tp}; m = 1, …, M); $V_{hj,m}$ (j = 1, …, N_{hp} and m = 1, …, M−1) and $P_{wwf,m}$ (wf = 1, …, N_{wf}; m = 1, …, M)
- Dependent variables: $q_{hj,m}$, $P_{hj,m}$ (j = 1, …, N_{hp}; m = 1, …, M) and $P_{s1,m}$ (m = 1, …, M)

As a result, the control variable number (N_{cvs}) and the dependent variable number (N_{dvs}) are obtained by:

$$N_{cvs} = \left(N_{tp} - 1\right).M + (M - 1).N_{hp} + N_{wf}.M \tag{34}$$

$$N_{dvs} = 2.N_{hp}.M + M \tag{35}$$

In addition, the number of equality constraints (13) and (20) in Section 3 is also concerned in this paper to evaluate the complex level of a solved system. Basically, a day with twenty-four hours is an optimal horizon and there will be 48 equality constrains for the case.

5.2. Determination of dependent variables

Among dependent variables, $P_{s1,m}$ (m = 1, 2, …, M) and $q_{hj,m}$ (j = 1, …, N_{hp} and m = 1, …, M) are slack variables used to guarantee the exactness of equality constraints (13) and (20) whereas other dependent variables $P_{hj,m}$ (j = 1, …, N_{hp} and m = 1, …, M) do not act as slack variables. The process of determining the dependent variables and exactly satisfying the equality constraints (20) and (13) is performed as follows:

Step 1: $V_{hj,0}$ and $V_{hj,M}$ (j = 1, …, N_{hp}) are set to given parameters $V_{hj,Available}$ and $V_{hj,Require}$ as shown in formulas (21) and (22).

Step 2: $P_{si,m}$ (i = 2, …, N_{tp}; m = 1, …, M), $V_{hj,m}$ (j = 1, …, N_{hp} and m = 1, …, M−1) and $P_{wwf,m}$ (wf = 1, …, N_{wf}; m = 1, …, M) are found by using initialization or new solution generation mechanisms of the proposed method.

Step 3: Currently, $V_{hj,m}$ (j = 1, …, N_{hp} and m = 1, …, M) are known and substituted in the constraint (20) to calculate discharge $q_{hj,m}$ (j = 1, …, N_{hp} and m = 1, …, M) as follows:

$$q_{hj,m} = V_{hj,m-1} - V_{hj,m} + I_{hj,m} \tag{36}$$

Step 4: The discharge function (19) is used to find $P_{hj,m}$ (j = 1, …, N_{hp} and m = 1, …, M) as follows:

$$P_{hj,m} = \frac{\left(b_{hj} \pm \sqrt{\left[b_{hj}{}^2 - 4\left(a_{hj} - q_{hj,m}\right) \times c_{hj}\right]}\right)}{2 \times c_{hj}}; P_{hj,m} \geq 0 \tag{37}$$

Step 5: For solving the equality constraint (13), $P_{s1,m}$ (m = 1, 2, …, M) acts as slack variables and it is found by using the constraint as the following formula.

$$P_{s1,m} = \sum_{i=2}^{N_{tp}} P_{si,m} + \sum_{j=1}^{N_{hp}} P_{hj,m} + \left(\sum_{wf=1}^{N_{wf}} P_{wwf,m}\right) - P_{Load,m} - P_{Loss,m} \tag{38}$$

In the equation, power loss of the whole system is neglected while $P_{Load,m}$ is the input data of the problem. In addition, $P_{si,m}$ (m = 1, 2, …, 24 and i = 2, …, N_{tp}) and $P_{wwf,m}$ (wf = 1, …, N_{wf} and m = 1, …, M) are control variables, which are obtained by the proposed method. Unlike other parameters, $P_{hj,m}$ (wf = 1, …, N_{wf} and m = 1, …, M) are dependent variables but they are not slack variables because it is not used to balance any equality constraints.

5.3. Fitness function calculation

After having all slack variables including $q_{hj,m}$ by using (36) and $P_{s1,m}$ by using (38), the balance of the constraints (20) and (13) is converted into the penalty terms regarding the violation of $q_{hj,m}$ and $P_{s1,m}$ in fitness function. If $q_{hj,m}$ and $P_{s1,m}$ are less than lower bounds $P_{si,min}$ and $q_{hj,min}$ or higher than upper bounds $P_{si,max}$ and $q_{hj,max}$, penalty terms are calculated as follows:

$$\Delta q_{hj,m} = \begin{cases} \left(q_{hj,m} - q_{hj,max}\right)^2, if q_{hj,m} > q_{hj,max} \\ \left(q_{hj,min} - q_{hj,m}\right)^2, if q_{hj,m} < q_{hj,min} \\ 0 otherwise \end{cases} \tag{39}$$

$$\Delta P_{s1,m} = \begin{cases} \left(P_{s1,m} - P_{s1,max}\right)^2, if P_{s1,m} > P_{s1,max} \\ \left(P_{si,min} - P_{s1,m}\right)^2, if P_{s1,m} < P_{s1,min} \\ 0 otherwise \end{cases} \tag{40}$$

In the two equations above, $\Delta q_{hj,m}$ and $\Delta P_{s1,m}$ are the two penalty terms associated with the violation of $q_{hj,m}$ and $P_{s1,m}$. In addition, the dependent variables $P_{hj,m}$ are also checked and penalized if lower or upper bound is violated. The penalty term $\Delta P_{hj,m}$ associated with the violation of $P_{hj,m}$ is determined by applying the following model.

$$\Delta P_{hj,m} = \begin{cases} \left(P_{hj,m} - P_{hj,max}\right)^2, if P_{hj,m} > P_{hj,max} \\ \left(P_{hj,min} - P_{hj,m}\right)^2, if P_{hj,m} < P_{hj,min} \\ 0, otherwise \end{cases} \tag{41}$$

After determining all variables and all penalty terms, fitness function of each solution s (Ft_s) is calculated to evaluate quality of available solutions. Ft_s is obtained by adding objective TFC to the sum of penalty terms. TFC is easily found by applying Equation (6) while penalty terms are obtained by using Equations (39)-(41). $V_{hj,0}$ and $V_{hj,M}$ are always within the allowed range because they are respectively equal to known parameters, $V_{hj,Avaiable}$ and $V_{hj,Require}$. As a result, fitness function of each solution s is calculated by the following equation

$$Ft_s = TFC + K_1 \times \sum_{j=1}^{N_{hp}} \sum_{m=1}^{M} \Delta q_{hj,m} + K_2 \times \sum_{j=1}^{N_{hp}} \times \sum_{m=1}^{M} \Delta\Delta P_{hj,m} + K_3 \times \sum_{m=1}^{M} \Delta P_{s1,m} \tag{42}$$

where K_1, K_2, and K_3 are the penalty coefficients.

5.4. Correction for violated new solutions

After updating new solutions, newly found decision variables $P_{si,m}$ ($i = 2, \ldots, N_{tp}$; $m = 1, \ldots, M$),$V_{hj,m}$ ($j = 1, \ldots, N_{hp}$; $m = 1, \ldots, M-1$) and $P_{wwf,m}$ ($wf = 1, \ldots, N_{wf}$; $m = 1, \ldots, M$) can violate operation limits. So, they need verification and correction. They will be set to minimum limits if they are smaller than the limits or they will be set to maximum limits if they have greater values than the limit.

5.5. The procedure of using HPCSA for solving OWHTSS problem

The HPCSA implementation for OWHTSS problem is presented in steps below and summarized in Fig. 7.

Step 1: Assign values for Tol^{up}, Tol^{low}, P_s, $Iter^{max}$ and $M_F = 1.0$

Step 2: Randomly generate the initial population

Step 3: - Determine dependent variables as shown in Section 5.2

- Determine Ft_s as shown in Section 5.3

Step 4: Determine *Gbest* and start the first iteration ($Iter = 1$)

Step 5: - Produce So_s^{new} using Eqs. (28)-(30)

- Implement Section 5.4 for fixing the new solutions

Step 6: - Determine dependent variables as shown in Section 5.2

- Determine Fts as shown in Section 5.3

Step 7: Compare $So_s^{new}$and $So_s$to retain better one using Eq. (33)

Step 8: Calculate N_{csp} and MN_{sb} using Algorithm 1 and Tol using Eq. (31)

Step 9: - Produce So_s^{new} using Eq. (32)

- Implement Section 5.4 for fixing the new solutions

Step 10: - Determine dependent variables as shown in Section 5.2

- Determine Fts as shown in

Step 11: Compare $So_s^{new}$and $So_s$to retain better one using Eq. (33)

Step 12 Determine *Gbest*

Step 13: If $Iter < Iter^{max}$, set $Iter = Iter + 1$ and back to Step 5. Otherwise, stop searching and show obtained results.

6. Numerical results

In the section, four different HTSs are solved for reaching optimal operation parameters. The results from HPCSA are compared to those of other implemented methods (WCA, CSA, SDCSA and ACSA) and other previous methods mentioned in literature. The detail of these four systems and the selections of parameters for the proposed HPCSA are presented as follows:

6.1. Parameters of the four test systems and the proposed method

6.1.1. Parameters of the four test systems

In the Problem Formulation section, we have presented and explained objective and constraints of the problem. For better understanding of the four test systems and input data of the systems, main parameters of the problem are summarized and the selection of data for studied systems is explained in detail. The main parameters of TPPs, HEPs, WPPs and load demand are as follows:

i) Data of TPPs include coefficients of cost function of TPPs (a_{si}, b_{si}, c_{si}, α_{si} and β_{si}) and power generation limits of TPPs including $P_{si,\min}$ and $P_{si,\max}$

ii) Data of HEPs include coefficients of water discharge function (a_{hj}, b_{hj} and c_{hj}), power generation limits ($P_{hj,\min}$ and $P_{hj,\max}$), volume reservoir limits ($V_{hj,\min}$ and $V_{hj,\max}$), initial and end reservoir volume ($V_{hj,0}$ and $V_{hj,M}$), and water inflow to reservoir over 24 h ($I_{hj,m}$)

iii) Data of WPPs include rated power ($P_{wwf,rate}$), electricity prices such as direct price (e_{wf}), underestimation price (g_{wf}) and overestimation price (h_{wf}), scale factor (c_{wwf}) and shape factor (k_{wwf}), and wind information such as wind velocity at each period ($V_{wwf,m}$), lower bound ($V_{wwf,ci}$), upper bound ($V_{wwf,co}$) and rated wind velocity ($V_{wwf,rate}$)

iv) Load demand at each period $P_{Load,m}$

All the parameters above have to be suitably selected for the four studied systems by using data from previous studies to assure that the proposed HPCSA and other applied methods can find valid solutions with operation parameters within allowable ranges. Detail of the four systems is as follows:

1. Test System 1: One TPP and one HEP are optimally scheduled over six twelve-hour subintervals [17]. In Appendix, Table A1 shows data of the TPP meanwhile Table A2 and Table A3 show data of the HEP.
2. Test System 2: Four TPPs and four HEPs supply electricity to loads over 24 one-hour periods. Nonconvex fuel cost functions are considered for TPPs. Data of HEPs are modified from the first system. All data of the system are given in Table A4, Table A5, and Table A6 in Appendix. In addition, data of the system is also found in the study [80].
3. Test System 3: This is a modified system by adding two WPPs to Test System 2. Data of the two WPPs are taken from [81] including the rated power (80 MW and 120 MW), rated velocity (15 and 16 m/s), and velocity of wind over 24 h reported in Table A6 in Appendix. The whole data of the system are also reported in the study [80]. Other data of WPPs such as prices, scale factor, shape factor, minimum velocity and maximum velocity are not used in the Test System 3 because the wind velocity is supposed to be predicted correctly.
4. Test System 4: This system is significantly different from Test system 3 by replacing the two WPPs with two new WPPs. The two new WPPs have the rated power of 75 MW and 60 MW, and considers the uncertainty of velocity characteristic. Because Test system 4 considers uncertainty of wind, other parameters (such as prices, scale factor, shape factor, minimum velocity, and maximum velocity) are used for simulation. The data of the two WPPs are taken from [79] and reported in Table A7in Appendix.

6.1.2. Selection of parameters for the proposed HPCSA

As applying the proposed method, four main parameters that need to have suitable values are population P_s, the maximum iteration number $Iter^{max}$, and lower and upper bounds of tolerance Tol^{low} and Tol^{up}. Among the four parameters, P_s and

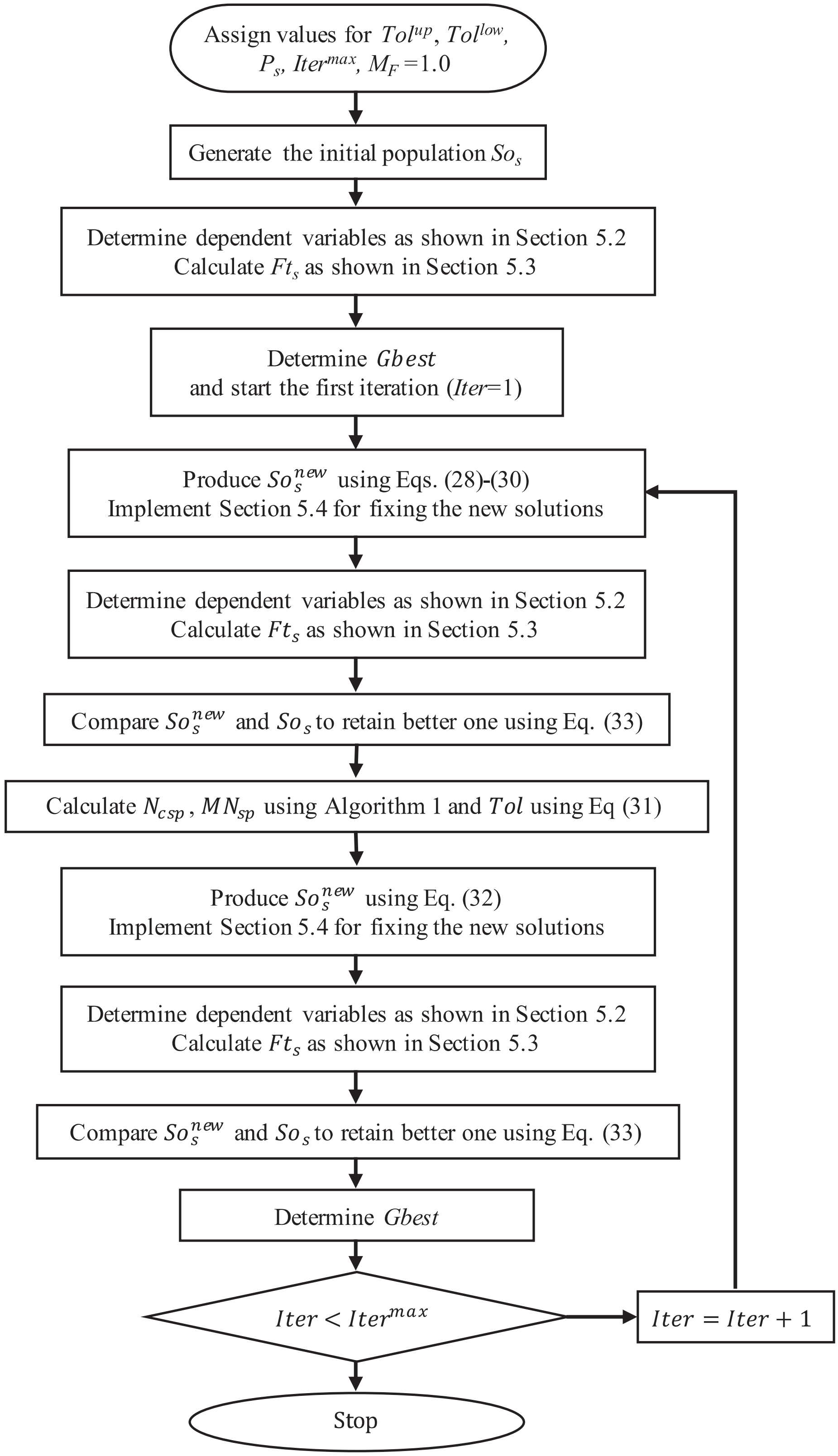

Fig. 7 The iterative algorithm for solving OWHTSS problem.

$Iter^{rmax}$ are called base control parameters while Tol^{low} and Tol^{up} are called advanced control parameters. All metaheuristic algorithms have the same base control parameters as the proposed method whereas the advanced control parameters of different algorithms are unlike. Some methods have advanced parameters but other ones have only base control parameters. Basically, the two groups of control parameters affect the results of solving process but base control parameters have clearer impacts on optimal solutions and implementation time. As applying the proposed method, the selection of the four parameters is very important for reaching the best performance. The population selection directly affects the implementation time and optimal solutions of each iteration while the selection of the maximum iteration number directly affects the whole implementation and the final optimal solution of each run. So, the selection of the two base parameters has the impact on found results of each run including optimal solution reflected via cost function in $ and implementation time in seconds. If the implementation time is the sole purpose of the study, the lowest values of the two parameters are preferred. Otherwise, high values of the two parameters are selected and they can result in good cost functions. However, both the cost function and the implementation time are considered when setting the parameters. Good optimal solutions with suitable cost function are prior but the implementation time has to be not very long. For high dimension systems with a high number of control and dependent variables, and a high number of constraints, the base parameters must be set to high values. On the contrary, lower values are set for the parameters. The settings are successful if all the constraints of the considered problem are satisfied after running the proposed method. And then cost function is selected to be the compared value. In order to calculate N_{cvs} and N_{dvs}, the Eqs. (34) and (35) are used. Among the four studied test systems, the first system is the smallest and the last system is the largest. Although the third and the fourth systems have the same number of HEPs, TPPs and WPPs, the four system is more complicated due to the uncertainty of wind. In fact, power generation of WPPs over 24 h in the third system is calculated by using exactly known wind velocity and the generation is not control or dependent variables. For the first system with one TPP and one HEP operated in six periods, N_{cvs} and N_{dvs} are 5 and 18 whereas the equality constraint number is 12 (6 is for the constraint (13) and 6 is for the constraint (20)). The second system and the third system have the same N_{cvs} and the same N_{dvs}, which are, respectively, equal to 164 and 216. The last system has the highest N_{cvs} with 212 variables but it has the same N_{dvs} as the second system and the third system. Test Systems 2, 3 and 4 have the same equality constraints, which is 48 in which 24 is the constraint (13) and 24 is the constraint (20). As a result, the base control parameters are set to the smallest values for the first system and the highest values for the last system. By experiment, P_s is set to 10 and 20, and $Iter^{max}$ is set to 40 for Test System 1. For Systems 2, 3 and 4, P_s is set to 200 while $Iter^{max}$ is set to 5000 and 10,000 for reaching 50 successful runs. However, the obtained results of the settings must be analyzed and discussed for concluding the most suitable selection.

On the contrary to the base control parameters, advanced control parameters Tol^{low} and Tol^{up} have no impacts on the implementation time. But they still influence the obtained results and their settings also have to be decided reasonably and effectively. Basically, Tol^{low} is chosen in the range between 0 and 0.5, while Tol^{up} is chosen in the range between higher than 0.5 and 1.0. If Tol^{low} is set to 0.5, using the smallest step size is highly possible. Similarly, if Tol^{up} is selected to be close to 0.5 and far away 1.0, the largest step size is highly used. If the two cases take place, the medium step size is insignificantly used. On the other hand, if Tol^{low} is set close to 0 and Tol^{up} is set close to 1.0, the medium step size is highly used. Meanwhile, the use of the smallest and the largest step sizes is very low. Hence, for balancing the use of the three step sizes, Tol^{low} and Tol^{up} should be set to the middle points of their range, i.e. 0.25 for Tol^{low} and 0.75 for Tol^{up}.

The applied methods are simulated on MATLAB program run on a personal computer: Core i7-2.4 GHz, RAM 4 GB. For each study case, each method is implemented for getting 50 successful runs and then success rate is also reported.

6.2. Comparison and discussion on system 1

For this section, the robustness of the proposed HPCSA over CSA and SDCSA is proved as solving System 1. For running all these methods, $P_s = 10$ and $Iter^{max} = 40$ are selected. The collected results are the best, mean and maximum costs for 50 values of fuel cost from 50 successful runs. Then, standard deviation and average simulation time are also reported for comparisons. Table 1 summarizes the comparisons of those parameters. As observing the table, HPCSA is the best method with the lowest minimum, mean and maximum costs, which are, respectively, \$709862.049, \$709900.94, \$11811.47 and 224.5 whereas CSA suffers from the highest minimum cost with \$709881.17 and SDCSA suffers from the highest mean cost with \$712539.43 and the highest maximum costs with \$719078.47. Further calculation indicates that CSA and SDCSA reach greater minimum, mean, and maximum costs than HPCSA by \$19.121 and \$16.491, \$372.33 and \$2638.49, and \$939.15 and \$7267, respectively. About the standard deviation (Std. Dev.), HPCSA reaches the smallest value with 224.5 whereas the worst value of 2380.09 is found by SDCSA. In this regard, CSA with 506.9334 is better than SDCSA but still worse than HPCSA. The results imply that the search process of HPCSA is the most effective and stable among the three executed methods.

With respect to convergence process of the optimum solution, Figs. 8 and 9 are plotted to show the best run and mean of all runs, respectively. In the figures, the curve of CSA is in blue, that of SDCSA is in black and that of HPCSA is in red. In Fig. 8, at the 2nd iteration the minimum cost of HPCSA is less than \$711000 whereas CSA and SDCSA have much higher values. At the fifteenth iteration, CSA and SDCSA still suffer from higher cost than \$711,000. Observing the last iterations from subfigure in Fig. 8, HPCSA reaches the best cost at the 30th iteration but CSA and SDCSA are searching and their costs at the final iteration are still higher than the best cost of HPCSA. In Fig. 9, HPCSA's mean costs almost do not change but those of CSA and SDCSA highly fluctuate from the first iteration to the 10th iteration. The mean costs of CSA and SDCSA continue to be decreased gradually from the 10th iteration to the last iteration. The analysis indicates that HPCSA is much more robust and faster than CSA and SDCSA for the first system.

For further performance investigation, HPCSA and SDCSA continue to solve the system by increasing P_s to 20 and retaining $Iter^{max} = 40$ for comparing to other CSA methods [28–32]. Table 2 shows the results from compared CSA methods. The minimum cost comparison indicates the proposed HPCSA and other CSA methods such as OECSA-LD [28], OECSA-CD [28], CSA-CD [29], CSA-GD [29], CSA-LD [29] and ASCSA [32] can reach the same quality of found

Table 1 Comparison for results obtained by HPCSA, CSA and SDCSA.

Method	Minimum Cost ($)	Average Cost ($)	Maximum Cost ($)	Std. Dev.	Cpu. Time (s)
CSA	709881.17	710273.27	712750.62	506.9334	0.02
SDCSA	709878.54	712539.43	719078.47	2380.09	0.02
HPCSA	709862.049	709900.94	711811.47	224.5	0.02

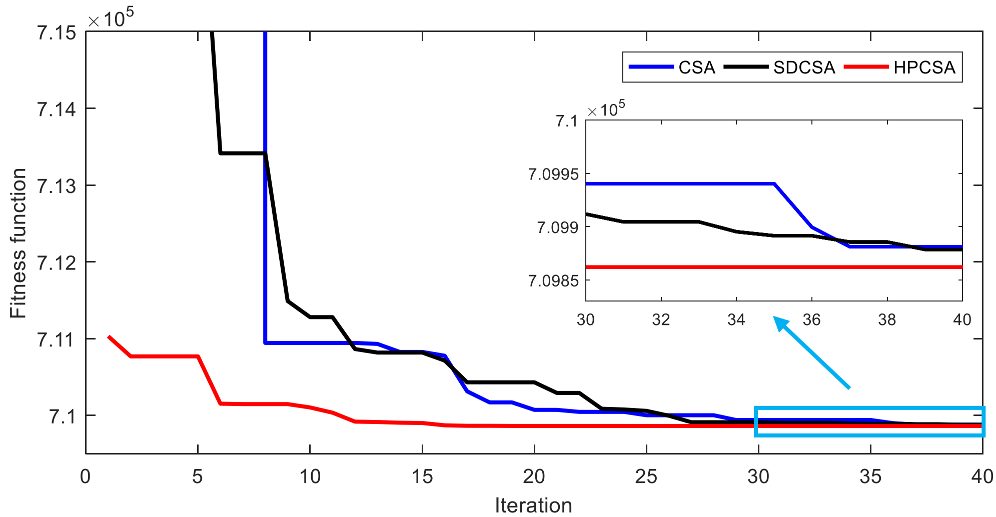

Fig. 8 Fitness vs iteration of the best run obtained by CSA, SDCSA and HPCSA.

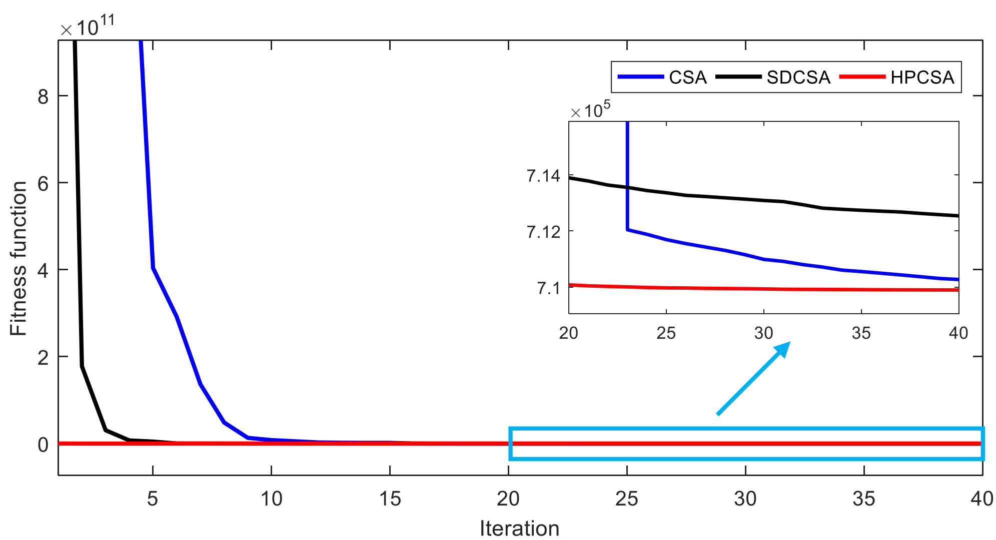

Fig. 9 Average fitness vs iteration of 50 successful runs obtained by CSA, SDCSA and HPCSA.

solutions with the best cost value of $709862.049 whilst MCSA [30], ACSA [30], ICSA [31], CSA [32], MCSA [32] and SDCSA reach higher minimum cost. The mean and maximum costs, and standard deviation of HPCSA are smaller than most methods excluding OECSA-LD [28] and CSA-LD [29]. However, it is noted that OECSA-LD [28] and CSA-LD [29] were run by using $Iter^{max}$ = 300 and $Iter^{max}$ = 400. Furthermore, HPCSA is much quicker than other compared CSA methods because these methods were applied by setting $Iter^{max}$ to the range from 80 to 400. Clearly, HPCSA is the most robust CSA method with the following advantages:

1. Find better solutions than approximately all other CSA methods.
2. Reach more stable solution searching process than all other CSA methods.
3. Use the same population and the same or much smaller number of iterations.

Table 2 Comparison of results obtained by the proposed and other CSA methods.

Method	P_s	$Iter^{max}$	Minimum Cost ($)	Average Cost ($)	Maximum Cost ($)	Std. Dev.	Cpu. Time (s)
OECSA-LD [28]	10	300	709862.049	709862.049	709862.049	0.0000	0.18
OECSA-CD [28]	10	300	709862.049	709862.164	709862.671	0.1830	0.18
CSA-CD [29]	50	400	709862.049	709862.052	709862.060	0.0033	0.28
CSA-GD [29]	50	400	709862.049	709862.050	709862.059	0.0030	0.3
CSA-LD [29]	50	400	709862.049	709862.049	709862.051	0.0005	0.33
MCSA [30]	8	100	709862.054	709905.5	710966.963	189.488	0.13
ACSA [30]	8	100	709862.050	709867.65	709989.94	18.690	0.12
ICSA [31]	10	100	709862.052	709862.13	709862.83	0.1600	0.12
CSA [32]	20	80	709862.052	709888.105	710487.279	95.577	0.034
MCSA [32]	20	80	709862.051	709863.007	709865.793	0.88	0.069
ASCSA [32]	20	40	709862.049	709862.162	709863.242	0.263	0.03
SDCSA	20	40	709865.141	710378.679	713016.151	751.786	0.03
HPCSA	20	40	709862.049	709862.049	709862.069	0.002	0.03

4. Take equal or less simulation time.

In addition, HPCSA and other applied methods, which are not CSA variants, are compared each other as shown in Table 3. The minimum cost reflects the same high-quality solution from HPCSA and others like IEPA [19], FIEPA [19], EFIEPA [19], RFEP [21] and CSO [27] whereas other methods, such as GSBA, SAA, GA and EPA, cannot reach the same best solution. However, it does not mean these methods and HPCSA with equal solution quality are equally strong because those methods are run by setting much higher values to population and the number of iterations. Namely, P_s is set to the range between 30 and 60, and $Iter^{max}$ is set to the range between 70 and 500. Furthermore, the average simulation time of HPCSA is also much shorter than that of SAA [15], EPA [16], EPA [17], IMEPA [19], FIEPA [19], IFEPA [19] and CSO [27]. In fact, that of HPCSA is 0.03 s but that of others is from 4.54 s to 2640 s. Although HPCSA and these methods were run on different computers with different processors, the deviation of simulation time is significant. This is also derived from the deviation of population and iteration number. The process speed comparison of HPCSA with GSBA [14], GA [16] and RFEPA [21] is not performed because these method's computation time (Cpu. Time) was not reported. In summary, HPCSA can reach either better or the same results with others but it is always faster than others for System 1.

6.3. Comparison and discussion on system 2

To investigate the further performance of the proposed HPCSA, a much larger system is established including four HEP and four TPPs taking valve effects into account. Furthermore, twenty-four periods are scheduled for more complicated constraints. These additional factors are the challenges for distinguishing the performance of HPCSA, WCA, CSA [80], MASCSA [80], SDCSA [80]and ACSA.

In the first simulation, three applied methods including WCA, ACSA and the proposed method are simulated by applying the fair comparison criterion with CSA, MASCSA and SDCSA, which were run by setting $P_s = 200$ and $Iter^{max} = 10{,}000$. So, these two settings are reapplied for ACSA and the propose method but $P_s = 400$ and $Iter^{max} = 10{,}000$ are the settings for WCA. However, the results from HPCSA and others are much different where all costs from HPCSA are greatly lower. For better view of the strong search of HPCSA, it is simulated fifty more runs by lowering the iterations to a half, i.e. $Iter^{max} = 5000$. For the sake of simplicity, HPCSA with $Iter^{max} = 10{,}000$ is represented by HPCSA1 and HPCSA with $Iter^{max} = 5000$ is represented by HPCSA2. Results for the two simulations are reported in Table 4. To show the influence of $Iter^{max}$ on the proposed HPCSA performance, the minimum cost, the average cost, the maximum cost and standard deviation of

Table 3 The best cost's comparison with different methods.

Method	Minimum cost ($)	P_s	$Iter^{max}$	Cpu. Time (s)
GSBA [14]	709877.38	–	–	–
SAA [15]	709874.36	–	200	901
GA [16]	709863.56	30	300	–
EPA [16]	709862.06	30	300	8
EPA [17]	709863.29	50	400	2640
IMEPA [19]	709862.05	60	500	159.18
FIEPA [19]	709862.05	60	300	101.4
IFEPA [19]	709862.05	60	150	59.7
RFEPA [21]	709862.05	–	300	–
CSO [27]	709862.05	30	70	4.54
HPCSA	709862.049	10	40	0.03

Table 4 Result comparisons for Test System 2.

Method	WCA	CSA [80]	MASCSA [80]	SDCSA [80]	ACSA	HPCSA1	HPCSA2
P_s	400	200	200	200	200	200	200
$Iter^{max}$	10,000	10,000	10,000	10,000	10,000	10,000	5,000
Minimum cost ($)	35038.51	35640.09	35447.25	35550.06	35273.64	34857.24	35014.25
Average cost ($)	36251.45	36835.21	36355.55	36694.27	36086.4	35903.64	36325.34
Maximum cost ($)	37951.43	38616.82	37533.4	38595.07	36826.15	37391.19	37918.26
Std. dev.	506.3741	595.36	458.1301	628.65	390.761	570.65	726.3221
Cpu. Time (s)	655.5	437.30	457.92	498.71	490.9704	439.13	215.69
Success rate (%)	92.6	71.4	100	100	100	100	100

Noted that HPCSA1 is HPCSA with $Iter^{max}$ = 10,000; HPCSA2 is HPCSA with $Iter^{max}$ = 5000.

HPCSA1 and HPCSA2 are compared each other. These values of HPCSA1 are $34857.24, $35903.64, $37391.19 and 570.65 respectively whereas those of HPCSA2 are $ 35014.25, $36325.34, $37918.26 and 726.3221 respectively. Clearly, HPCSA1 reaches less costs and less standard deviation than HPCSA2 by $157.01, $421.71, $ 527.07 and 155.672, respectively. However, the success rate is equal to 100% for both HPCSA1 and HPCSA2. The comparisons indicate that the efficiency of the HPCSA is higher when setting higher value to $Iter^{max}$ but HPCSA still reaches the best constraint handling ability when $Iter^{max}$ is set to smaller values. Both the better results of HPCSA1 and the worse results of HPCSA2 are compared to other methods in Table 4. For a better view of performance comparison, saving costs of HPCSA1 and HPCSA2 as compared to other methods are calculated and plotted in Figs. 10 and 11, respectively. Different color bars are used in the figures where the blue bars, orange bars and grey bars represents the saving minimum cost, the saving average cost and the saving maximum cost. The values above bars illustrate that HPCSA1 can reach less minimum cost than WCA, CSA, SDCSA, ACSA and MASCSA by $181.27, $782.85, $692.82, $416.4 and $590.01, respectively. The cost reduction is converted into the improvement level [82] for clearer comparisons. HPCSA1 can get the improvement level over WCA, CSA, SDCSA, ACSA and MASCSA to be 0.52%, 2.2%, 1.95%, 1.18%, and 1.66%. Similarly, as compared to these methods, the reduction cost and improvement level of HPCSA2 are $24.26, $625.84, $535.81, $259.39 and $433, and 0.07%, 1.76%, 1.51%, 0.74% and 1.22%. The values in percent reveal that HPCSA can reach better solutions than WCA, CSA, SDCSA, ACSA and MASCSA. It is emphasized that HPCSA2 uses only 5,000 iterations but others use 10,000 iterations. For the case of using the same 10,000 iterations, the improvement of HPCSA1 over the others is considerable. Furthermore, HPCSA reaches lower average cost and lower maximum cost than the three methods, excluding the maximum cost comparison between HPCSA1 and ACSA, the average cost comparison between HPCSA2 and WCA, and the average and maximum cost comparisons between HPCSA2 and ACSA. Computation time from HPCSA2 is much shorter than that of WCA, CSA, SDCSA and ACSA. Meanwhile, the computation time of HPCSA1 is also shorter because the mutation factor is fixed at 1.0 causing the reduction of computation steps. About the success rate, the proposed method, MASCSA, SDCSA and ACSA can reach the best constraint handling ability with the 100% whilst WCA and CSA have weaker ability with the 92.6% success rate and the 71.4% success rate.

The best run from all 50 independent trial runs obtained by WCA, ACSA and HPCSA are shown in Figs. 12 and 13, in which Fig. 12 shows the best run of HPCSA with 10,000 iter-

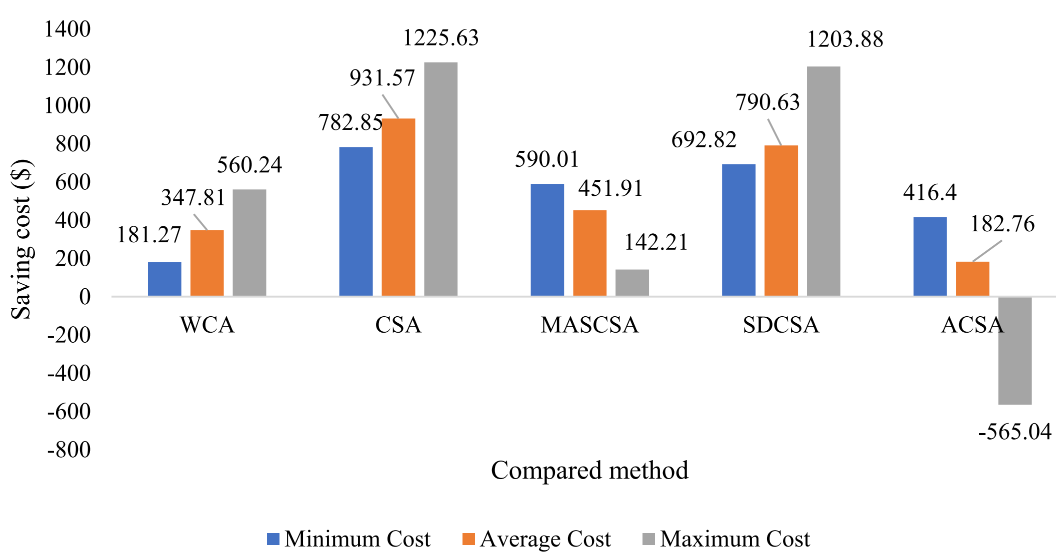

Fig. 10 Saving cost of HPCSA1 as compared to others for Test System 2.

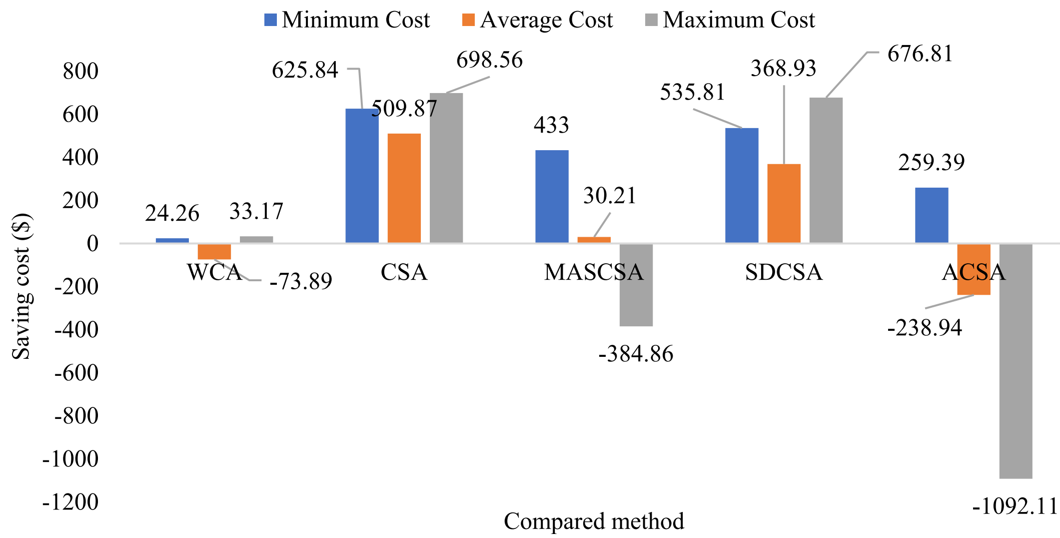

Fig. 11 Saving cost of HPCSA2 as compared to others for Test System 2.

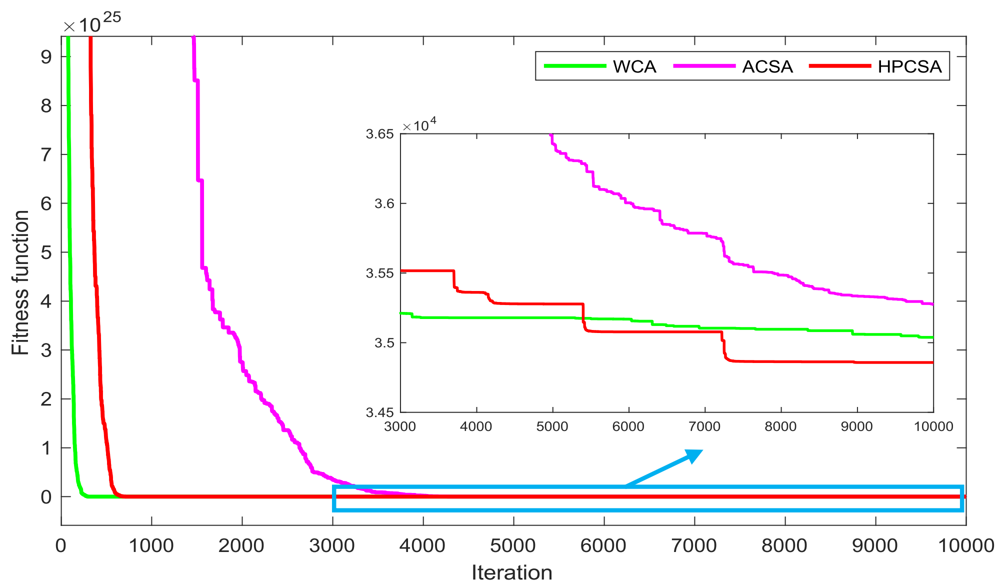

Fig. 12 The feature of the best run obtained by HPCSA and other methods with the same $Iter^{max}$ = 10,000 for System 2.

ations and Fig. 13 shows the best run of HPCSA with 5000 iterations. It is noted that the curve of WCA and ACSA are the same in Figs. 12 and 13 but those are different for HPCSA. Fig. 12 points out that HPCSA is much faster than ACSA but it is not faster than WCA for the whole best run. In fact, all red points are below those of pink points while red points are above green points for the first 5500 iterations. Hereafter the 5500th iteration to the final iteration, red points are below green points. Furthermore, hereafter the 5500th iteration to the final iteration, red points have a high reduction and they always have much less cost than green points. And even the solution of HPCSA at 7000 iterations is much better than that of WCA and ACSA at final iteration. In Fig. 13, HPCSA can be much faster than both ACSA and WCA from the beginning to the end of the search process. Even the solution of HPCSA at the 3000th iteration is much better than that of ACSA at the end, and the soliton of HPCSA at the 5000th iteration is much better than that of WCA at the 10,000 iteration. The cost of 50 runs from HPCSA, WCA and ACSA with $Iter^{max}$ = 10,000 is ranged in the ascending order and plotted in Fig. 14. The curve of HPCSA in red is always below the curves of ACSA and WCA in pink and green excluding the comparison with WCA for the cost of the 47th, 48th and 49th arranged solutions. Clearly, HPCSA is much faster and more effective than ACSA and WCA for the system.

6.4. Comparison and discussion on system 3

In this section, WCA, ACSA and HPCSA are run on System 3. Like results from System 2, the proposed method is simulated two times of 50 runs with two settings, P_s = 200 and

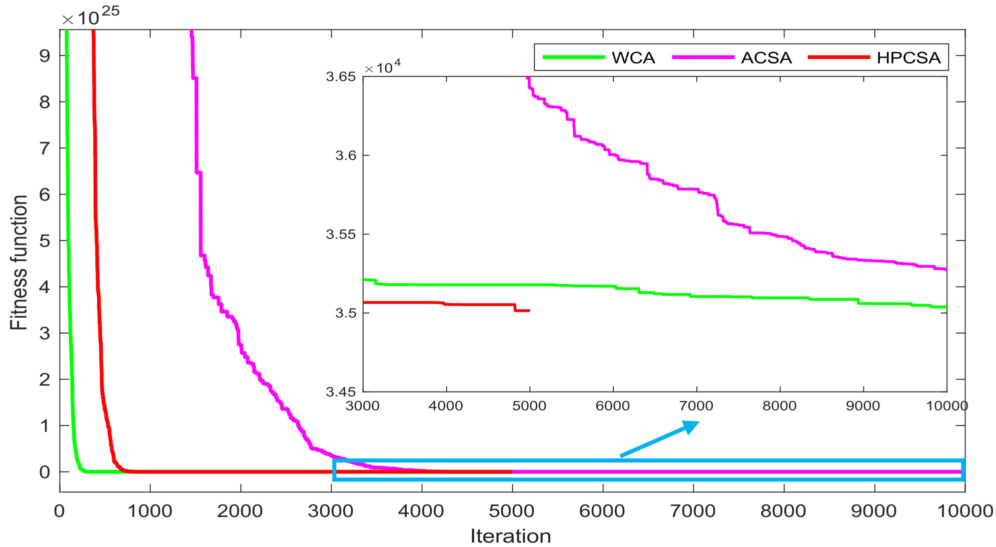

Fig. 13 The feature of the best run obtained by HPCSA and other methods with different settings for $Iter^{max}$.

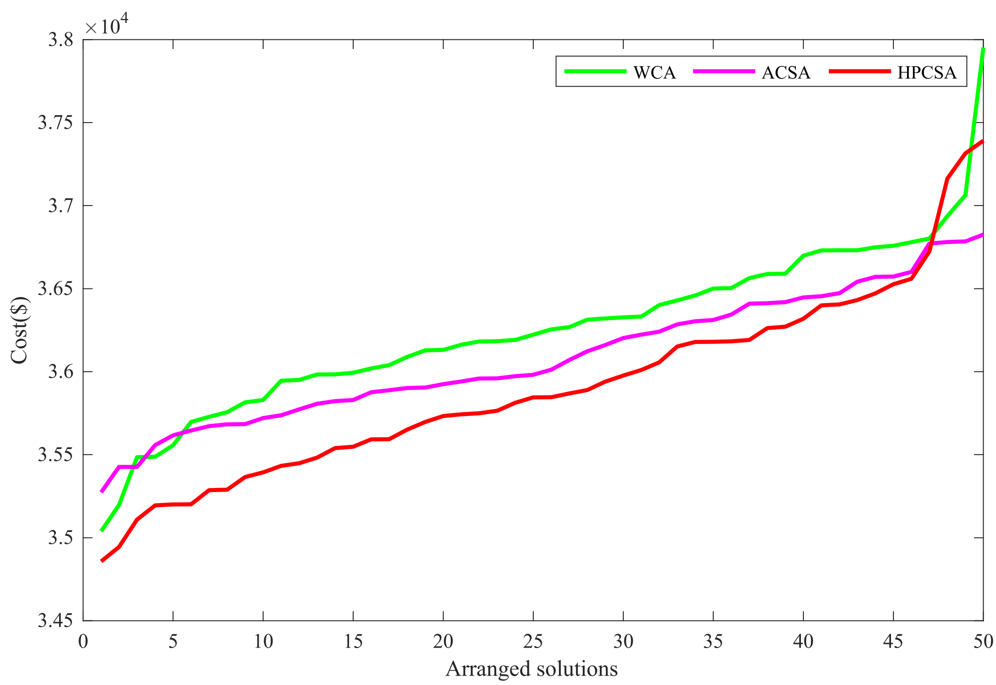

Fig. 14 The cost of 50 runs arranged from the best to worst values for System 2.

Table 5 Result comparisons for test system 3.

Method	WCA	CSA [80]	MASCSA [80]	SDCSA [80]	ACSA	HPCSA1	HPCSA2
P_s	400	200	200	200	200	200	200
$Iter^{max}$	10,000	10,000	10,000	10,000	10,000	10,000	5,000
Minimum cost ($)	27,131.26	27,890.67	27205.16	27,628.06	27,538.26	26,439.00	26,918.94
Average cost ($)	27,952.23	28,682.37	28109.42	28,576.17	28,338.26	27,906.79	28,088.37
Maximum cost ($)	29,117.93	29,793.52	29346.04	29,638.01	30,026.01	29,536.88	29,831.74
Std. dev.	394.2705	471.41	421.88	494.50	476.0928	728.17	775.0605
Cpu. Time (s)	648.4	440.5	462.4	499. 1	490.97	434.6	217.0
Success rate (%)	100	69.4	100	76.9	81.97	100	100

Noted that HPCSA1 is HPCSA with $Iter^{max}$ = 10,000; HPCSA2 is HPCSA with $Iter^{max}$ = 5000.

$Iter^{max}$ = 10,000, and P_s = 200 and $Iter^{max}$ = 5,000 while ACSA and WCA are simulated one time of 50 runs with setting of 400 and 10,000, and 200 and 10,000 for P_s and $Iter^{max}$. The best results of all executed methods accompany with CSA, SDCSA and MASCSA in [80] are reported in Table 5. HPCSA also has two results with different iteration numbers in terms of HPCSA1 and HPCSA2. HPCSA1 and HPCSA2 are compared to others calculating the smaller minimum, mean and maximum costs by plotting Figs. 15 and 16. The bars' direction and numbers above indicate that HPCSA1 can reach much less minimum cost than other ones from \$692.26 to \$1451.67 equivalent to smaller than from 2.55% to 5.20% of others. Although HPCSA2 cannot reach the same better results as HPCSA1, it can reach the cost reduction by from \$212.32 to \$971.73 corresponding to the improvement from 0.78% to 3.48%. Similarly, HPCSA1 can reach a much better average cost and a maximum cost than other methods excluding the maximum cost comparison with WCA whereas HPCSA2 cannot reach a better average cost and a maximum cost than other ones. This issue is also easily understood because HPCSA2 has used only 5000 iterations but others have used 10,000 iterations. Nevertheless, the maximum cost is not the most important comparison factor but the minimum cost.

In addition, HPCSA is more robust than other ones about the ability of dealing with all constraints. Both HPCSA1 and HPCSA have the 100% success rate but other methods with 10,000 iterations such as CSA [80], SDCSA [80] and ACSA only reach the success rate of 69.4%, 76.9% and 81.97%, respectively. WCA and MASCSA [80] also have the same 100% success rate; however, the methods have used 10,000 iterations. Furthermore, HPCSA is always faster than others ones for the two cases of using either 5000 or 10,000 iterations, especially for the comparison with WCA. The computation time is 217.0 and 434.6 s for HPCSA with 5000 and 10,000 iterations but that of others are 648.4, 440.5, 462.4, 499.1 and 490.97 s. In summary, HPCSA can reach better cost than all methods, higher or the same success rate with others but it is much faster than others. As a result, the conclusion is that

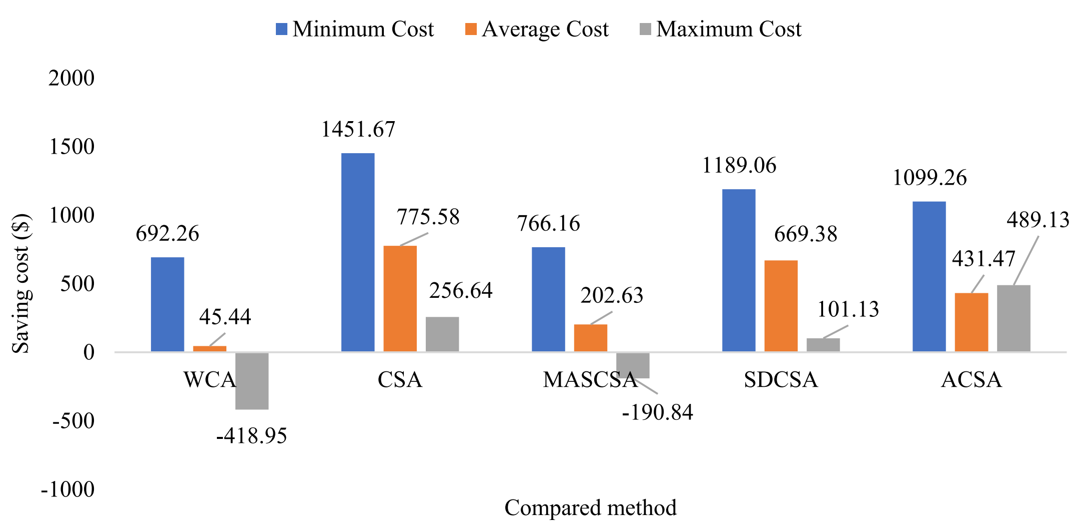

Fig. 15 Saving cost of HPCSA1 as compared to others for Test System 3.

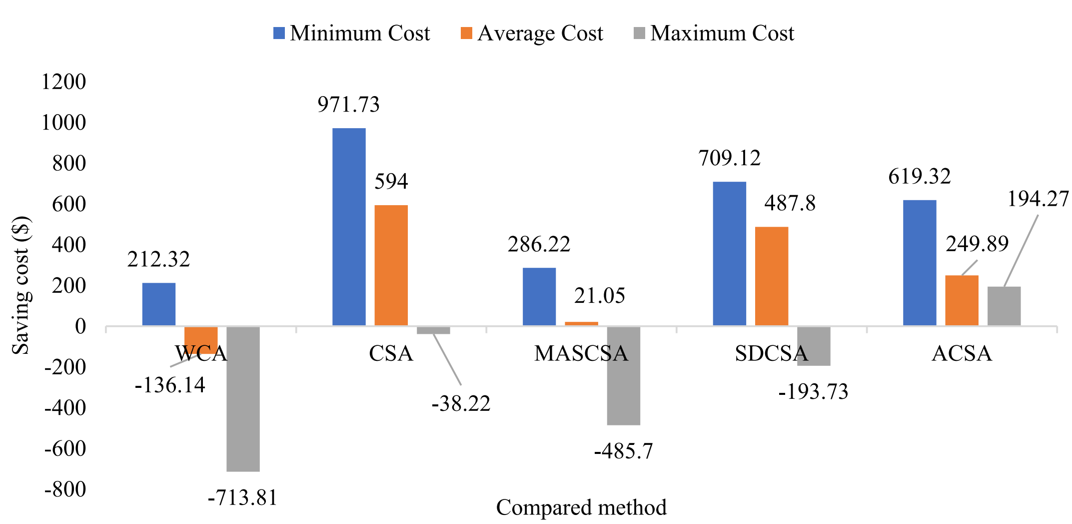

Fig. 16 Saving cost of HPCSA2 as compared to others for Test System 3.

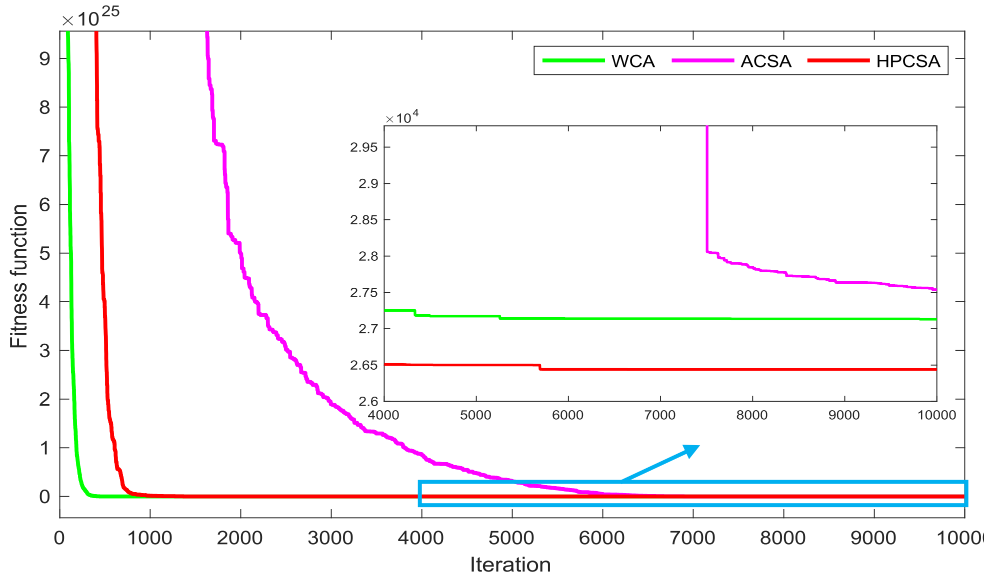

Fig. 17 Fitness vs iteration of the best run with the same $Iter^{max}$ for Test System 3.

HPCSA is more robust than WCA, ACSA and other methods in [80] including CSA, MASCSA and SDCSA for Test System 3.

The best convergence characteristics obtained by WCA, ACSA and HPCSA are depicted in Figs. 17 and 18 for seeing the whole computation process. In Fig. 17, the curves of WCA, ACSA and HPCSA are the reuslts of running the three methods by using 10,000 iterations. In Fig. 18, the curves of WCA and ACSA are redepicted for comparisong with another curve of HPCSA with the setting of 5000 iterations. The two figures have the same results once HPCSA have smaller cost than both WCA and ACSA although these costs of HPCSA found at the 5000th iteration and those of others found at the 10,000th iteration. These costs of HPCSA in Figs. 17 and 18 are less than $26,500 and $27,000 but that of WCA and ACSA is much higher than $27,000. Clearly, the perofmance of HPCSA is higher and HPCSA is about two times faster than WCA and ACSA. In addition, 50 cost values of WCA, ACSA and HPCSA are sorted in ascending order of values and plotted in Fig. 19 for the comaprsion of stability over 50 different runs. The red curve of HPCSA has thirty solutions and fourty-two solutions with less cost than pink curve of ACSA and green curve of WCA, respectively. From the 1st to the 30th solution, HPCSA has much less cost than solution of WCA. Similarly, HPCSA also obtains much smaller cost than ACSA from the 1st to the 42th solution. From the discussion on graphic results, the search performance of HPCSA is significantly more effective than WCA and ACSA in terms the effectiveness and the stability.

From results' analysis on Test Systems 2 and 3, it can be summarized as follows:

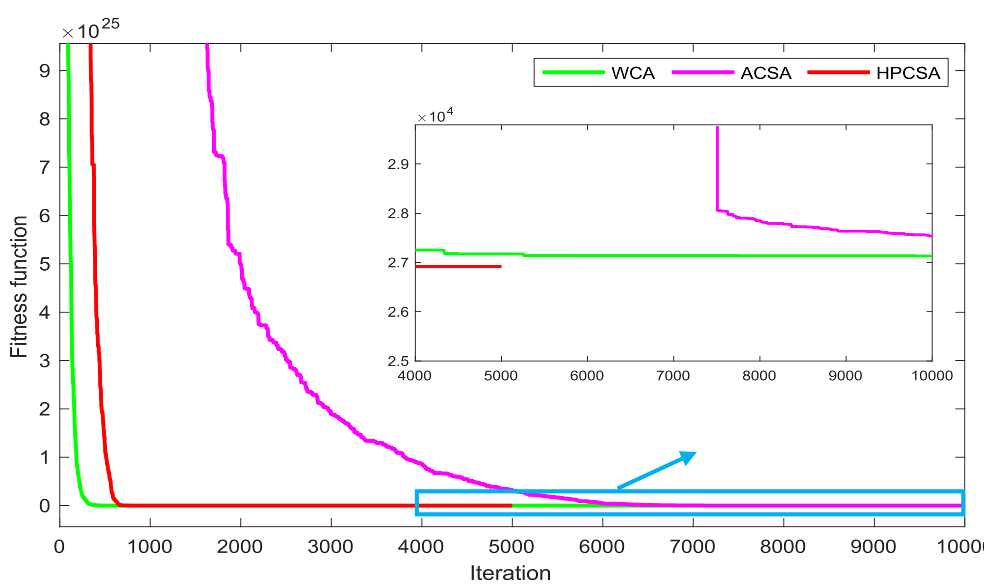

Fig. 18 Fitness vs iteration of the best run with different $Iter^{max}$ for System 3.

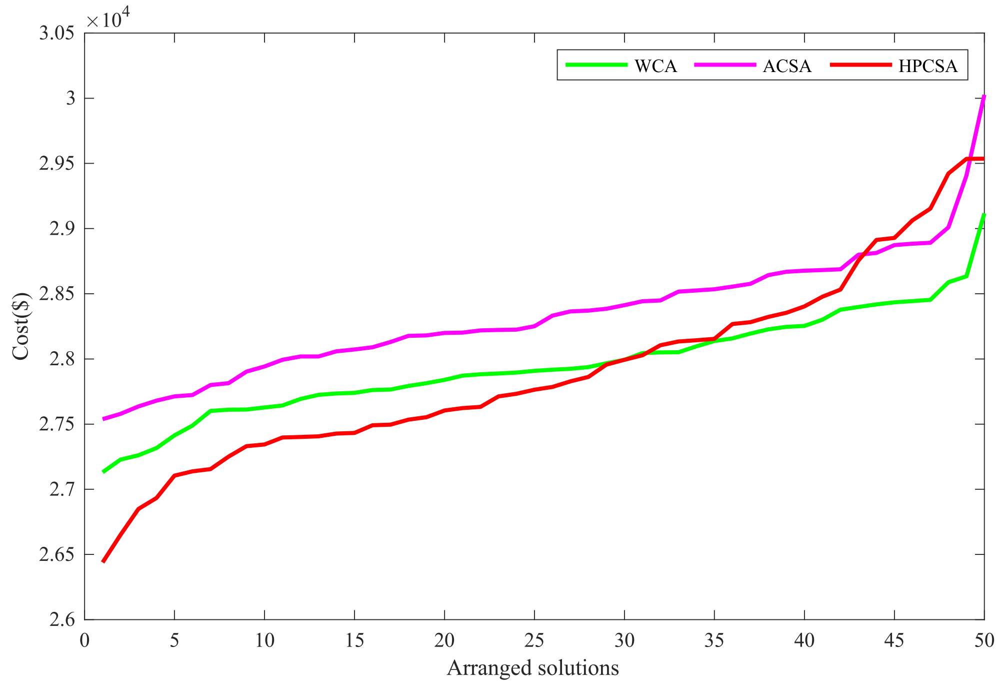

Fig. 19 The cost of 50 runs arranged from the best to worst values for System 3.

1. HPCSA can find valid solutions for all trial runs by setting the iteration number to 5000 iterations, but other methods fail to reach one valid solution for 50 trial runs.
2. HPCSA can find better solutions with much less fuel cost than other methods even for the case using only half iteration number of others.
3. HPCSA is much faster than others, about higher two times.

6.5. Comparison and discussion on system 4

In this section, WCA, CSA, SDCSA, ACSA and HPCSA are run on the most complicated system with four thermal plants, four hydroelectric plants and two wind power plants. Because this system is first developed and solved by HPCSA together with WCA, CSA, SDCSA and ACSA. $P_s = 200$ and $Iter^{max} = 10{,}000$ are selected for CSA methods but $P_s = 400$ and $Iter^{max} = 10{,}000$ are selected for WCA. On the contrary to System 2 and System 3, HPCSA cannot reach better results than all other methods for the system as setting $Iter^{max} = 5{,}000$. So, Table 6 only reports the results from HPCSA and other ones for the setting of $Iter^{max} = 10{,}000$. The minimum cost, average cost, and maximum cost of HPCSA obtained from 50 different runs are \$36201.55, \$37768.79, and \$39,449.23, respectively. And these costs are less than those of other methods excluding the comparison with ACSA for the maximum cost. The maximum cost of ACSA is \$39031.30. For showing saving cost (in \$) of HPCSA as compared to others, Fig. 20 is plotted. Then, the saving cost in \$ is converted into in % and plotted in Fig. 21. From the minimum cost bars in blue, HPCSA can reduce from \$759.49 to \$1426.96 equivalent to 2.05% of the cost of the second-best method ACSA and 3.79% of the cost of the worst method CSA. From the mean cost bars in red, the proposed HPCSA still outperforms other ones with saving cost from \$7.8 to \$1393.11, which are equivalent to 0.02% and 3.56% of the second-best method ACSA and the worst method WCA. For the maximum cost bars in grey, the proposed method is worse than ACSA by reaching higher cost of \$417.93 but its maximum cost is much less than that of other ones, namely the less costs equaling \$591.01 and \$1585.69 as compared to the third-best method CSA and the worst method

Table 6 Result comparisons for test system 4.

Method	WCA	CSA	SDCSA	ACSA	HPCSA
P_s	400	200	200	200	200
$Iter^{max}$	10,000	10,000	10,000	10,000	10,000
Minimum cost (\$)	37,405.12	37,628.51	37,198.45	36,961.04	36,201.55
Average cost (\$)	39,161.90	38,724.29	38,675.19	37,776.59	37,768.79
Maximum cost (\$)	41,034.92	40,040.24	40,465.66	39,031.30	39,449.23
Std. dev.	1,150.83	591.64	716.43	397.02	691.55
Cpu. Time (s)	1334.4	769.1	809.8	860.6	758.0
Success rate (%)	45.9	94.3	100.0	100.0	100.0

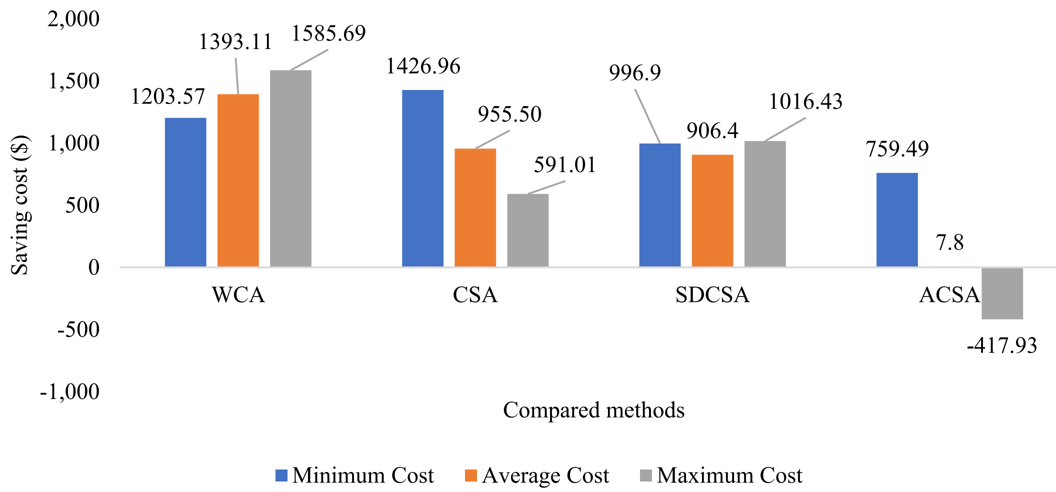

Fig. 20 Saving cost in ($) of HPCSA as compared to other ones for System 4.

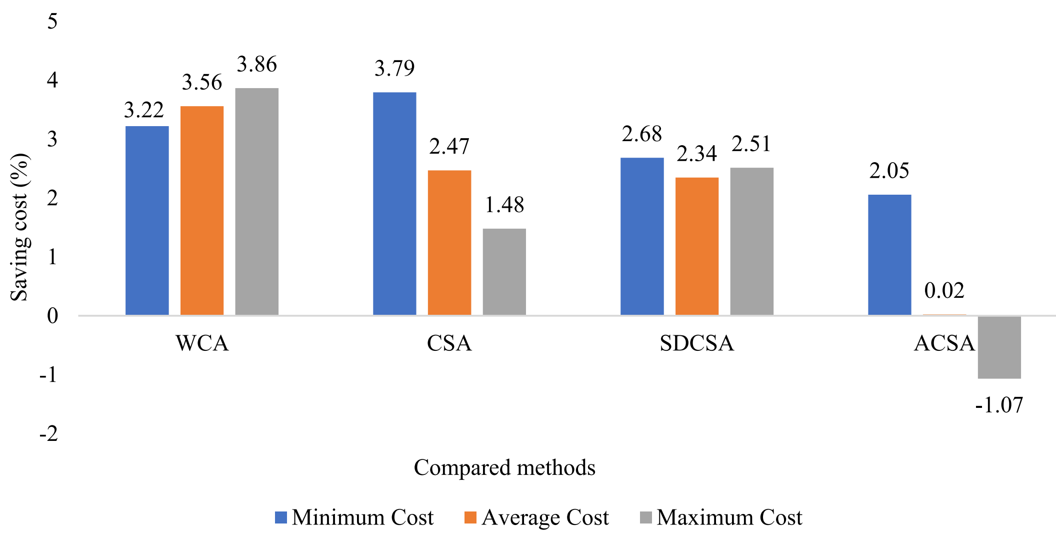

Fig. 21 Cost improvement percentage of HPCSA as compared to other ones for System 4.

WCA. The saving costs are equal to 1.48% and 3.86% of maximum cost of CSA and WCA. Although HPCSA reaches higher maximum cost than ACSA, it is still the best method among the five applied ones because HPCSA has the smallest minimum and mean costs.

For further investigation of the applied methods' performance, the fifty costs of all methods are sorted in the ascending order and plotted in Fig. 22. And the whole search of the best run among 50 runs is plotted in Fig. 18. In Fig. 22, the cost curve of HPCSA in red is below other curves excluding the pink curve of ACSA from the 34th solution to the 50th solution. In Fig. 23, the proposed method finds better solutions sooner than others and the obtained solutions of HPCSA are much better than those of all other ones for from the 7000th iteration to the final iteration. Furthermore, the obtained solution of HPCSA at the 7000th iteration is also much better than that of other ones at the final iteration. The result analysis above reveals HPCSA is greatly more robust than WCA, CSA, SDCSA and ACSA.

Optimal solutions are reported in Table A8, Table A9, Table A10 and Table A11 in the Appendix.

7. Conclusions

In this paper, the optimal Wind-Hydro-Thermal system scheduling problem was solved for reaching the most effective electric generation cost from all TPPs and successfully handling all concerned constraints by using the proposed HPCSA and four other methods such as WCA, CSA, SDCSA and ACSA. The solved problem was a big challenge for all the applied methods once it considered the valve effects of thermal power plants, reservoir volume constraints of hydroelectric plants and the uncertainty of wind feature over twenty-four one-hour periods. The applied methods were executed for reaching the most effective electric generation cost and successfully handling all concerned constraints. Four test systems arranged from the simplest to the most complicated were uti-

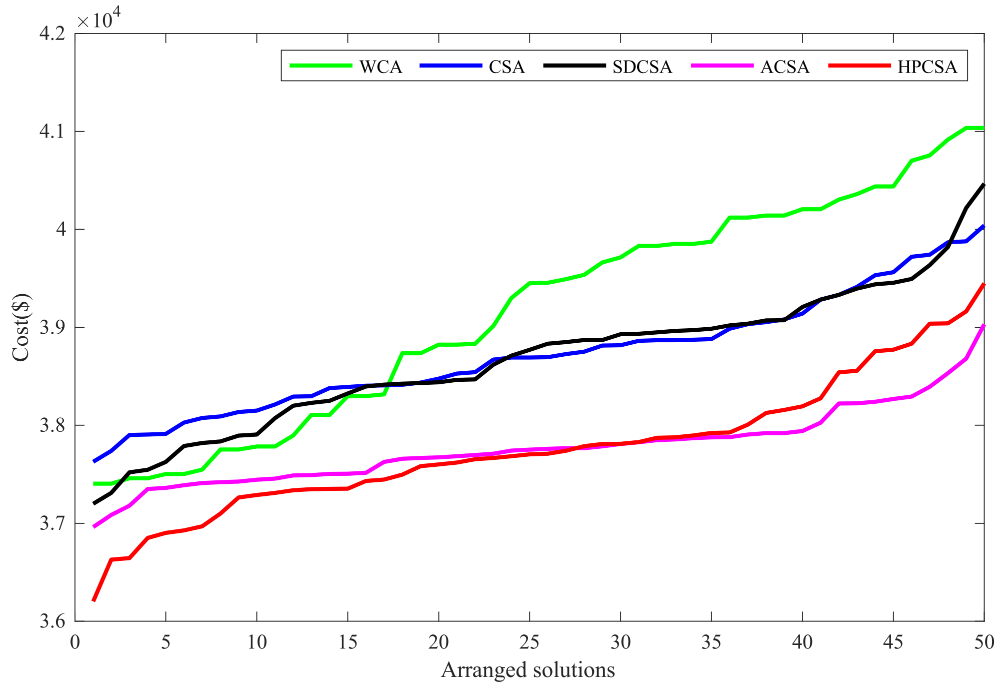

Fig. 22 The cost of 50 runs arranged from the best to worst values for System 4.

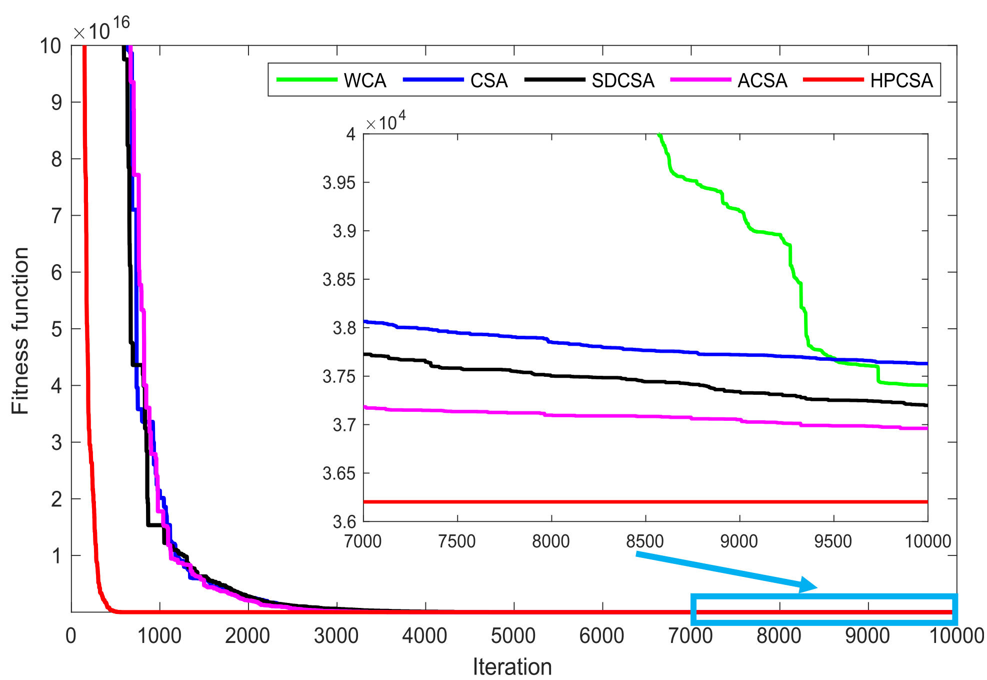

Fig. 23 The best characteristic of the executed 50 runs for System 4.

lized in which the largest system was comprised of four thermal power plants, four hydroelectric plants and two wind power plants where wind speed feature was supposed to be uncertain. For each method, fifty successful runs were obtained to collect fifty optimal cost values and find the minimum, average and maximum cost together with the standard deviation. In addition, a success rate to reach the fifty successful run and the average computation time of each run were also used as comparison criteria. For the first system with comparison to many previous methods and many modified versions of CSA, HPCSA could find either the same or less cost than others but fifty optimal solutions of HPCSA had a better mean cost and a better maximum cost. Furthermore, HPCSA was also faster than others in terms of computation time and values of population and iterations. The three values of HPCSA were respectively 0.03 s, 20 and 40 but those of others were much higher and up to 2640 s, 60 and 500, respectively. For the comparison of Test Systems 2 and 3, HPCSA was clearly superior to WCA, CSA, SDCSA and ACSA. HPCSA only used 200 and 5,000 but CSA, SDCSA and ACSA used 200 and 10,000 whilst WCA used 400 and 10,000 for population and iterations. But the best cost was found by HPCSA and the highest success rate of 100% was also from HPCSA while others had the same or much lower success rate. Namely, the success rate of WCA, CSA, MASCSA, SDCSA and ACSA was 92.6%, 71.4%, 100%, 100% and 100% for System 2, and 100%, 69.4%, 100%, 76.9% and 81.97% for System 3. For the last system with the most complicated feature and the highest number of control variables, HPCSA was still the best method reaching the lowest cost, the best mean cost, and the highest success rate of 100% although it had the same settings as CSA, SDCSA and ACSA with 200 for population and 10,000 for iterations. WCA was still run by using 400 for population and 10,000 for iterations. The success rate of WCA, CSA, SDCSA and ACSA was 45.9%, 94.3%, 100% and 100% respectively. From the analysis above, it can indicate that the proposed HPCSA is a powerful approach for the considered OWHTSS problem with a high dimension, valve effects of TPPs and the uncertainty of wind feature.

Funding

This research received no external funding.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix. See Table A1 and A2–A11.

References

[1] T.T. Nguyen, D.N. Vo, An efficient cuckoo bird inspired meta-heuristic algorithm for short-term combined economic emission hydrothermal scheduling, Ain Shams Eng. J. 9 (4) (2018) 483–497.

Table A1 Data of thermal units for Test System 1.

i	a_{si}	b_{si}	c_{si}	α_{si}	β_{si}	$P_{si,min}$(MW)	$P_{si,max}$(MW)
1	575	9.2	0.00184	0	0	150	1500

Table A2 Generation function and limits of the hydroelectric plant of System 1.

j	a_{hj}	b_{hj}	c_{hj}	$P_{hj,\min}$ (MW)	$P_{hj,\max}$ (MW)	$V_{hj,0}$ (Acre-ft)	$V_{hj,M}$ (Acre-ft)	$V_{hj,\min}$ (Acre-ft)	$V_{hj,\max}$ (Acre-ft)	$q_{hj,\min}$ (Acre-ft)	$q_{hj,\max}$ (Acre-ft)
1	330	4.97	0	0	1000	10,0000	60,000	60,000	120,000	330	5300

Table A3 Load and inflow of Test System 1.

m	$P_{Load,m}$(MW)	$I_{h1,m}$(acre-ft/h)
1	1200	2000
2	1500	2000
3	1100	2000
4	1800	2000
5	950	2000
6	1300	2000

Table A4 Data of thermal units of System 2 and System 3.

i	$P_{si,min}$(MW)	$P_{si,max}$(MW)	a_{si}	b_{si}	c_{si}	α_{si}	β_{si}
1	10	500	60	1.8	0.0011	14	0.04
2	10	675	100	2.1	0.0012	16	0.038
3	10	550	120	1.7	0.0013	18	0.037
4	10	500	40	1.5	0.0014	20	0.035

Table A5 The generation function and limits of hydroelectric plants of System 2, 3 and 4.

j	$P_{hj,\min}$ (MW)	$P_{hj,\max}$ (MW)	a_{hj}	b_{hj}	c_{hj}	$V_{hj,\min}$ (Acre-ft)	$V_{hj,\max}$ (Acre-ft)	$V_{h1,0}$ (Acre-ft)	$V_{h1,M}$ (Acre-ft)	$q_{hj,\min}$ (Acre-ft)	$q_{hj,\max}$ (Acre-ft)
1	0	1000	330	4.970	0.0001	60,000	120,000	100,000	80,000	330	5400
2	0	1000	350	5.20	0.0001	60,000	120,000	100,000	90,000	350	5650
3	0	1000	280	5.0	0.00011	60,000	120,000	100,000	85,000	280	5390
4	0	1000	300	4.80	0.00011	60,000	120,000	100,000	85,000	300	5210

Table A6 Load demand and inflows of Systems 2, 3 and 4, and wind data of System 3.

m	$P_{Load,m}$ (MW)	$I_{h1,m}$ (acre-ft/h)	$I_{h2,m}$ (acre-ft/h)	$I_{h3,m}$ (acre-ft/h)	$I_{h4,m}$ (acre-ft/h)	$V_{w1,m}$ *(m/s)*	$V_{w2,m}$ *(m/s)*
1	1200	1000.000	800.000	800.000	600.000	13.25	11.8
2	1500	600.000	500.000	600.000	600.000	14	12
3	1100	700.000	500.000	700.000	700.000	12.75	12.2
4	1800	900.000	700.000	900.000	900.000	11.9	12.4
5	1200	900.000	700.000	900.000	900.000	12.5	12.5
6	1300	800.000	1000.000	800.000	800.000	13.9	14
7	1200	800.000	800.000	800.000	800.000	11.8	15
8	1500	700.000	800.000	700.000	700.000	12.7	14.5
9	1100	500.000	800.000	500.000	500.000	12.9	13
10	1800	500.000	800.000	500.000	500.000	12.2	13.75
11	1200	500.000	1000.000	500.000	500.000	15	13.4
12	1300	500.000	500.000	500.000	500.000	13.25	13.4
13	1200	800.000	500.000	700.000	800.000	14.3	12.8
14	1500	900.000	600.000	500.000	900.000	14.1	12.25
15	1100	600.000	600.000	600.000	600.000	14.25	11.4
16	1800	500.000	500.000	500.000	900.000	11.75	11.5
17	1200	950.000	950.000	950.000	900.000	13.75	11
18	1300	650.000	650.000	650.000	900.000	12.6	11.25
19	1200	550.000	550.000	550.000	700.000	11.5	11.1
20	1500	600.000	800.000	600.000	600.000	11.9	11
21	1100	600.000	800.000	600.000	600.000	14.5	11.45
22	1800	350.000	800.000	350.000	700.000	16	11.8
23	1200	600.000	1000.000	600.000	600.000	12.7	11.75
24	1300	400.000	400.000	800.000	800.000	13	12.25

Table A7 Wind data of System 4.

wf	$P_{wwf,rate}$(MW)	c_{wwf}	k_{wwf}	e_{wf} ($/MWh)	g_{wf} ($/MWh)	h_{wf} ($/MWh)	$V_{wwf,ci}$(m/s)	$V_{wwf,rate}$(m/s)	$V_{wwf,co}$(m/s)
1	75	9	2	1.6	1.5	3.0	3	16	25
2	60	10	2	1.75	1.5	3.0	3	16	25

Table A8 Optimal control variables found by HPCSA for System 1.

m	$V_{h1,m}$ (acre-ft)
1	101927.5689
2	85963.1031
3	93854.7533
4	60000.0000
5	70435.9926
6	60,000

Table A9 Optimal control variables found by HPCSA for System 2.

m	$V_{h1,m}$ (acre-ft)	$V_{h2,m}$ (acre-ft)	$V_{h3,m}$ (acre-ft)	$V_{h4,m}$ (acre-ft)	$P_{s2,m}$ (MW)	$P_{s3,m}$ (MW)	$P_{s4,m}$ (MW)
1	99735.0786	100449.99	100519.95	100299.16	258.0216	10.0749	369.0397
2	96713.9475	100298.48	100839.93	100489.4	10.0000	10.0000	500.0000
3	97071.5892	96827.062	101258.34	100806.29	10.0000	94.9488	120.5108
4	96836.4317	92641.402	101874.7	99406.056	10.0000	179.8184	10.0000
5	97270.327	90973.588	102491.39	98439.32	10.0000	179.8158	105.8780
6	97663.6205	90257.648	102906.42	96759.337	10.0000	96.3282	279.2792
7	98068.4376	86996.878	102957.85	96337.433	10.0000	10.9846	10.0000
8	98350.2031	87400.574	100478.09	94577.366	10.0033	10.0000	189.5256
9	98200.9405	87478.756	100693.17	93092.596	10.0008	349.6186	10.0000
10	96080.5991	87538.122	99472.917	92732.272	175.3443	434.5071	10.2067
11	95933.4019	88187.876	98519.729	92425.213	92.6741	179.8157	279.2813
12	95161.3554	88333.25	96533.476	91537.616	10.0000	179.8157	12.3795
13	95583.6922	88474.752	96496.852	87737.865	10.0000	10.0059	189.5184
14	91597.7423	88715.891	96435.62	87905.258	10.0000	264.7213	10.0000
15	91806.3774	88962.76	96620.278	86971.43	10.0000	179.8154	369.0162
16	90826.8306	89111.574	95864.711	85992.639	92.6737	264.7217	369.0340
17	91072.46	89359.922	96247.782	86589.395	92.6637	550.0000	189.5210
18	90359.2258	88876.778	95587.991	87109.168	10.0000	264.7037	279.2792
19	89369.8584	88054.904	94813.665	87441.369	10.0001	94.9075	189.5193
20	85999.0607	88439.013	93247.012	87672.125	10.0000	10.0000	189.5184
21	86175.3547	88888.525	91039.928	87851.939	10.0000	10.0000	369.0363
22	81553.0297	89318.414	87764.588	88147.519	10.0001	10.0000	10.6657
23	80653.6497	89950.05	85963.404	88037.371	10.0000	179.8156	99.7588
24	60000.0001	120,000	60,000	119900.65	10.0000	10.0000	99.7594

Table A10 Optimal control variables found by HPCSA for System 3.

m	$V_{h1,m}$ (acre-ft)	$V_{h2,m}$ (acre-ft)	$V_{h3,m}$ (acre-ft)	$V_{h4,m}$ (acre-ft)	$P_{s2,m}$ (MW)	$P_{s3,m}$ (MW)	$P_{s4,m}$ (MW)	$P_{w1,m}$ (MW)	$P_{w2,m}$ (MW)
1	98583.1927	100443.6145	100175.4074	98158.7428	10.0000	10.0000	10.0000	99	54.4
2	98405.4221	100465.8123	99047.1274	94968.3235	10.0000	94.9079	24.7805	108	56
3	97781.6541	100567.0146	98633.0415	93156.1309	10.0000	10.0000	10.0067	93	57.6
4	94951.5839	100813.6506	99027.3652	91956.9025	10.0000	179.8141	189.5196	82.8	59.2
5	92756.3648	100928.6709	99644.5401	92222.3763	10.0000	17.8147	189.5203	90	60
6	91664.2507	99773.7491	100163.7416	92653.6351	92.6694	179.8153	10.0000	106.8	72
7	91154.5437	99902.5256	100683.7385	92071.7927	10.0000	10.0000	369.0392	81.6	80
8	90579.7357	97712.8920	101103.4112	92471.7899	92.6736	10.0000	369.0392	93	76
9	90150.7814	97081.3519	101322.3452	92670.6490	10.0000	347.7433	10.0001	94.8	64
10	86874.8166	96595.4640	101527.6198	92844.8920	10.0000	94.9092	500.0000	86.4	70
11	84647.2925	96654.2302	101744.4100	92832.0287	10.0000	10.0000	189.5198	120	67.2
12	84715.7654	95362.2575	101964.3846	91132.2896	10.0003	179.8156	10.0002	99	67.2
13	84441.5367	95504.3057	101800.7883	89342.5996	10.0000	10.0000	99.7598	111.6	62.4
14	84457.2880	93860.6785	99621.5695	89416.2362	10.0000	10.0000	10.0000	109.2	58
15	84188.7047	94074.1849	98550.1590	89296.8550	92.2565	10.0000	189.5194	111	51.2
16	84110.4452	91152.3443	95425.1933	89046.6344	10.0004	10.0000	10.0000	81	52
17	84710.9124	91381.2027	93428.5204	88832.4973	10.0000	10.0000	10.0000	105	48
18	83219.9489	91116.7336	93681.4926	88402.2666	10.0001	264.6652	10.0037	91.2	50
19	83138.8991	90431.2253	93274.4675	88787.8967	10.0000	94.9076	354.4397	78	48.8
20	81031.2979	88860.1369	92200.3571	88918.2908	10.0000	10.0012	10.0001	82.8	48
21	81070.0188	89136.2841	92357.9131	88188.5190	10.0001	10.0000	499.9999	114	51.6
22	80939.6886	89522.9293	88937.1772	86068.4279	10.0000	10.0000	189.5194	120	54.4
23	79945.5482	89971.6622	89209.9373	84947.7732	91.8236	10.0000	189.5196	92.4	54
24	80000.0000	90000.0000	85000.0000	85000.0000	10.0000	10.0000	10.0000	96	58

Table A11 Optimal control variables found by HPCSA for System 4.

m	$V_{h1,m}$ (acre-ft)	$V_{h2,m}$ (acre-ft)	$V_{h3,m}$ (acre-ft)	$V_{h4,m}$ (acre-ft)	$P_{s2,m}$ (MW)	$P_{s3,m}$ (MW)	$P_{s4,m}$ (MW)	$P_{w1,m}$ (MW)	$P_{w2,m}$ (MW)
1	98988.66	99635.69	100060.5	98531.27	10	10	99.76303	20.0895	22.3205
2	98887.29	95618.45	100380.2	96922.86	10	10	10	21.3095	23.4965
3	98411.42	93353.1	100269.1	97243.74	10	10	189.5196	21.3095	28.1475
4	96737.47	91150.34	99312.83	97661.06	92.67349	179.8158	99.87106	24.8455	29.2985
5	96879.1	90897.98	97530.5	97949.01	10	94.90791	189.5196	0.001	75
6	95586.57	90733.53	95313.94	98388.07	10	10	99.79311	0.001	29.8045
7	95799.38	90870.28	95547.6	98573.25	10	550	189.5196	22.4265	26.4235
8	95,420	90893.06	92955.94	97328.78	10	94.98054	10	23.8595	27.2245
9	95520.83	91275.98	92961.74	94,452	10	10	189.5196	21.9415	0.001
10	94993.21	91562.36	88761.25	93037.09	10	179.8158	99.75985	21.1755	27.8645
11	94833.45	92039.13	86995.67	93131.34	10	179.8158	279.2794	22.4005	26.1465
12	93709.54	91884.19	86207.62	91939.07	10	264.7196	10	19.4955	22.2555
13	92737.82	90928.61	86586.68	92,244	91.92801	179.8158	189.5196	0.001	22.7085
14	90962.27	89777.39	86685.03	92129.54	10	179.8158	99.81829	60	75
15	90485.59	89529.33	86976.07	92297.61	10	264.7237	279.2794	21.7265	0.001
16	89844.34	88557.17	87179.48	90420.96	10	264.7237	369.0388	19.6835	0.001
17	90286.41	88684.37	87817.16	89870.77	92.67349	349.6284	189.5196	0.001	29.7735
18	90238.56	87931.89	88184.34	90448.68	92.67349	179.8158	369.0392	25.3995	28.2295
19	90456.7	87956.41	88452.38	90136.23	175.3108	179.8158	369.018	22.5775	25.2405
20	90630.17	88401.86	88713.65	88322.97	92.67349	264.7237	369.0392	60	0.001
21	89090.51	88850.02	89016.39	88231.61	10	264.7237	189.5196	22.0035	0.001
22	87059.31	89300.01	84764.23	87630.04	10	10	99.76358	21.6905	25.8575
23	83477.84	89950.01	84692.57	86869.88	10	10	10.00004	20.2785	0.001
24	120,000	60,000	60,000	120,000	10	10	10.00036	10.7815	14.5985

[2] T.T. Nguyen, N. Vu Quynh, M.Q. Duong, L. Van Dai, Modified differential evolution algorithm: A novel approach to optimize the operation of hydrothermal power systems while considering the different constraints and valve point loading effects, Energies 11 (3) (2018) 540.

[3] S. Ghosh, M. Kaur, S. Bhullar, V. Karar, Hybrid ABC-BAT for solving short-term hydrothermal scheduling problems, Energies 12 (3) (2019) 551.

[4] M. Basu, A simulated annealing-based goal-attainment method for economic emission load dispatch of fixed head hydrothermal power systems, Int. J. Electr. Power Energy Syst. 27 (2) (2005) 147–153.

[5] A.K. Sharma, Short term hydrothermal scheduling using evolutionary programming, in Thesis submitted in partial fulfillment of the requirements for the award of degree of Master of Engineering in Power Systems & Electric Drives, Patiala, Thapar University, 2009.

[6] I.A. Farhat, M.E. El-Hawary, Optimization methods applied for solving the short-term hydrothermal coordination problem, Electr. Power Syst. Res. 79 (9) (2009) 1308–1320.

[7] M. Basu, Artificial immune system for fixed head hydrothermal power system, Energy 36 (2011) 606–612.

[8] I.A. Farhat, M.E. El-Hawary, Fixed-head hydro-thermal ascheduling using a modified bacterial foraging algorithm, in: 2010 IEEE Electrical Power & Energy Conference IEEE, August 2010, p. 1-6.

[9] J. Sasikala, M. Ramaswamy, Optimal gamma based fixed head hydrothermal scheduling using genetic algorithm, Expert Syst. Appl. 37 (2010) 3352–3357.

[10] M. Murali, M.S. Kumari, M. Sydulu, Short-range fixed head hydrothermal scheduling using fast genetic algorithm, in: 2012 7th IEEE Conference on Industrial Electronics and Applications (ICIEA), July 2012, p. 1313–1318.

[11] N. Narang, J.S. Dhillon, D.P. Kothari, Scheduling short-term hydrothermal generation using predator prey optimization technique, Appl. Soft Comput. 21 (2014) 298–308.

[12] V.N. Dieu, W. Ongsakul, Enhanced merit order and augmented Lagrange Hopfield network for hydrothermal scheduling, Int. J. Electr. Power Energy Syst. 30 (2008) 93–101.

[13] V.N. Dieu, W. Ongsakul, Improved merit order and augmented Lagrange Hopfield network for short term hydrothermal scheduling, Energy Convers. Manage. 50 (2009) 3015–3023.

[14] A. Wood, B. Wollenberg, G.B. Sheblé, Power Generation, Operation and Control, John Wiley & Sons, 2013.

[15] K.P. Wong, Y.W. Wong, Short-term hydrothermal scheduling part. I. Simulated annealing approach, IEE Proceedings-Generation Transmission and Distribution 141 (1994) 497–501.

[16] P.C. Yang, H.T. Yang, C.L. Huang, Scheduling short-term hydrothermal generation using evolutionary programming techniques, IEE Proc.-Generat. Transmission Distribut. 143 (1996) 371–376.

[17] P.K. Hota, R. Chakrabarti, P.K. Chattopadhyay, Short-term hydrothermal scheduling through evolutionary programming technique, Electr. Power Syst. Res. 52 (1999) 189–196.

[18] N. Sinha, R. Chakrabarti, P.K. Chattopadhyay, Fast evolutionary programming techniques for short-term hydrothermal scheduling, IEEE Trans. Power Syst. 18 (2003) 214–220.

[19] N. Sinha, R. Chakrabarti, P.K. Chattopadhyay, Fast evolutionary programming techniques for short-term hydrothermal scheduling, Electric Power Syst. Res. 66 (2003) 97–103.

[20] C. Nallasivan, D.S. Suman, J. Henry, S. Ravichandran, A novel approach for short-term hydrothermal scheduling using hybrid technique, in: 2006 IEEE Power India Conference, IEEE, April 2006, pp. 5.

[21] B. Türkay, F. Mecitoğlu, S. Baran, Application of a fast evolutionary algorithm to short-term hydro-thermal generation scheduling, Energy Sources Part B 6 (2011) 395–405.
[22] C. Samudi, G.P. Das, P.C. Ojha, T.S. Sreeni, S. Cherian, Hydro thermal scheduling using particle swarm optimization, in: 2008 IEEE/PES Transmission and Distribution Conference and Exposition, IEEE, April 2008, pp. 1–5.
[23] S. Thakur, C. Boonchay, W. Ongsakul, Optimal hydrothermal generation scheduling using self-organizing hierarchical PSO, in: IEEE PES General Meeting, IEEE, July 2010, pp. 1–6.
[24] S. Padmini, C.C.A. Rajan, Improved PSO for short term hydrothermal scheduling, IEEE Conference in India (2011) 332–334.
[25] S. Padmini, C.C.A. Rajan, P. Murthy, Application of improved PSO technique for short term hydrothermal generation scheduling of power system, in: International Conference on Swarm, Evolutionary, and Memetic Computing, Springer, Berlin, Heidelberg. December 2011, pp. 176–182.
[26] M.S. Fakhar, S.A.R. Kashif, M.A. Saqib, T. ul Hassan, Non cascaded short-term hydro-thermal scheduling using fully-informed particle swarm optimization, Int. J. Electr. Power Energy Syst. 73 (2015) 983–990.
[27] R.K. Swain, A.K. Barisal, P.K. Hota, R. Chakrabarti, Short-term hydrothermal scheduling using clonal selection algorithm, Int. J. Electr. Power Energy Syst. 33 (2011) 647–656.
[28] T.T. Nguyen, D.N. Vo, W. Ongsakul, One rank cuckoo search algorithm for short-term hydrothermal scheduling with reservoir constraint, in: 2015 IEEE Eindhoven PowerTech, IEEE, June 2015, pp. 1-6.
[29] T.T. Nguyen, D.N. Vo, B.H. Dinh, Cuckoo search algorithm using different distributions for short-term hydrothermal scheduling with reservoir volume constraint, Int. J. Electr. Eng. Informat. 8 (2016) 76.
[30] B.H. Dinh, T.T. Nguyen, D.N. Vo, Adaptive cuckoo search algorithm for short-term fixed-head hydrothermal scheduling problem with reservoir volume constraints, Int. J. Grid Distribut. Comput. 9 (2016) 191–204.
[31] T.T. Nguyen, D.N. Vo, T. Deveikis, A. Rozanskiene, Improved cuckoo search algorithm for nonconvex hydrothermal scheduling with volume constraint, Elektronika ir Elektrotechnika 23 (2017) 68–73.
[32] T.T. Nguyen, D.N. Vo, B.H. Dinh, An effectively adaptive selective cuckoo search algorithm for solving three complicated short-term hydrothermal scheduling problems, Energy 155 (2018) 930–956.
[33] M.S. Fakhar, S.A.R. Kashif, N.U. Ain, H.Z. Hussain, A. Rasool, I.A. Sajjad, Statistical performances evaluation of APSO and improved APSO for short term hydrothermal scheduling problem, Appl. Sci. 9 (12) (2019) 2440.
[34] M. Nazari-Heris, A.F. Babaei, B. Mohammadi-Ivatloo, S. Asadi, Improved harmony search algorithm for the solution of non-linear non-convex short-term hydrothermal scheduling, Energy 151 (2018) 226–237.
[35] H. Yin, F. Wu, X. Meng, Y. Lin, J. Fan, A. Meng, Crisscross optimization based short-term hydrothermal generation scheduling with cascaded reservoirs, Energy 203 (2020) 117822.
[36] G. Chen, M. Gao, Z. Zhang, S. Li, Hybridization of chaotic grey wolf optimizer and dragonfly algorithm for short-term hydrothermal scheduling, IEEE Access 8 (2020) 142996–143020.
[37] A. Mohammadi, M.H. Ahmadi, M. Bidi, F. Joda, A. Valero, S. Uson, Exergy analysis of a combined cooling, heating and power system integrated with wind turbine and compressed air energy storage system, Energy Convers. Manage. 131 (2017) 69–78.
[38] M.H. Mohammadnezami, M.A. Ehyaei, M.A. Rosen, M.H. Ahmadi, Meeting the electrical energy needs of a residential building with a wind-photovoltaic hybrid system, Sustainability 7 (3) (2015) 2554–2569.
[39] E.E. Elattar, Optimal power flow of a power system incorporating stochastic wind power based on modified moth swarm algorithm, IEEE Access 7 (2019) 89581–89593.
[40] H.T. Jadhav, R. Roy, Gbest guided artificial bee colony algorithm for environmental/economic dispatch considering wind power, Expert Syst. Appl. 40 (2013) 6385–6399.
[41] H. Rezk, A. Fathy, A.A.Z. Diab, M. Al-Dhaifallah, The application of water cycle optimization algorithm for optimal placement of wind turbines in wind farms, Energies 12 (22) (2019) 4335.
[42] X.S. Yang, S. Deb, Cuckoo search via Lévy flights, in: 2009 World congress on nature & biologically inspired computing (NaBIC) IEEE, December 2009, pp. 210–214.
[43] H. Rakhshani, A. Rahati, Snap-drift cuckoo search: a novel cuckoo search optimization algorithm, Appl. Soft Comput. 52 (2017) 771–794.
[44] S. Kumar, R. Naresh, Efficient real coded genetic algorithm to solve the non-convex hydrothermal scheduling problem, Int. J. Electr. Power Energy Syst. 29 (10) (2007) 738–747.
[45] M. Basu, Economic environmental dispatch of hydrothermal power system, Int. J. Electr. Power Energy Syst. 32 (6) (2010) 711–720.
[46] S. Mirjalili, A.H. Gandomi, S.Z. Mirjalili, S. Saremi, H. Faris, S. M. Mirjalili, Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems, Adv. Eng. Softw. 114 (2017) 163–191.
[47] G.F. Gomes, S.S. da Cunha, A.C. Ancelotti, A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates, Eng. Comput. 35 (2) (2019) 619–626.
[48] Q. Askari, M. Saeed, I. Younas, Heap-based optimizer inspired by corporate rank hierarchy for global optimization, Expert Syst. Appl. 161 (2020) 113702.
[49] M. Khishe, M.R. Mosavi, Chimp optimization algorithm, Expert Syst. Appl. 149 (2020) 113338.
[50] J.S. Chou, D.N. Truong, A novel metaheuristic optimizer inspired by behavior of jellyfish in ocean, Appl. Math. Comput. 389 (2021) 125535.
[51] A. Faramarzi, M. Heidarinejad, B. Stephens, S. Mirjalili, Equilibrium optimizer: a novel optimization algorithm, Knowl.-Based Syst. 191 (2020) 105190.
[52] F. Zaman, S.M. Elsayed, T. Ray, R.A. Sarker, Configuring two-algorithm-based evolutionary approach for solving dynamic economic dispatch problems, Eng. Appl. Artif. Intell. 53 (2016) 105–125.
[53] Z. Wang, C. Wang, J. Wu, Wind energy potential assessment and forecasting research based on the data pre-processing technique and swarm intelligent optimization algorithms, Sustainability 8 (2016) 1191.
[54] T.T. Nguyen, D.N. Vo, The application of an effective cuckoo search algorithm for optimal scheduling of hydrothermal system considering transmission constraints, Neural Comput. Appl. 31 (2019) 4231–4252.
[55] A. Agasthian, R. Pamula, L.A. Kumaraswamidhas, Fault classification and detection in wind turbine using Cuckoo-optimized support vector machine, Neural Comput. Appl. 31 (5) (2019) 1503–1511.
[56] I. Ullah, S. Hussain, Time-constrained nature-inspired optimization algorithms for an efficient energy management system in smart homes and buildings, Appl. Sci. 9 (2019) 792.
[57] T.T. Nguyen, T.T. Nguyen, D.N. Vo, An effective cuckoo search algorithm for large-scale combined heat and power economic dispatch problem, Neural Comput. Appl. 30 (2018) 3545–3564.
[58] B. Yang, J. Miao, Z. Fan, J. Long, X. Liu, Modified cuckoo search algorithm for the optimal placement of actuators problem, Appl. Soft Comput. 67 (2018) 48–60.

[59] Z. Sun, X. Gu, Hybrid algorithm based on an estimation of distribution algorithm and cuckoo search for the no idle permutation flow shop scheduling problem with the total tardiness criterion minimization, Sustainability 9 (2017) 953.

[60] Y. Zhang, H. Zhao, Y. Cao, Q. Liu, Z. Shen, J. Wang, M. Hu, A hybrid ant colony and cuckoo search algorithm for route optimization of heating engineering, Energies 11 (2018) 2675.

[61] T. Kang, J. Yao, M. Jin, S. Yang, T. Duong, A novel improved cuckoo search algorithm for parameter estimation of photovoltaic (PV) models, Energies 11 (2018) 1060.

[62] L. Xiao, M. Dridi, A. Hajjam El Hassani, H. Fei, W. Lin, An improved cuckoo search for a patient transportation problem with consideration of reducing transport emissions, Sustainability 10 (2018) 793.

[63] X. Meng, J. Chang, X. Wang, Y. Wang, Multi-objective hydropower station operation using an improved cuckoo search algorithm, Energy 168 (2019) 425–439.

[64] X. Zhang, Q. Wan, Y. Fan, Applying modified cuckoo search algorithm for solving systems of nonlinear equations, Neural Comput. Appl. 31 (2019) 553–576.

[65] X. Yuan, H. Tian, Y. Yuan, Y. Huang, R.M. Ikram, An extended NSGA-III for solution multi-objective hydro-thermal-wind scheduling considering wind power cost, Energy Convers. Manage. 96 (2015) 568–578.

[66] J. Zhou, P. Lu, Y. Li, C. Wang, L. Yuan, L. Mo, Short-term hydro-thermal-wind complementary scheduling considering uncertainty of wind power using an enhanced multi-objective bee colony optimization algorithm, Energy Convers. Manage. 123 (2016) 116–129.

[67] R.A. Moraes, T.S. Fernandes, A.G. Arantes, C. Unsihuay-Vila, Short-term scheduling of integrated power and spinning reserve of a wind-hydrothermal generation system with ac network security constraints, J. Control Automat. Electri. Syst. 29 (2018) 1–14.

[68] S.K. Damodaran, T.K. Sunil Kumar, Hydro-thermal-wind generation scheduling considering economic and environmental factors using heuristic algorithms, Energies 11 (2018) 353.

[69] B.P. Cotia, C.L. Borges, A.L. Diniz, Optimization of wind power generation to minimize operation costs in the daily scheduling of hydrothermal systems, Int. J. Electr. Power Energy Syst. 113 (2019) 539–548.

[70] K. Dasgupta, P.K. Roy, V. Mukherjee, Power flow based hydro-thermal-wind scheduling of hybrid power system using sine cosine algorithm, Electr. Power Syst. Res. 178 (2020) 106018.

[71] Department of Energy (DoE) (2010) EIA-0484 international energy outlook 2010. U.S. Department of Energy, Washington.

[72] J. Hetzer, C.Y. David, K. Bhattarai, An economic dispatch model incorporating wind power, IEEE Trans. Energy Convers. 23 (2) (2008) 603–611.

[73] F. Yao, Z.Y. Dong, K. Meng, Z. Xu, H.H.C. Iu, K.P. Wong, Quantum-inspired particle swarm optimization for power system operations considering wind power uncertainty and carbon tax in Australia, IEEE Trans. Ind. Inf. 8 (4) (2012) 880–888.

[74] A. Panda, M. Tripathy, Security constrained optimal power flow solution of wind-thermal generation system using modified bacteria foraging algorithm, Energy 93 (2015) 816–827.

[75] X. Liu, W. Xu, Minimum emission dispatch constrained by stochastic wind power availability and cost, IEEE Trans. Power Syst. 25 (3) (2010) 1705–1713.

[76] D.P. Kothari, I.J, Dhillon, Power system optimization, second edition, New Delhi, 2011.

[77] F. Yao, Z.Y. Dong, K. Meng, Z. Xu, H.H.C. Iu, K.P. Wong, Quantum-inspired particle swarm optimization for power system operations considering wind power uncertainty and carbon tax in Australia, IEEE Trans. Ind. Inf. 8 (2012) 880–888.

[78] X. Liu, Economic load dispatch constrained by wind power availability: a wait-and-see approach, IEEE Trans. Smart Grid 1 (2010) 347–355.

[79] P.P. Biswas, P.N. Suganthan, G.A. Amaratunga, Optimal power flow solutions incorporating stochastic wind and solar power, Energy Convers. Manage. 148 (2017) 1194–1207.

[80] T.T. Nguyen, L.H. Pham, F. Mohammadi, L.C. Kien, Optimal scheduling of large-scale wind-hydro-thermal systems with fixed-head short-term model, Appl. Sci. 10 (8) (2020) 2964.

[81] H. Zhang, D. Yue, X. Xie, C. Dou, F. Sun, Gradient decent based multi-objective cultural differential evolution for short-term hydrothermal optimal scheduling of economic emission with integrating wind power and photovoltaic power, Energy 122 (2017) 748–766.

[82] T.T. Nguyen, A high performance social spider optimization algorithm for optimal power flow solution with single objective optimization, Energy 171 (2019) 218–240.