An Innovative Hybrid Algorithm for Solving Combined Economic and Emission Dispatch Problems

Pooja Verma (poojatwri28@gmail.com)
Indira Gandhi National Tribal University
Raghav Prasad Parouha
Indira Gandhi National Tribal University

Research Article

Keywords: Economic load dispatch, Combined economic emission dispatch problem, Meta-heuristic algorithms, Hybrid algorithm

Posted Date: February 23rd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1376881/v1

An innovative hybrid algorithm for solving combined economic and emission dispatch problems

*Pooja Verma, Department of Mathematics, Indira Gandhi National Tribal University, Amarkantak, M.P., India; Email: poojatwri28@gmail.com
Raghav Prasad Parouha, Department of Mathematics, Indira Gandhi National Tribal University, Amarkantak, M.P., India; Email: rparouha@gmail.com
*Corresponding author

Abstract: As environmental concerns have grown, the combined economic emission dispatch (CEED) problem has gotten a lot of attention. Both the cost of fuel and the emission pollution caused by it must be kept to a minimum. As a result, this paper presents an innovative hybrid approach (i*h*PSODE) for solving CEED problems. This hybrid technique incorporated novel differential evolution (nDE) and particle swarm optimization (nPSO). Where nDE introduces a new mutation approach and crossover rate (to prevent premature convergence) as well as nPSO introduces a new acceleration coefficient, inertia weight and position improve equation (to alleviate the stagnation). So as to balance among local and global search ability, after i*h*PSODE population evaluation, the best half individuals are determined and the rest individuals are discarded. Then, nPSO is used in the current population (to sustain exploration and exploitation) and nDE is employed in the nPSO generated population (to improve convergence accuracy). The competence of the proposed algorithms (i*h*PSODE, nPSO and nDE) are inspected on 23 unconstrained benchmark function and then solved 3 test system (3-, 6- and 40-unit) of economic load dispatch (ELD) and 3 test system (3-, 10- and 40-unit) of CEED problem. The experiments have denoted that the proposed algorithms show competitive results and significant performances.

Keywords: Economic load dispatch, Combined economic emission dispatch problem, Meta-heuristic algorithms, Hybrid algorithm.

1. Introduction

In modern power system operation, economic load dispatch (ELD) is a critical optimization problem (Mansor et al. 2018). The basic goal of the ELD problem is to lower total generation costs while keeping load demand and other equality and inequality limitations in mind. Apart from the producing capacity limits, the classic ELD primarily analyses the power balance constraint. But, due to practical limitations in power system operation, ELD essentially consider a multiplicity of real-world constraints like transmission loss, prohibited operating zones, multi-fuel options, ramp rate limits, spinning reserve along with system power demand etc. It resulted with a non-convex nonlinear ELD problem and finding an optimum solution of this type problem is very challenging and time-consuming. The environmental constraint, which consists of Carbon oxides (Cox), Nitrogen oxides (NOx) and Sulphur oxides (Sox), infects the air, is one of those limitations that is always taken into account. By properly allocating load among available generators, the hazardous environmental pollutants released by fossil-fuel power plants can be decreased. However, the power plant's operational costs would rise. As a result, a solution must be found that balances both emissions and fuel costs. It can have been attained by 'combined economic and emission dispatch (CEED)' problem. The key goal of the CEED problem is to concurrently reduce fuel costs and emissions while meeting equality and inequality limits plus load demand. Mathematically, the ELD and CEED problem can be expressed as an optimization (minimization) problem, as shown below.

1.1 Mathematical problem formulation

Economic load dispatch (ELD)

In ELD the total fuel cost ($/hr) can be mathematical expressed as below:

$$F_t = \sum_{i=1}^{n} (a_i P_i^2 + b_i P_i + c_i) \tag{1}$$

Also, in view of valve-point loadings effect the fuel cost function given as follows.

$$F_t = \sum_{i=1}^{n} \left[(a_i + b_i P_i + c_i P_i^2) + \left| e_i \sin\left(f_i \left(P_i^{min} - P_i \right) \right) \right| \right] \tag{2}$$

where F_t: total fuel cost of generations ($/hr); a_i, b_i and c_i: cost factors of a generator i; e_i and f_i: fuel cost quantities for valve point effects of a generator i; P_i: power output of the i^{th} generator.

The constraints of ELD problems are listed as follows.

- Generator limits

$$P_i^{\min} \leq P_i \leq P_i^{\max}$$

where $P_i^{\min}$ (minimum) and $P_i^{\max}$ (maximum) power generation by unit i.

- Power balance

$$\sum_{i=1}^{n} P_i = P_D(\text{total load demand}) + P_L(\text{total transmission line loss})$$

$$P_L = \sum_{i=1}^{n}\sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} P_i B_{oi} + B_{oo}$$

where B_{ij}, B_{oi} and B_{oo} are transmission loss coefficient.

- Prohibited operating zone

$$P_i^{\min} \leq P_i \leq P_{i,1}^{l} : P_{i,k-1}^{u} \leq P_i \leq P_{i,k}^{l} \text{:}\ P_{i,n_i}^{u} \leq P_i \leq P_i^{\max};\ k = 2,3, \dots n_i$$

where n_i: number of prohibited operating zone and $P_{i,k}^{l}$ & $P_{i,k}^{u}$: lower and upper limit of k^{th} prohibited zone of generating unit i.

- Ramp rate limit

$$\max\left(P_i^{\min}, P_i^{t-1} - \mathrm{DR}_i\right) \leq P_i^t \leq \min(P_i^{\max}, P_i^{t-1} + \mathrm{UR}_i)$$

where P_i^t & P_i^{t-1} current & previous output power and UR_i & DR_i: up & down ramp limit of generating unit i.

Combined economic emission dispatch (CEED)

When generator units burn fossil fuels, pollutants such as SOx, NOx, and COx are released into the atmosphere. The overall emission of these pollutants, known as emission constrained dispatch (ECD), can be written as.

$$E_t = \sum_{i=1}^{n}\left[\underbrace{10^{-2}(\alpha_i P_i^2 + \beta_i P_i + \gamma_i)}_{\text{quadratic function}} + \underbrace{\psi_i \exp(\lambda_i P_i)}_{\text{exponential function}}\right] \quad (3)$$

where E_t: total amount of emissions (lb/hr) and α_i , β_i , γ_i, ψ_i λ_i: emission coefficients of the i^{th}unit. Moreover, simultaneously minimizing two objective function F_t and E_t is the main target of the CEED problem. By a price penalty factor (h) methodology this bi-objective problem can be transformed into a single objective problem as follows.

$$\varphi_t = F_t + h \times E_t \quad (4)$$

where φ_t; total cost of the system operation. The price penalty factor h can be calculated by the following procedures for a particular load demand.

(i). Calculate the ratio $\frac{F_t(P_i^{\max})}{E_t(P_i^{\max})} = h_i\, , i = 2,3, \dots n$ \$/kg.
(ii). Sort the obtained h_i values in ascending order.
(iii). Add $P_i^{\max}$ of each unit one at a time starting from the unit with smallest h_i until $\sum P_i^{\max} \geq P_D$.
(iv). Catch the last value of h_i that attains the previous situation which signifies the price penalty factor for the given load.

Equation (5) can be modified as follow (for providing a balance between minimization of the fuel cost and emission).

$$\varphi_t = w \times F_t + (1 - w) \times h \times E_t \quad (5)$$

where w (specifies type of the optimization problem) is the weight factor and if –

(i). w =1 infers ELD problem
(ii). w =0 implies ECD problem
(iii). w =0.5 indicates CEED problem

1.2 Related literature survey

To handle engineering optimization challenges, many traditional optimization techniques such as Newton and quasi-Newton have been created. Moreover, they have some intrinsic limitations, such as high computing cost, local optimal stagnation, and search space derivation (Simpson et al. 1994). It's also challenging to locate the best solution during the problem-solving process. To circumvent the limitations of traditional optimization approaches, meta-heuristics algorithms (MAs) have been developed to handle complicated engineering optimization problems. The MAs can be separated into 4 sets according to the mechanical variances as-

MAs	Motivated from	Example
Evolutionary algorithms (EAs)	biology	genetic algorithm (GA) (Davis 1991), differential evolution (DE) (Storn and Price 1997), Backtracking Search Optimization Algorithm (BSA) (Civicioglu P 2013; Hassan and Rashid 2019a; Hassan and Rashid 2019a) etc.
Swarm intelligence algorithms (SIAs)	social bugs	particle swarm optimization (PSO) (Kennedy and Eberhart 1995), artificial bee colony (ABC) (Karaboga and Basturk 2007), cuckoo search (CS) (Yang and Deb 2009), krill herd (KH) (Gandomi and Alavi 2012) grey wolf optimizer(GWO) (Mirjalili et al. 2014) dragonfly algorithm (DA) (Mirjalili 2016), harris hawks optimization (HHO) (Heidari et al. 2019) etc.
Physics-based algorithms (PBAs)	natural phenomenon	harmony search (HS) (Geem et al. 2001), gravitational search algorithm (GSA) (Rashedi et al. 2009), water cycle algorithm (WCA) (Eskandar et al. 2012), equilibrium optimizer (EO) (Faramarzi et al. 2019) etc.
Human behaviour based algorithms (HBAs)	human being	teaching-learning-based optimization (TLBO) (Rao et al. 2011), mine blast algorithm (MBA) (Sadollah et al. 2013) etc.

The employment of the "trial-and-error" method in looking for solutions is the fundamental advantage of these algorithms. As a result, these methods have been successfully used to global optimization issues. Between popular MAs, PSO and DE successfully applied in diverse optimization parts due to their dominant search ability and simple arrangement. However, various drawbacks of DE and PSO limit their application in complex optimization contexts.

The fundamental drawback of PSO is that it can quickly become trapped in a local optimal solution zone. As a result, in PSO, speeding convergence speed and avoiding local optimal solutions are two essential challenges. To address such concerns, many potential changes to the PSO have been proposed in recent literature. Espitia and Sofrony (2018) proposed VPSO in which particle vortex behavior and particle circular motions are implemented for improving search capacity and escaping the local minima respectively. To find the optimum of the current search and gradually explore the search space, Yu et al. (2018) proposed SHPSO by the implementation of social learning PSO. Chen et al. (2018) devised PSOCO where two distinct crossover operations are taken to produce promising exemplars in order to balance diversity. A self-adaptive tool and unique factor (w, c_1, c_2) utilized for enhancing each particle position depending on their fitness in UAPSO which is devised by Isiet and Gadala (2019). Hosseini et al. (2019) developed HAFPSO where fractional-order derivatives and hunter-attack strategy are used to accelerate convergence and avoid stagnation respectively. Khajeh et al. (2019) proposed MPSO where a novel particle initializing scheme with random distribution is used for uniformly covering the search space. Ang et al. (2019) invented CMPSOWV, in which two diversity maintenance schemes (multi-swarm technique and probabilistic mutation operator) are used to prevent the premature convergence. Lanlan et al. (2020) proposed NOPSO where non-inertial velocities update formula, opposition based learning strategy and adaptive elite mutation strategies are employed to avoid local optimum trapping.

Xiong et al. (2020) proposed NMSPSO where three strategies- novel information exchange strategy (for information transfer between sub-swarms), novel leaning strategy (for speed up the convergence) and novel mutation strategy (for better exploration) are incorporated. In the ground of real-world problems, DE also has noteworthy performance and become a great optimizer. Still, it has few concerns like local exploitation ability and convergence rate. So as to reduce its concerns, hordes of effective and robust DE has been intended in the literature. Qiu et al. (2018) proposed MMDE where a novel bottom-boosting mechanism (to maintain the reliability), partial-regeneration strategy (to identified the promising solutions) and mutation operator DE/current/1 (to explore over solution space) are introduced. Zhang and Li (2018) developed DEPS in which a modified parent selection scheme is chosen to use the experience of successful parents while selecting them in mutation operator. Huang et al. (2018) invented hypercube-based NCDE where hypercube neighbourhood based mutation (to maintain the neighborhood size in a reasonable range) and self-adaptive techniques (to control the hypercube's radius vector) are used. Yang et al. (2019) developed daDE in which a modified mutation rule is created to utilize the information of the current and the former individual's altogether. It has great benefits on the robustness and convergence speed. Liu et al. (2019) proposed HDEMCO where two layers' top layer (where multi cross operation perform that provides rapid convergence) and bottom layer (where success-history-based adaptive DE is implemented for better global search) are considered. Gui et al. (2019) devised MRDE in which different trial vector generation strategies, regroup scheme and an adaptive strategy are performed to speed up the convergence rate. Li et al. (2019) developed EJADE where a sorting mechanism and a dynamic population reduction strategy are employed to speed up the convergence rate and maintain the diversity respectively. Hu et al. (2020) invented BADE where an annealing strategy allow the searching space to explore and accelerate the convergence too. Ben (2020) proposed ADE where initial population and a new difference vector (in mutation phase) are created by the knowledge of damage scenario structure and dispersion of individuals respectively.

Moreover, to increase the performance of a single algorithm, one of the primary research advices is the hybrid strategy. Because of diverse optimization methods have dissimilar search behaviours and benefits. Hence, to reduce their individual weaknesses (like premature convergence, stacks at local optima etc.), hybrid methods are now preferred more to solve complex optimization problems. As a result, many hybrid algorithms for DE and PSO are presented in the literature in order to improve their performance. Seyedmahmoudian et al. (2015) proposed DEPSO, where DE is employed to adds diversity on traditional PSO. However, it may not appropriate for multimodal optimization problems. Parouha and Das (2015) devised DPD in which DE is executed in the inferior and superior groups, while PSO is employed in the mid-group. But, for solving complex real-world problems it may need some moderations. Tang et al. (2016) proposed HNTVPSO-RBSADE, which employs a nonlinear time varying PSO and a ranking based self-adaptive DE to result in dynamic exploration and exploitation. Parouha and Das (2016a) developed MBDE in which swarm mutation and swarm crossover for DE is used to direct knowledge and improve the solution quality. Parouha and Das (2016b) proposed DE-PSO-DE in which the population is divided into three groups (A, B, & C) and executed in parallel manner. Famelis et al. (2017) devised DE-PSO where DE is merged with a velocity-update rule of PSO to enhance diversity. Mao et al. (2018) proposed DEMPSO in which DE is added first to lessen the search space and then acquired populations used modified PSO (MPSO) as an initial population to speed up the convergence rate. Tang et al. (2018) developed SAPSO – mSADE in which self-adaptive PSO (SAPSO) and modified self-adaptive DE (mSADE) are evolved to balance diversity and reduce potential stagnation respectively. Too et al. (2019) invented BPSODE where the strength of binary PSO (BPSO) and binary DE (BDE) are inherited and computed in sequence. Dash et al. (2020) proposed HDEPSO in which three DE operations (mutation, modified crossover and selection) are fused with the best particles of PSO for enhancing global searching ability. Parouha and Verma (2021) proposed innovative hybrid algorithm of DE and PSO for bound-unconstrained optimization and ELD problem with or without valve point effects. It integrated with novel PSO (to escape stagnation) and DE (to avoid premature convergence). Further, Verma and Parouha (2021) applied the innovative hybrid algorithm to solve non-convex dynamic economic dispatch problem and numerical, graphical as well as comparative results shows its capability to solve considered optimization problems.

Moreover, a related recent review of DE, PSO and their hybrids as well as other MAs variants for solving CEED problem are mentioned as further. Mahdi et al. (2018) proposed QBA, in which a cubic criterion function is employed to represent this problem to reduce the nonlinearities of the system. The addition of quantum behaviour in bats, which eventually contributes to the diversification of population and diversifies the foraging habitats. In addition, early convergence in BA can be prevented. Jiang et al. (2019) devised GPSOA, where it integrates PSO with gravitation laws of GSA. Here, the particle's velocity is reorganized by random support of PSO and GSA. Additionally, Weibull-based probability density function is also used, to designate the stochastic individualities of wind speed and output power. Rezaie et al. (2019) proposed CIHSA, which is the combination of IHSA and CHSA. Where IHSA has a suitable convergence rate with high accuracy and CHSA has a high strength to find the best solutions in altered runs. Moreover, to dynamically tuning the parameters, employing virtual harmony memories and generate random numbers it uses chaotic patterns. Goudarzi et al. (2019) proposed MGAIPSO, it utilizes three operators, an arithmetic crossover and a mutation operator from GA to produce elite off-springs and maintain diversity; a non-linear time-varying double-weighted technique from PSO to obtain a substantial balance between exploration and exploitation. Edwin Selva Rex et al. (2019) proposed a novel hybrid algorithm (GA–WOA) using GA and Whale optimization for solving CEED problem. This method verified on 4 different test systems and it is superior to other heuristic methods with slight increase in the CPU execution time. Rashid et al. (2020) invented MIW-PSO, in which a multiple inertia weight is incorporated in PSO to improve its convergence characteristics for minimizing fuel cost and pollutant emission in the uncertain energy production expense and random load. Bibi et al. (2020) developed GOA, where the comfort zone operator of GOA assists in extracting stupendous simulations results of minimized fitness of multi objective functions that shows the efficiency of GOA in term of highly optimal and feasible solution satisfying all the system equality and inequality operational constraint. Khatsu et al. (2020) proposed PPSO in which a linear and non-linear time varying weight inertia factor (LWF and NLWF) are introduced in PPSO to enhance its searching ability. Goyal et al. (2020) proposed BBO, where an optimum generator scheduling has been achieved by employing BBO with all system constraints. Sakthivel et al. (2020) proposed MOSSA, it integrates squirrel search algorithm along with Pareto-dominance principle to generate non-dominated solutions. Also, it employed outward elitist depository tool with flocking distance categorization (to retain the distribution diversity) and utilized fuzzy decision-making strategy (to select the finest cooperated solution). Ajayi and Heymann, (2021) devised MSA. It is encouraged by the effort of moths the moonlight towards. Also, to provide the essential spinning reserve capacity, it slated thermal generators as solar energy is intermittent in nature. Hassan et al. (2021) proposed CAEO, which uses the chaotic maps which enhance a variety of the solution spaces and improve the convergence capabilities to achieve the optimum solutions as well as avoid the local minima.

1.3 Research gaps (Inspiration/motivation)

Despite the fact that a large number of MAs have been introduced in the literature, they have not been able to solve a wide range of problems (Wolpert and Macready 1997). Particularly, for some problems an algorithm can produce satisfactory outcomes but not for others. As a result, for solving a variety of optimization problems there is a necessity to develop some efficient algorithms. Furthermore, over separate efforts hybrid algorithms are now favored more to solve complex real-world problems. Hence, from the inspiration of above mentioned facts and literature investigation motivation of this study is to design an effective and innovative hybrid algorithm with the combination of novel variants of MAs. Between popular MAs, PSO and DE as well as their hybrids successfully applied in diverse optimization parts due to their leading search ability and simple structure. Consequently, after a wide analysis of the literature on diverse alternatives of PSO and DE with their hybrids the subsequent resulting opinions is evaluated and encouraged form them.

(i). The PSO is largely dependent on its parameters (which direct particles to the optimum) and position update (to balance diversity). As a result, numerous investigators have attempted to improve the accuracy and speed of PSO by modifying its control parameters and position update equation.

(ii). In DE mutation and crossover schemes with their associated control factors, are used to generate the global best solution, which is favorable for refining convergence performance. Hence, the most appropriate schemes and their related factor are regarded as a critical research study for DE.

(iii). Owing to their efficiency in solving complex real-world problems hybrid methods have caught the more interest of investigators. As per investigation, PSO and DE have balancing properties and their hybrids have recently gained more popularity to solve numerous real-world problems. From now, to the best of our information, figuring out how to hybridize PSO and DE is still an open research problem.

1.4 Contribution

Inspired by above remarks and literature study planned the following for solving CEED problems.

(i). an innovative hybrid algorithm (ihPSODE): it integrated with suggested novel PSO (to increase the population diversity) and novel DE (it produce perturbations to avoid the algorithm trapping into local optima).

(ii). novel PSO (nPSO): it involves new gradually changeable (increasing and/or decreasing) parameters and new equation for position update.

(iii). novel DE (nDE): it includes combination of novel operators (mutation and crossover) and new related control parameters.

The proposed algorithms have validated on 23 typical unconstrained benchmark functions then used to solve 3 test systems (3, 6 and 40-unit test system) of ELD and 3 test systems (3, 10 and 40-unit test system) of CEED problem. Comparative experiments endorse the efficiency and superiority of the presented algorithms.

The rest of this paper is presented in the following. Section 2 presents the proposed algorithms in details. Proposed algorithms are tested on 23 unconstrained benchmark functions in Section 3. In Section 4, the proposed algorithms applied on 3 different test systems of ELD and CEED problems. Conclusions would be drawn in section 5 with future work.

2. Proposed methodology

This section briefs the basics PSO and DE then discussed and the proposed methodology in detail.

Particle Swarm Optimization (PSO) (Kennedy and Eberhart 1995)

In a D-dimensional optimization space $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ and $v_i = (v_{i,1}, v_{i,2}, \dots, v_{i,D})$ denote the position and velocity of the i^{th} particle, respectively. During the evolutionary process, each individual constantly adjusts its velocity and position by following the personal best experience $pbest_i = (pbest_{i,1}, pbest_{i,2}, \dots, pbest_{i,D})$ and the population best experience $gbest_j = (gbest_1, gbest_2, \dots, gbest_D)$. The specific mathematical formulations are presented as:

$$v_{i,j}^{t+1} = wv_{i,j}^{t} + c_1 r_1 (pbest_{i,j}^{t} - x_{i,j}^{t}) + c_2 r_2 \left(gbest_j^{t} - x_{i,j}^{t}\right) \tag{6}$$

$$x_{i,j}^{t+1} = x_{i,j}^{t} + v_{i,j}^{t+1} \tag{7}$$

where t denotes the number of iterations. c_1 and c_2 are two important control parameters called acceleration coefficients. $r_1$and r_2 are two randomly selected numbers within the range [0,1]. w is the inertia weight

Differential Evolution (DE) (Storn and Price 1997)

In DE algorithm, an initial population which includes np individuals are randomly generated in the D-dimensional solution space. In the searching space, each individual represents a candidate solution. Three main operations on DE given as follows.

(i). Mutation: After initialization, to increase the population diversity, DE hires the mutation operator to create a mutation vector (mutant individual) $v_{i,j}^{t}$ with respect to the target vector $x_{i,j}^{t}$ at generation t as

$$v_{i,j}^{t+1} = x_{r_1}^{t} + F\left(x_{r_2}^{t} - x_{r_3}^{t}\right) \tag{8}$$

where r_1, r_2 and r_3 are jointly different integers taken from [1, NP] and are distinct from the index i. The control parameter F is a scaling factor, which amplifies the difference vector.

(ii). Crossover: After the above operation, a binomial crossover which recombines the target vector $x_{i,j}^{t}$ and the mutation vector $v_{i,j}^{t+1}$, is usually applied to generate a new trail vector $u_{i,j}^{t+1}$ as follows.

$$u_{i,j}^{t+1} = \begin{cases} v_{i,j}^{t+1}\ ; \text{if } rand(j) \le C_r \text{ or } j = rnbr(i) \\ x_{i,j}^{t}\ ;\ \text{if } rand(j) > C_r \text{ or } j \ne rnbr(i) \end{cases} \tag{9}$$

where $j = 1, \ldots, D$. $rand(j)$ is a random number on the interval $[0, 1]$. Crossover rate C_r is another control parameter in DE algorithm. To ensure that $u_{i,j}^{t+1}$ get at least one parameter from $v_{i,j}^{t+1}$, $rnbr(i)$ is a randomly selected integer within $[1, D]$.

(iii). Selection: following the crossover operation, DE applies a greedy strategy to select a vector for the next generation as follows. If the trial vector $u_{i,j}^{t+1}$ has better or equal fitness, it will be retained in the next generation. Otherwise, the target vector $x_{i,j}^{t}$ will survive.

$$x_{i,j}^{t+1} = \begin{cases} u_{i,j}^{t+1}\ ;\ \text{if } f(u_{i,j}^{t+1}) \le f(x_{i,j}^{t}) \\ x_{i,j}^{t}\ ;\ \text{Otherwise} \end{cases} \tag{10}$$

2.1 novel PSO (nPSO)

The specific operation of nPSO may be separated as below steps.

1-step Generate initial population $x_i^t = \{x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,D}^t\}$ $i = 1, 2, \ldots, np$, randomly follows the uniform distribution.

2-step Compute the objective function value $f(x_i^t)$.

3-step $pbest$ and $gbest$ selection with updating.

4-step Each individual adjusts its velocity and position as follows-

As per earlier investigation in PSO-

(i). small and large w values effect exploration and exploitation in earlier and latter stages correspondingly.

(ii). acceleration coefficient c_1 (decreasing) and c_2 (increasing) supports personal and global best results individually.

(iii). accuracy of achieved solution is not high and particles oscillate near local optimum for next iterations as good situation can be used in positions update procedure.

Encouraged by above statistic in nPSO following parameter suggested.

(i). a linearly decreased w $\left(w = \frac{1}{2}\left(1 + \frac{t_{max} - t}{t_{max}}\right)\right)$.

(ii). gradually decreased c_1 $\left(c_1 = \left(\frac{1}{1+\frac{t}{t_{max}}}\right)\right)$ and gradually increased c_2 $\left(c_2 = \left(\frac{1}{2-\frac{t}{t_{max}}}\right)\right)$ acceleration factor.

(iii). $n_t = e^{\left(1-\frac{t_{max}+t}{t_{max}-t}\right)}$ (a non-linear decreasing factor), where t & t_{max} is current iterations & maximum number of iterations respectively, introduced in position update equation, it may have benefitted once nPSO executes local search in later iterations as particles get nearby to the global best result.

The offered factor behavior illustrated in Fig. 1 & Fig. 2.

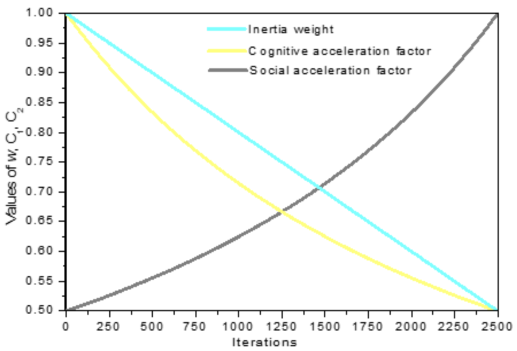


Fig. 1 w, c_1 & c_2 behavior during evolution process c_1

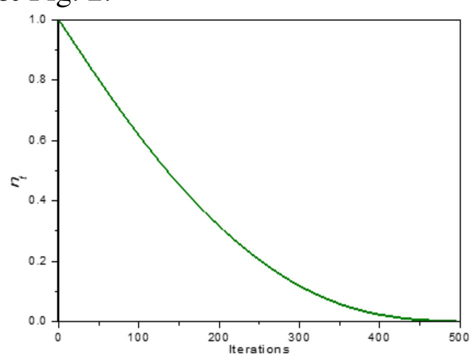


Fig. 2 Factor n_t behavior during evolution process

Under the above circumstance, eq. (6)- to update velocity and eq. (7)- to update position can be replaced as follows.

$$v_i^{t+1} = \left(\frac{1}{2}\left(1 + \frac{t_{max}-t}{t_{max}}\right)\right) v_i^t + \left(\frac{1}{1+\frac{t}{t_{max}}}\right) r_1 (pbest_i^t - x_i^t) + \left(\frac{1}{2-\frac{t}{t_{max}}}\right) r_2 (gbest - x_i^t) \quad (11)$$

$$x_i^{t+1} = x_i^t + n_t v_i^{t+1} \quad (12)$$

5-step Stop if reaching the convergence/termination condition otherwise repeat the steps from 2.

2.2 novel DE (nDE)

The specific operation of nDE may be divided into the following stages.

1-step In the D- dimensional solution space, an initial population generated uniform randomly which includes np individuals.

2-step Compute the objective function value.

3-step (i). Mutation

Mathematically, the proposed mutation strategy can be described as follows.

$$v_{i,j}^{t+1} = x_{i,j}^t + \tau \times rand(0,1) \times (best_{i,j}^t - x_{i,j}^t) \quad (13)$$

where v_i^t: mutant vector, x_i^t- target vector, τ: convergence parameter, $rand(0,1)$: uniformly scattered random number among 0 & 1 and $best_i$: best vector.

In eq. (13), importance of τ has an significant influence on presentation of nDE & i*h*PSODE (defined below). The vibrant amendments of τ are specified as below and its behavior depicted in Fig. 3 as early, middle and later stages on evolution procedure.

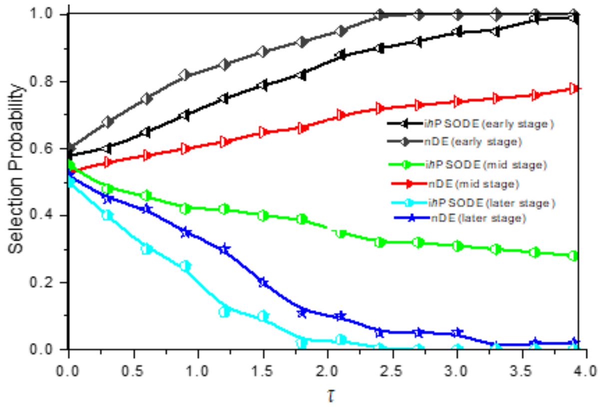


Fig. 3 Factor τ behavior during evolution process

it may observe that from Fig. 3 during the evolution following stages.

early	middle	later
the choice probability of mutation scheme is nearer to 100% and population diversity can be sustained effectually when $\tau \geq 1.5$. Thus, as far as possible i*h*PSODE and nDE is capable to examine additional promising regions.	the choice probability befits minor, which is not helpful for both to converge and hence τ would not be excessively large ($\tau < 2.5$). Thus, i*h*PSODE and nDE can accelerate search while retaining global exploration capability.	the choice probability of i*h*PSODE and nDE would be improved when $\tau \leq 2.2$, as i*h*PSODE and nDE can concentrate on improving local exploitation facility, convergence speed and precision.

As all, a smaller value of τ is helpful for global convergence ability as well as a larger value of τ is useful to retain population diversity. Therefore, advised to take the value of τ in between or equal to 1.5 & 2.2 which balance local exploitation and global exploration.

(ii). Crossover

it can be observed that from eq. (9) if C_r becomes with following values

larger	smaller
v_i^t (mutation vector) pays extra to the u_i^t (trial vector), it is useful to speed up local search and convergence rate, but it lose those target vectors (x_i^t) which have superior fitness.	x_i^t donates more to the u_i^t, it is helpful to preserve global search ability and population diversity, but it makes slow searching process and not able to generate new individual structures.

Hence, proposed novel $C_r = 0.9 - e^{\left(1-\frac{t_{max}}{t_{max}-t+1}\right)} \times 0.7$ is use in nDE to overcome the above issues. It can maintain population diversity and accelerate convergence of the nDE. Ultimately, 90% trial vectors can be created by mutation operators if C_r is equal to 0.9 along with as crossover probability increases it may upsurge mutation degree of vectors.

(iii). Selection

Following the crossover operation, nDE applies a greedy strategy eq. (10) to select a vector for the next generation.

4-step Stop if reaching the convergence/termination condition otherwise repeats the steps from 2.

2.3 innovative *hybrid* PSODE (i*h*PSODE)

Primarily, i*h*PSODE is created on involving higher competence of the contributed nPSO and nDE. The specific operation of *i*hDEPSO can be revealed as below stages.

1-step In the D- dimensional solution space, an initial population generated uniform randomly which includes np individuals.
2-step Compute the objective function value.
3-step Sort population according to according to their performance as well as recognize best half and remove the rest population.
4-step Apply nPSO
5-step Apply nDE (in offspring generated by nPSO)
6-step Merge the population produced by nPSO and nDE
7-step Stop if reaching the termination condition otherwise repeats the steps from 2.

3. Validation of presented algorithms

To confirm the performance of the presented algorithms, experiments are investigated on twenty-one classical unconstrained benchmark functions (cubf$_s$). Among these functions, $f_1 \sim f_7$, $f_8 \sim f_{13}$ and $f_{14} \sim f_{23}$ are unimodal, multimodal and fixed-dimension benchmark functions, respectively. The data of these cubf$_s$ are scheduled in Table 1. Simulations were piloted on Intel (R) Core (TM) i5-2350, CPU @ 2.30GHz with 4 GB RAM and simulated in C language (C-free Standard 4.0). Furthermore, to handle constraint, a penalty term is added to the objective function. Because of its higher efficiency, the bracket operator penalty (Deb 1995) was chosen for this study. Besides after several tryouts fine-tuning value of R= $1e^{03}$ is recommended for presented algorithms. In each table, overall best values are emphasized with bold of corresponding algorithms.

The produced result by proposed algorithms on 23 cubfs is compared with traditional algorithms (HHO (Heidari et al. 2019) & EO (Faramarzi et al. 2019)), PSO variants (HEPSO (Mahmoodabadi et al. 2014) & RPSOLF (Yan et al. 2017)), DE variants (JADE (Zhang and Sanderson 2009) & SHADE (Tanabe and Fukunaga 2013)) and hybrid variants (FAPSO (Xia et al. 2018), & PSOSCALF (Chegini et al. 2018)). Table 2 lists the parameters of all of the above-mentioned compared and proposed algorithms. Stopping criteria, independent run and population size of presented algorithms are taken as same or the least of comparative methods for fair comparison. Table 3 shows the comparative experimental results of 30 independent runs in terms of mean/average value (avg), standard deviation (std), and ranking (rank) of the objective function values. The comparative results of the algorithms directly from the original papers.

Table 1 Classical unconstrained benchmark functions (cubf$_s$)

cubf$_s$ formulation	Type	Dim.	Range	f_{min}
$f_1(x) = \sum_{i=1}^{D} x_i^2$	unimodal	30	[-100,100]	0
$f_2(x) = \sum_{i=1}^{D}\lvert x\rvert + \prod_{i=1}^{D}\lvert x_i\rvert$		30	[-10, 10]	0
$f_3(x) = \sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$		30	[-100,100]	0
$f_4(x) = max_i\lvert x_i\rvert, 1 \le i \le D$		30	[-100,100]	0
$f_5(x) = \sum_{i}^{D-1}[100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$		30	[-30, 30]	0
$f_6(x) = \sum_{i}^{D}(\lfloor x_i + 0.5\rfloor)^2$		30	[-100,100]	0
$f_7(x) = (\sum_{i}^{D} i\, x_i^4) + rand[0,1)$		30	[-1.28, 1.28]	0
$f_8(x) = \sum_{i}^{D} -x_i sin(\sqrt{\lvert x_i\rvert})$	multimodal	30	[-500,500]	-418.9829*D
$f_9(x) = \sum_{i}^{D}[x_i^2 - 10cos(2\pi xi) + 10]$		30	[-5.12,5.12]	0
$f_{10}(x) = -20exp\left(-\sqrt[0.2]{\frac{1}{D}\sum_{i}^{D} {x_i}^2}\right) - \exp\left(\frac{1}{D}\sum_{i}^{D} cos2\pi xi\right) + 20 + e$		30	[-32,32]	0
$f_{11}(x) = \frac{1}{4000}\sum_{i}^{D} x_i^2 - \prod_{i}^{D} cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$		30	[-600, 600]	0
$f_{12}(x) = \frac{\pi}{D}\left\{10sin^2(xy) + \sum_{i}^{D}(x_i - 1)^2\left(1 + sin^2(xy_{i+1)})\right) + (y_D - 1)^2\right\} + \sum_{i}^{D} U(x_i, 10,100,4)$		30	[-50,50]	0
$f_{13}(x) = 0.1\{sin^2(3\pi x_i)\} + \sum_{i=1}^{D}(x_i - 1)^2\,[1 + sin^2(3\pi x_i + 1)] + (x_D - 1)^2 + \sum_{i}^{D} U(x_i, 5, 100,4)$		30	[-50,50]	0
$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25}\left(\frac{1}{j+1+\sum_{i=0}^{1}(x_i - a_{ij})^6}\right)\right)^{-1}$	fixed-dimension	2	[-65, 65]]	1
$f_{15}(x) = \sum_{i=0}^{10}\left(a_i - \frac{x_0(b_i^2+b_i x_1)}{(b_i^2+b_i x_2+x_3)}\right)^2$		4	[-5,5]	0.00030
$f_{16}(x) = 4x_0^2 - 2.1x_0^4 + \frac{1}{3}x_0^6 + x_0x_1 - 4x_1^2 + 4x_1^4$		2	[-5,5]	−1.0316
$f_{17}(x) = \left(x_1 - \frac{5.1}{4\pi^2}x_0^2 + \frac{5}{\pi}x_0 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_0 + 10$		2	[-5,5]	0.398
$f_{18}(x) = 1 + (x_0 + x_1 + 1)^2(19 - 14x_0 + 3x_0^2 - 14x_1 - 6x_0x_1 + 3x_1^2)\{30 + (2x_0 - 3x_1)^2(18 - 32x_0 + 12x_0^2 + 48x_1 - 36x_0x_1 + 27x_1^2)\}$		2	[-2,2]	3
$f_{19}(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$		3	[1,3]	−3.86
$f_{20}(x) = -\sum_{i=1}^{4} c_i exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$		6	[0,1]	−3.32
$f_{21}(x) = -\sum_{i=1}^{5}((x - a_i)^T(x - a_i) + c_i)^{-1}$		4	[0,10]	−10.1532
$f_{22}(x) = -\sum_{i=1}^{7}((x - a_i)^T(x - a_i) + c_i)^{-1}$		4	[0,10]	−10.4028
$f_{23}(x) = -\sum_{i=1}^{10}((x - a_i)^T(x - a_i) + c_i)^{-1}$		4	[0,10]	−10.5363

Table 2 Parameter for cubf$_s$ of compared and presented algorithms

Algorithm	Reference	Control Parameter		Population Size	Stopping criterion	Run
		Term	Values			
HHO	(Heidari et al. 2019)	escaping energy	$E < 0.5$, $E \ge 0.5$	30	500	30
EO	(Faramarzi et al. 2019)	a_1, a_2, GP	{1,1.5,2,2.5,3},{0.1,0.5,1,1.5,2},(0.1.0.25,0.5,0.75,0.9}	30	500	30
HEPSO	(Mahmoodabadi et al. 2014)	P_C, P_B	0.95,0.02	50	500	30
RPSOLF	(Yan et al. 2017)	w,c_1,c_2,c_3, β, ε	0.55,1.49,1.49,1.5,0.99	50	500	30
JADE	(Zhang and Sanderson 2009)	F_i, CR_i,	randc$_i$(μ_F , 0.1),randn$_i$(μ_{CR}, 0.1)	50	1000	30
SHADE	(Tanabe and Fukunaga 2013)	P*best*, Arc rate	0.1, 2	30	500	30
FAPSO	(Xia et al. 2018)	-	-	50	5000	30
PSOSCALF	(Chegini et al. 2018)	w_{min}, w_{max}, c_{1min}, c_{1max},c_{2min}, c_{2max}, β	0.4,0.9,0.5,2.5,0.5,2.5,1.5	50	500	30
nPSO	Presented algorithms	-	-	30	500	30
nDE		τ	[1.5,2.2]	30	500	30
i*h*PSODE		-	-	30	500	30

It should have been noted that the average objective function values of the presented algorithms (nPSO, nDE and ihPSODE) are better and/or equal to the compared standard algorithms, PSO alternatives, DE alternatives, and hybrid variants, as shown in Table 3. The presented algorithms produce less std for most of the cubfs which terms their stability. Also, all algorithms are ranked separately (as 1 for best, 2 for following performer and so on) grounded on average result values in Table 3. From Table 3 it is decided that i*h*PSODE, nDE & nPSO ranked as 1st, 2nd & 4th successively. As well, average and overall rank of presented versus others algorithms are declared in Table 3. Then, it can be say that the proposed algorithms outperform others by rankings. Furthermore, the supremacy of the proposed algorithms over other algorithms is analyzed statistically using- (i) a one-tailed t-test (at 5% significance level of 98 degrees of freedom (df)) and (ii) Wilcoxon Signed Rank (WSR) test (at 5% significance level). The details of these tests can be found in (Das and Parouha 2015). In Table 4, results of *t*- and WSR statistical test on cubf$_s$ are reported. In most of consequence, it can be perceived that from this table presented algorithms has 'a$^+$' or 'a' sign in case of t-test which signify highly or significantly better than other respectively and '+' or '≈' sign in case WSR test which indicate better or equally performances. Moreover, *p*-values of the presented algorithms listed in Table 4 which is lower than others and indicating that their reliability to produce best results on the majority of runs.

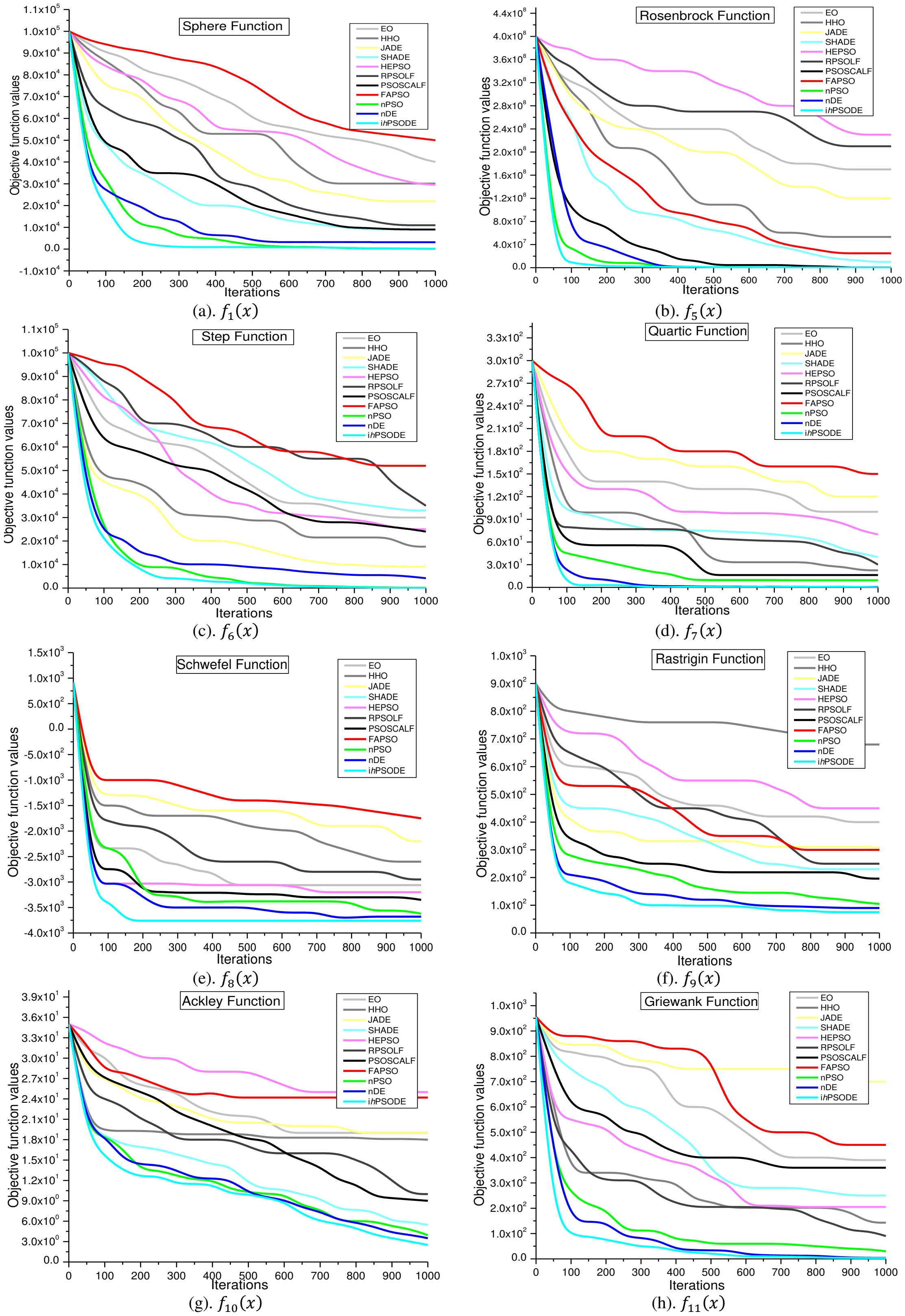
Table 3 Simulation results on cubf$_s$

$f_n(x)$	Criteria	Algorithms										
		Classical algorithms		PSO alternatives		DE alternatives		Hybrid alternatives		Presented algorithms		
		HHO	EO	HEPSO	RPSOLF	JADE	SHADE	FAPSO	PSOSCALF	nPSO	nDE	*ih*PSODE
$f_1(x)$	avg	2.03e+00	3.32e-40	16.26772	5.065e-269	1.87e-31	1.42e-09	2.87e−127	1.11014e-20	**0**	**0**	**0**
	std	4.04e−01	6.78e-40	10.01293	**0.00e+00**	6.43e-31	3.09e-09	1.76e−127	1.83289e-20	**0**	**0**	**0**
	rank	9	4	8	2	5	7	3	6	**1**	**1**	**1**
$f_2(x)$	avg	1.70e+00	7.12e-23	1.28424	1.000e-134	2.79e-15	0.0087	1.02e-17	4.09460e-11	**0**	**0**	**0**
	std	7.37e−02	6.36e-23	0.41611	3.753e-134	9.51e-15	0.0213	1.43e-17	5.68981e-11	**0**	**0**	**0**
	rank	9	3	8	2	4	7	5	6	**1**	**1**	**1**
$f_3(x)$	avg	1.17e+02	8.06e-09	7.423e+03	7.791e-249	1.10e-03	15.4352	1.68e-11	2.16858e-12	0.17e-129	**0**	**0**
	std	5.28e+00	1.60e-08	7.423e+03	0.00e+00	5.14e-03	9.9489	2.49e-11	1.03815e-11	1.67e-131	**0**	**0**
	rank	9	6	10	2	7	8	5	4	3	**1**	**1**
$f_4(x)$	avg	2.05e+00	5.39e-10	23.95145	1.937e-157	1.66e-03	0.9796	4.09e+03	8.47410e-08	8.35e-098	3.28e-101	**0**
	std	7.40e−02	1.38e-09	7.71460	1.061e-156	1.98e-03	0.7995	6.53e+02	1.23324e-07	7.08e-098	7.12e-103	**0**
	rank	9	5	10	2	7	8	11	6	4	3	**1**
$f_5(x)$	avg	2.95e+00	2.53e+01	2.380e+03	27.42672	1.18e+01	24.4743	6.55e-11	21.97646	4.85e-012	1.25e-021	**2.21e-033**
	std	8.36e−02	0.16e+00	1.852e+03	0.24848	1.57e+01	11.2080	1.99e-11	0.54774	3.21e-012	1.85e-023	**3.72e-037**
	rank	9	7	11	8	10	6	4	5	3	2	**1**
$f_6(x)$	avg	2.49e+00	8.29e-06	21.55405	2.98244	4.59e-31	5.31e-10	2.37e-12	7.13998e-12	1.75e-032	**0**	**0**
	std	8.25e−02	5.02e-06	9.33263	0.23250	1.65e-30	6.35e-10	1.84e-13	3.65884e-11	9.16e-035	**0**	**0**
	rank	9	8	6	10	3	7	4	5	2	**1**	**1**
$f_7(x)$	avg	8.20e+00	1.17e-02	0.12982	0.00104	6.49e-03	0.0235	**0**	0.00012	5.11e-001	2.19e-003	1.07e-003
	std	1.69e−01	6.54e-04	0.09727	7.644e-04	2.48e-03	0.0088	**0**	0.00010	1.70e-002	1.10e-004	1.40e-005
	rank	11	6	10	7	5	9	**1**	4	8	3	2
$f_8(x)$	avg	4.86e+00	-9016.34	-2.139e+03	-3.254e+03	**-1.24e+04**	-11713.1	2.48e-11	**-12569.48**	-6.37e+004	**-1.25e+004**	**-1.25e+004**
	std	1.03e+00	595.1113	8.282e+02	2.860e+02	1.27e+02	230.49	6.44e-12	2.39996e-07	2.10e-001	1.07e-017	**0**
	rank	4	6	5	7	**1**	8	2	**1**	9	**1**	**1**
$f_9(x)$	avg	3.77e+00	**0**	42.00118	**0**	1.71e-04	8.5332	**0**	**0**	**0**	**0**	**0**
	std	8.87e−01	**0**	7.08632	**0**	1.52e-04	2.1959	**0**	**0**	**0**	**0**	**0**
	rank	3	**1**	5	**1**	2	4	**1**	**1**	**1**	**1**	**1**
$f_{10}(x)$	avg	3.75e+00	8.34e-14	2.83842	4.085e-15	1.31e-14	0.3957	4.86e-15	2.24609e-11	1.97e-014	1.44e-015	**2.88e-016**
	std	8.75e−01	2.53e-14	0.66134	1.084e-15	2.46e-14	0.5868	1.74e-15	2.33542e-11	**0**	**0**	**0**
	rank	8	7	9	3	5	10	4	8	6	2	**1**
$f_{11}(x)$	avg	4.17e+00	**0**	1.16858	**0**	2.87e-03	0.0048	1.74e-16	**0**	3.37e-111	**0**	**0**
	std	5.56e−01	**0**	0.12602	**0**	7.85e-03	0.0077	3.60e-16	**0**	1.11e-119	**0**	**0**
	rank	7	**1**	6	**1**	4	5	3	**1**	2	**1**	**1**
$f_{12}(x)$	avg	1.90e+01	7.97e-07	0.47856	0.26157	1.73e-02	0.0346	1.57e-32	8.46465e-14	3.34e-002	1.05e-032	**3.73e-033**
	std	3.31e+00	7.69e-07	0.22623	0.03386	7.74e-02	0.0875	0.00e+00	2.79106e-13	1.02e-004	2.77e-034	**3.18e-034**
	rank	11	5	10	9	6	8	3	4	7	2	**1**
$f_{13}(x)$	avg	1.89e+01	0.029295	1.85056	2.05282	5.45e-24	7.32e-04	**1.58e-32**	0.00399	9.30e-004	2.09e-021	**1.58e-032**
	std	1.56e+00	0.035271	0.65246	0.16579	2.58e-23	0.0028	**0**	0.00928	3.71e-004	3.27e-023	**1.89e-043**
	rank	10	7	8	9	2	4	**0**	6	5	3	**1**
$f_{14}(x)$	avg	**9.98e−01**	**0.99800**	**0.99800**	1.54064	**9.98e-01**	**0.998004**	**9.98e-001**	1.13027	**9.98e-001**	**9.98e-001**	**9.98e-001**
	std	9.23e−01	1.54e-16	9.219e-17	1.84429	**0**	5.83e-17	1.27e-08	0.50338	**0**	**0**	**0**
	rank	**1**	**1**	**1**	3	**0**	**1**	**1**	2	**0**	**0**	**0**
$f_{15}(x)$	avg	3.10e−04	0.00239	6.404e-04	0.00171	3.01e-03	0.002374	3.95e-04	3.13244e-04	3.99e-004	3.83e-004	**3.02e-004**
	std	1.97e−04	0.00609	2.801e-04	0.00508	6.92e-03	0.0061	6.02e-08	2.17489e-05	2.96e-007	9.23e-012	**2.38e-018**
	rank	2	11	7	9	8	10	5	3	6	4	**1**
$f_{16}(x)$	avg	**−1.03e+00**	**-1.03161**	**-1.03161**	**-1.03161**	**-1.03e+00**	**-1.03162**	**−1.03e+00**	**-1.0316**	**−1.03e+000**	**−1.03e+000**	**−1.03e+000**
	std	6.78e−16	6.04e-16	3.554e-15	1.650e-05	6.78e-16	6.51e-16	**0**	4.40244e-16	**0**	**0**	**0**
	rank	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**1**	**0**	**1**	**1**
$f_{17}(x)$	avg	**3.98e−01**	0.397887	0.39787	**0.39837**	**3.98e-01**	0.397887	**3.98e-001**	**0.39788**	**3.98e-001**	**3.98e-001**	**3.98e-001**
	std	2.54e−06	**0**	6.594e-13	5.267e-04	**0.00e+00**	3.24e-16	**0.00e+000**	3.66527e-15	**0**	**0**	**0**
	rank	**1**	2	2	**1**	**1**	2	**1**	2	**1**	**1**	**1**
$f_{18}(x)$	avg	**3**	**3**	0.65246	**3.00002**	**3**	**3**	**3**	**3**	**3**	**3**	**3**
	std	**0**	1.56e-15	5.146e-11	1.658e-05	1.82-15	1.87e-15	5.31e-016	5.96540e-13	1.33e-018	**0**	**0**
	rank	**1**	**1**	2	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
$f_{19}(x)$	avg	**−3.86e+00**	**-3.86278**	**-3.86278**	-3.85923	**-3.86e+00**	**-3.86278**	−3.87e+000	**-3.86278**	**−3.86e+000**	**−3.86e+000**	**−3.86e+000**
	std	2.44e−03	2.59e-15	1.008e-13	0.00283	2.71e-15	2.69e-15	2.11e-004	8.31755e-15	3.36e-021	**0**	**0**
	rank	**1**	**1**	**1**	3	**1**	**1**	2	1	**1**	**1**	**1**
$f_{20}(x)$	avg	**−3.322**	-3.2687	-3.31803	-3.10441	-3.29e+00	-3.27047	-3.29e+000	-3.27168	-3.27e+000	**-3.32e+000**	**-3.32e+000**
	std	0.137406	0.05701	0.02170	0.15760	5.11e-02	0.0599	**0**	0.06371	**0**	**0**	**0**
	rank	**1**	6	2	7	4	5	3	5	5	**1**	**1**
$f_{21}(x)$	avg	**−10. 1531**	-8.55481	**-10.15319**	-4.76171	-9.14e+00	-9.2343	-7.51e+00	**-10.15319**	-9.87e+001	**-1.01e+001**	**-1.01e+001**
	std	0.885673	2.76377	2.680e-05	0.73723	2.06e+00	1.3969	1.21e-01	4.46227e-15	1.07e-011	**0**	**0**
	rank	**1**	5	**1**	6	4	3	3	**1**	2	**1**	**1**
$f_{22}(x)$	avg	**−10.4015**	-9.3353	**-10.39978**	-4.81927	-9.88e+00	-10.2809	-6.04e+01	**-10.40294**	-9.87e+000	**-1.04e+001**	**-1.04e+001**
	std	1.352375	2.43834	0.01728	0.75699	1.61e+00	1.3995	2.14e-01	1.80672e-15	2.08e-002	2.91e-012	**1.10e-016**
	rank	**1**	5	**1**	7	3	2	6	**1**	4	**1**	**1**
$f_{23}(x)$	avg	**−10.5364**	-9.63655	**-10.53640**	-5.06376	-1.03e+01	63.333	-5.64e+00	**-10.53640**	-9.13e+000	**-1.05e+001**	**-1.05e+001**
	std	0.927655	2.38811	5.845e-07	0.82968	1.40e+00	80.872	5.70e-02	4.84794e-15	2.05e-003	1.07e-021	**0**
	rank	**1**	3	**1**	6	2	7	5	**1**	4	**1**	**1**
Sum of rank		119	102	125	107	87	124	75	75	78	35	**24**
Average		5.17	4.43	5.43	4.65	3.78	5.39	3.26	3.26	3.39	1.52	**1.04**
Overall rank		8	6	10	7	5	9	3	3	4	2	**1**

Table 4 Statistical outcomes on cubf$_s$

Vs	Standards	Algorithm									
		Classical algorithms		PSO alternatives		DE alternatives		Hybrid alternatives		Presented algorithms	
		HHO	EO	HEPSO	RPSOLF	JADE	SHADE	FAPSO	PSOSCALF	nDE	*ih*PSODE
nPSO	Better	11	21	20	19	20	13	15	19	0	0
	Equal	4	2	2	2	2	9	4	3	7	8
	Worst	8	0	1	2	1	1	4	1	16	15
	R^+	293	387	312	323	335	305	300	382	350	400
	R^-	172	78	153	142	130	160	165	83	115	65
	p-value	5.1e-09	5.3e-10	5.7e-10	5.1e-09	6.2e-10	4.6e-08	5.8e-10	5.6e-10	6.2e-09	5.3e-10
	t-test	a	a	a	a+	a	a+	a+	a+	a	a+
	Decision	≈	≈	≈	+	+	+	+	+	+	+
Vs		HHO	EO	HEPSO	RPSOLF	JADE	SHADE	FAPSO	PSOSCALF	nPSO	*ih*PSODE
nDE	Better	15	21	21	21	20	13	14	18	15	0
	Equal	8	2	2	2	3	9	9	5	8	16
	Worst	0	0	0	0	0	1	0	0	0	7
	R^+	416	313	329	465	345	355	377	323	342	315
	R^-	49	152	136	79	120	130	88	142	123	150
	p-value	5.6e-10	5.2e-10	6.2e-10	6.9e-07	8.2e-10	5.8e-10	6.2e-11	4.3e-09	5.1e-10	6.9e-07
	t-test	a	a	a	a	a	a+	a+	a	a+	a+
	Decision	+	+	+	≈	+	≈	≈	+	+	+
Vs		HHO	EO	HEPSO	RPSOLF	JADE	SHADE	FAPSO	PSOSCALF	nDE	nPSO
*ih*PSODE	Better	14	0	20	20	20	13	14	15	8	15
	Equal	7	6	3	3	3	10	9	8	15	8
	Worst	2	17	0	0	0	0	0	0	0	0
	R^+	321	294	330	313	329	367	293	377	323	304
	R^-	144	171	135	152	136	98	172	88	142	161
	p-value	6.2e-10	5.1e-10	5.1e-07	5.1e-10	4.6e-08	5.7e-10	6.2e-09	5.3e-08	4.6e-10	5.7e-07
	t-test	a	a	a	a+	a	a+	a	a+	a+	a
	Decision	+	+	+	≈	≈	+	≈	+	+	+

To demonstrate that presented algorithms satisfied convergence speed, the convergence curves of comparative and presented algorithms on eight ($f_1(x)$, $f_5(x)$, $f_6(x)$, $f_7(x)$, $f_8(x)$, $f_9(x)$, $f_{10}(x)$ and $f_{11}(x)$) typical 30-*D* cubf$_s$ are plotted and presented separately in Fig. 4(a-h). From this figure it can be see that almost all of the considered benchmark functions, either unimodal or multimodal, would be quickly optimized by the presented algorithms (nPSO, nDE, and i*h*PSODE).



(a). $f_1(x)$ (b). $f_5(x)$ (c). $f_6(x)$ (d). $f_7(x)$ (e). $f_8(x)$ (f). $f_9(x)$ (g). $f_{10}(x)$ (h). $f_{11}(x)$

Fig. 4(a-h) Convergence of different algorithms

Likewise, an effort is completed to catch global optimal solution entire of 690 runs (30 population size with 30 runs for each cubf$_s$) and demonstrated in Fig. 5. It states that the presented algorithms provide the best optimal solutions. Apart from that, the computational time of the presented and equated algorithms on each cubf$_s$ is calculated and illustrated in Fig. 6 via box plots. This figure shows that the presented algorithms take less time to attain the best value for the entire set of cubf$_s$.

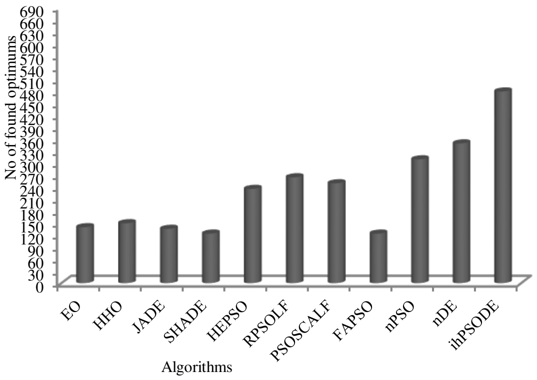


Fig. 5 Global optimal solution out of 690 runs

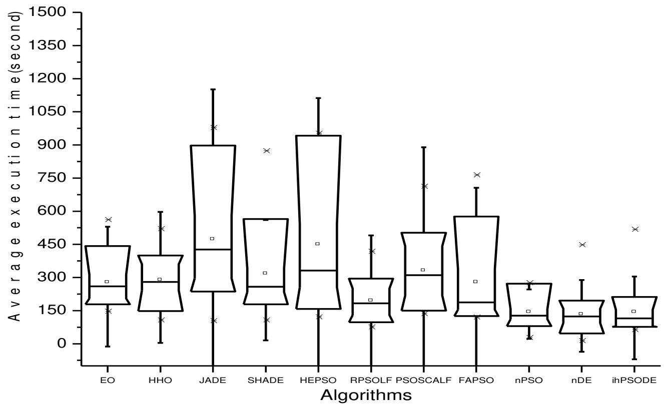


Fig. 6 Processing times of algorithms for cubf$_s$

In general, it can be decided that the performance of presented algorithms is superior to or at least equal to other intelligent optimization algorithms (classical, PSO, DE and Hybrid variants) on most test functions. In conclusion, presented algorithms (nPSO, nDE, and i*h*PSODE) can be considered as an effective and efficient method.

4. Presented algorithms for CEED problem

To further investigate the feasibility of presented algorithms (nPSO, nDE, and i*h*PSODE) in real-life problems, two large scale power engineering optimization problem (ELD and CEED) are considered here. These problem include 3 test systems (3, 6 and 40-unit test system) of ELD and 3 test systems (3, 10 and 40-unit test system) of CEED problem. The obtained best solutions are utilized to evaluate the feasibility of different algorithms.

Unit Test Systems

Problem	Unit Test Systems (UTS$_{ys}$)	Description
ELD	UTS-1 (3-unit test system) (Hardiansyah et al. 2013)	it involves load demand of 300MW.
	UTS-2 (6-unit test system) (Serapiao 2013)	it involves 700MW total load demands.
	UTS-3 (40-unit test system) (Hardiansyah 2013)	it consist valve loading point effect and involves load demand of 10500MW.
CEED	UTS-4 (3-unit test system) (Devi and Krishna 2008)	it consider emission impact and involves 400 MW and 500 MW load demand as well.
	UTS-5 (10-unit test system) (Basu 2011)	it consider valve point effects and involves 2000MW total load demand.
	UTS-6 (40-unit test system) (Basu 2011)	it consists of non-smooth fuel cost and emission functions and involves 10500MW total load demand.

Computational Steps of i*h*PSODE for CEED problem

The steps of i*h*PSODE for solving CEED problem are given as below:

1-step Read the P_D (Power Demand)
2-step Compute h (price penalty factor)
3-step t (iteration) = 1
4-step Generate initial population vector of real power generator (based on prohibited zone and ramp limit constraints)
5-step Evaluate the fitness function using equation (5)
6-step Sort the population (as per fitness function value)
7-step Apply nPSO (in best half population)

8-step Apply nDE (in offspring generated by nPSO)
9-step Merge the population produced by nPSO and nDE
10-step Stop if reaching the termination condition otherwise repeats the steps from 5.
11-step Print the results (generator schedule, minimized operating cost, corresponding fuel cost, and emission output)

4.1 Results and discussions

The results of presented hybrid algorithms (nPSO, nDE and i*h*PSODE) are compared with CDE-SQP (Coelho and Mariani 2006), PPSO (Chen and Yeh 2006), APPSO (Chen and Yeh 2006), SFL (Serapião 2009), BFO (Serapião 2009), CCPSO (Park et al. 2010), SOMA (Coelho and Mariani 2010), CSOMA (Coelho and Mariani 2010), DE/BBO (Bhattacharya and Chattopadhyay 2010), EDA/DE (Wang et al. 2010), ARCGA (Amjady and Rad 2010), TSAGA (Subbaraj et al. 2011), MODE (Basu 2011), NSGAII (Basu 2011), PDE (Basu 2011), SPEA-2 (Basu 2011), GA (Kumar and Alwarsamy 2012), PSO (Kumar and Alwarsamy 2012), GSA (Güvenç et al. 2012), ABC_PSO (Manteaw and Odero 2012), DE (Kumar and Alwarsamy 2012), QP (Hardiansyah et al. 2013), EMOCA (Zhang et al. 2013), SA (Hardiansyah et al. 2013), CS (Serapiao 2013), DHS (Wang and Li 2013), ABC (Serapiao 2013), FA (Serapiao 2013), MPSO (Hardiansyah 2013), MABC/D/Cat (Secui 2015) and MABC/D/Log (Secui 2015), WOA (Mirjalili and Lewis 2016), FPA (Abdelaziz et al. 2016), GA-WOA (Edwin Selva Rex et al. 2019), β-GWO (Betar et al. 2020), on ELD and CEED test systems.

For clearness, stopping criteria (500 iterations) population size (30) and independent run (25) of presented algorithms is taken same as relative algorithms. Rest parameter of proposed methods are same as above. The simulations result of presented algorithms with other comparative algorithms are listed in Table 5 (for UTS-1), Table 6 (for UTS-2), Table 7 (for UTS-3), Table 8 (for UTS-4), Table 9 (for UTS-5) and Table 10 (for UTS-6).

Table 5 Simulation results for UTS-1

Outputs (MW)	QP	SA	GA	PSO	DE	FPA	nPSO	nDE	i*h*PSODE
P^1	207.6799	207.6336	208.99	209.001	207.637	207.6316	204.3651	205.1546	205.3596
P^2	87.4010	87.2867	86.0041	85.92	87.2833	87.2886	81.3969	82.2746	82.0584
P^3	15.0000	15.0000	15.4163	15.0000	15.0000	15.0000	14.238	12.5708	14.582
P_L	10.0808	9.9203	10.4099	9.9833	9.9203	9.9202	8.6788	8.7645	7.7655
$\sum P_i$	310.0808	309.9203	310.4099	309.9211	309.9203	309.9202	308.6788	308.7645	309.7655
Cost($)	3621.50	3619.76	3624.28	3621.75	3619.8	3619.75	3619.88	3619.55	3619.45
CPU(s)	3.258	3.4017	1.4065	3.2065	4.503	0.4191	0.391	0.402	0.235

Table 6 Simulation results for UTS-2

Outputs (MW)	CS	ABC	FA	PSO	SFL	BFO	FPA	nPSO	nDE	i*h*PSODE
P^1	324.113	323.043	293.312	288.653	287.392	222.260	323.995	321.2581	321.001	321.995
P^2	76.859	54.965	79.546	82.753	67.637	58.777	76.846	73.4416	74.846	74.322
P^3	158.094	147.354	123.334	132.988	140.933	150.395	158.20	154.3169	155.4747	153.1017
P^4	50	50	69.7	50	98.357	106.963	50	50.0000	50.0000	50.0000
P^5	51.963	85.815	79.546	99.565	64.052	101.601	51.983	50.9834	50.1583	50.5813
P^6	50	50.233	63.778	57.768	53.15	72.559	50	50.0000	48.5200	50.0000
P_L	11.03	11.4	11.44	11.73	11.59	11.73	11.024	11.0122	11.0124	11.0245
$\sum P_i$	711.03	711.4	711.44	711.73	711.59	711.73	711.024	711.0122	711.0124	711.0245
Cost($)	8356.06	8372.27	8388.45	8401.45	8419.78	8428.69	8356.05	8368.8401	8359.2141	8356.1545
CPU(s)	2.65	4.51	3.56	1.99	2.54	1.25	0.796	1.851	1.018	0.653

Table 7 Simulation results for UTS-3

Outputs (MW)	PSO	PPSO	APPSO	MPSO	ARCGA	CCPSO	TSAGA	CDE_SQP	SOMA
P^1	113.116	111.601	112.579	113.9971	110.8252	114.0000	110.7998	111.7600	111.1110
P^2	113.01	111.781	111.553	112.6517	113.9112	111.0400	110.7999	111.5600	110.8299
P^3	119.702	118.613	98.751	119.4255	97.4000	97.3000	97.3999	97.3900	97.4122
P^4	81.647	179.819	180.384	189.0000	179.7331	179.6000	179.7331	179.7300	179.7443
P^5	95.062	92.443	94.389	96.8711	88.6454	90.7210	87.7999	91.6600	88.1510
P^6	139.209	139.846	139.943	139.2798	140.0000	140.0000	140.0000	140.0000	139.9959
P^7	299.127	296.703	298.937	223.5924	259.6000	260.0600	259.5997	300.0000	259.6065
P^8	287.491	284.566	285.827	284.5803	284.6000	285.8700	284.5997	300.0000	284.6045
P^9	292.316	285.164	298.381	216.4333	284.6000	284.7700	284.5997	284.5900	284.6149
P^{10}	279.273	203.859	130.212	239.3357	130.0000	130.0000	130.0000	130.0000	130.0002
P^{11}	169.766	94.283	94.385	314.8734	168.7985	94.0000	94.0000	168.7900	168.8029
P^{12}	94.344	94.090	169.583	305.0565	168.7994	168.3800	94.0000	94.0000	94.0000
P^{13}	214.871	304.830	214.617	365.5429	214.7600	214.4500	214.7598	214.7600	214.7591
P^{14}	304.790	304.173	304.886	493.3729	394.2800	394.0100	394.2794	394.2800	394.2716
P^{15}	304.563	304.302	304.467	304.547	280.4326	304.5200	394.2794	304.5200	304.5206
P^{16}	304.302	304.302	304.177	304.584	394.2800	304.5700	394.2794	304.5200	394.2834
P^{17}	489.173	489.544	498.452	435.2428	489.2798	489.2800	489.2794	489.2800	489.2912
P^{18}	491.336	489.773	497.472	417.6958	489.2800	489.5600	489.2794	489.2800	489.2877
P^{19}	510.880	511.280	512.816	532.1877	511.2806	511.2900	511.2794	511.2800	511.2977
P^{20}	511.474	510.904	548.992	409.2053	511.2800	511.2700	511.2794	511.2800	511.2791
P^{21}	524.814	524.092	524.652	534.0629	523.2803	523.2300	523.2794	523.2800	523.2958
P^{22}	524.775	523.121	523.399	457.0962	523.2800	523.6300	523.2794	523.2900	523.2849
P^{23}	525.563	523.242	548.895	441.3634	523.2800	523.8200	523.2794	523.2800	523.2856
P^{24}	522.712	524.260	525.871	397.3617	523.2800	523.6200	523.2794	23.2800	5523.2979
P^{25}	503.211	523.283	523.814	446.4181	523.2800	523.3300	523.2794	523.2800	523.2799
P^{26}	524.199	523.074	523.565	442.1164	523.2801	523.6800	523.2794	523.2800	523.2910
P^{27}	10.082	10.800	10.575	74.8622	10.0000	10.0000	10.0000	10.0000	10.0064
P^{28}	10.663	10.742	11.177	27.5430	10.0000	10.0000	10.0000	10.0000	10.0018
P^{29}	10.418	10.799	11.210	76.8314	10.0000	10.1600	10.0000	10.0000	10.0000
P^{30}	94.244	94.475	96.178	97.0000	88.7611	87.8700	87.8000	90.3300	96.2132
P^{31}	189.377	189.245	189.999	118.3775	190.0000	190.0000	190.0000	190.0000	189.9996
P^{32}	189.796	189.995	189.924	188.7517	190.0000	190.0000	190.0000	190.0000	189.9998
P^{33}	189.813	188.081	189.714	190.0000	190.0000	190.0000	190.0000	190.0000	189.9981
P^{34}	199.797	198.475	199.284	120.7029	164.8000	165.2300	164.7998	200.0000	164.9126
P^{35}	199.284	197.528	199.599	170.2403	164.8000	200.0000	194.3976	200.0000	199.9941
P^{36}	198.165	196.971	199.751	198.9897	164.8054	200.0000	200.0000	200.0000	200.0000
P^{37}	109.291	109.161	109.973	110.0000	110.0000	110.0000	110.0000	110.0000	109.9988
P^{38}	109.087	109.900	109.506	109.3405	110.0000	110.0000	110.0000	110.0000	109.9994
P^{39}	109.909	109.855	109.363	109.9243	110.0000	110.0000	110.0000	110.0000	109.9974
P^{40}	512.348	510.984	511.261	468.1694	511.2800	510.9800	511.2794	511.2800	511.2800
Fuel cost*10^5 $	1.22323	1.21788	1.22044	1.216492	1.214101	1.214630	1.214035	1.217419	1.21412
CPU(s)	35.851	15.6988	22.5841	63.5147	123.54	25.6984	45.2698	35.3652	15.26

Outputs (MW)	EDA/DE	CSOMA	DE/BBO	DHS	*B*-GWO	FPA	nPSO	nDE	i*h*PSODE
P^1	112.8544	110.8016	110.7998	110.7998	110.8001	72.4810	110.7918	110.7451	110.5643
P^2	111.7795	110.8068	110.7998	110.7998	110.8218	103.0314	110.7123	110.4518	103.1231
P^3	97.4059	97.4007	97.3999	97.3999	97.3934	83.2726	97.3933	97.1254	83.4532
P^4	179.7274	179.7333	179.7331	179.7331	179.7318	182.3106	179.7333	179.5241	117.3106
P^5	87.9306	87.8180	87.9576	87.7999	92.6135	76.1669	87.7333	87.4875	95.2234
P^6	139.9880	139.9997	140.0000	140.0000	140	126.1346	140	140	125.4556
P^7	259.7736	259. 6010	259.5997	259.5997	259.6065	258.8452	259.5342	259.1542	258.8567
P^8	284.6280	284.6000	284.5997	284.5997	284.6012	297.163	284.5234	284.3571	296.8765
P^9	284.7539	284.6005	284.5997	284.5997	284.5997	290.889	284.5432	284.7415	289.889
P^{10}	130.0291	130.0003	130.0000	130.0000	130	274.8232	129	129.875	129.8232
P^{11}	168.7908	168.7999	168.7998	94.0000	168.7992	356.9806	166	168.7752	356.9806
P^{12}	168.8084	168.7999	94.0000	94.0000	168.7930	124.4054	124	124	124.3345
P^{13}	214.7191	214.7599	214.7598	214.7598	214.7614	493.3764	214.2143	215.7452	213.3554
P^{14}	394.2888	394.2794	394.2794	394.2794	394.2808	344.9029	380.2322	394.2234	344.9444
P^{15}	304.5196	304.5196	394.2794	394.2794	394.2809	372.3864	392.2222	394.2132	372.3454
P^{16}	394.2952	394. 2794	394.2794	394.2794	304.5237	345.4624	392.2455	392.3433	392.4654
P^{17}	489.2905	489.2796	489.2794	489.2794	489.2802	422.6378	485.2344	484.2344	482.6456
P^{18}	489.2779	489.2795	489.2794	489.2794	489.2846	434.4065	462.2432	489.3494	462.4675
P^{19}	511.2861	511.2794	511.2794	511.2794	511.2710	461.3107	511.2234	513.2233	510.3466
P^{20}	511.2792	511.2796	511.2794	511.2794	511.2782	434.3828	511.5432	473.2124	510.7644
P^{21}	523.2858	523.2797	523.2794	523.2794	523.2798	545.2846	523.2343	524.2223	529.2456
P^{22}	523.2899	523.2798	523.2794	523.2794	523.2807	490.3572	522.2123	524.2232	523.3567
P^{23}	523.2783	523.2801	523.2794	523.2794	523.2798	506.0639	523.2794	523.2412	506.0677
P^{24}	523.3199	523.2795	523.2794	523.2794	523.2795	467.310	489.2794	472.2243	480.3107
P^{25}	523.2791	523.2797	523.2794	523.2794	523.2811	488.1203	524.2794	523.3422	522.1776
P^{26}	523.3076	523.2799	523.2794	523.2794	523.2766	486.9019	514.5434	523.4543	520.9666
P^{27}	10.0021	10.0004	10.0000	10.0000	10	16.8002	13	12	13.8654
P^{28}	10.0054	10.0004	10.0000	10.0000	10	39.3475	34	35.05	39.3456
P^{29}	10.0061	10.0003	10.0000	10.0000	10	23.6359	10	10	23.6123
P^{30}	88.8932	92.7158	97.0000	87.7999	87.8779	86.3295	87.7455	97	86.3345
P^{31}	189.9975	189.9998	190.0000	190.0000	190	165.9924	190	190	165.9567
P^{32}	189.9919	189.9998	190.0000	190.0000	190	174.5707	190	190	174.5544
P^{33}	189.9825	189.9998	190.0000	190.0000	190	184.0570	190	190	189.0123
P^{34}	164.9291	164.8014	164.7998	164.7998	164.8134	193.6668	163.7543	163.1232	164.3455
P^{35}	164.8031	164.8015	200.0000	200.0000	164.8051	191.6152	195	195	198.2541
P^{36}	164.9387	164.8051	200.0000	194.3978	164.8105	196.1763	194.3345	194	193.8754
P^{37}	109.9974	109.9998	100.0000	110.0000	110	90.0101	110	100	109.2514
P^{38}	109.9856	109.9998	110.0000	110.0000	110	37.5421	110	109	107.1548
P^{39}	109.9995	109.9996	110.0000	110.0000	110	89.423	110	110	89.4234
P^{40}	511.2813	511.2797	511.2794	511.2794	511.2946	471.4405	482.2143	481.3421	481.6645
Fuel cost*10^5 $	1.214187	1.214147	1.214208	1.214035	121415.09	1.210745	1.2121	1.2117	**1.2104**
CPU(s)	12.999	39.2514	19.2884	22.364	20.258	48.365	14.1002	12.3374	**11.5841**

Table 8 Simulation results for UTS-4

P_D	h	Power outputs (MW)	GA	PSO	FPA	WOA	GA-WOA	nPSO	nDE	i*h*PSODE
400 (MW)	43.55981	P^1	102.617	102.612	102.4468	102.4887	102.5355	101.6157	101.0747	101.6155
		P^2	153.825	153.809	153.8341	153.8043	153.7200	151.2972	151.8354	151.2563
		P^3	151.011	150.991	151.1321	151.1278	151.1046	147.0871	147.0899	147.1282
		P_L	7.41324	7.41173	7.4126	7.4208	7.4182	7.4184	7.4122	7.4102
		Fuel cost ($)	20840.1	20838.3	20838.1	20838	20836	20839.4512	20838.1544	20837.2154
		Emission (kg)	200.256	200.221	200.2238	200.2316	200.1748	200.2210	200.2204	200.2193
		Total cost($)	29563.2	29559.9	29559.81	29560	29556	29559.6854	29558.6500	**29557.3589**
		CPU(s)	0.282	0.235	0.175	0.297	0.783327	0.212	0.189	**0.115**
500(MW)	44.07915	P1	128.997	128.984	128.8074	128.5434	128.5344	125.8785	125.2001	125.997
		P2	192.683	192.645	192.5906	192.6543	192.7451	187.0074	186.6855	185.5519
		P3	190.11	190.063	190.2958	190.2876	190.2784	187.1141	188.1144	188.4511
		P_L	11.6964	11.6919	11.6938	11.6954	11.6811	11.6964	11.7854	11.6964
		Fuel cost ($)	25499.4	25495	25494.7	25495.744	25494.568	25493.4145	25495.1522	25499.4
		Emission (kg)	311.273	311.15	311.155	311.165	311.1485	311.1152	311.1093	311.1021
		Total cost($)	39220.1	39210.2	39210.15	39219.210	39218.541	39209.125	39210.168	**39208.181**
		CPU(s)	0.172	0.156	0.126	0.183	0.177	0.162	0.154	**0.122**

Table 9 Simulation results for UTS-5

Outputs(MW)	MODE	NSGAII	PDE	SPEA-2	GSA	ABC_PSO	EMOCA	FPA	nPSO	nDE	i*h*PSODE
P^1	54.9487	51.9515	54.9853	52.9761	54.9992	55	55	53.188	52.9215	52.1522	52.9755
P^2	74.5821	67.2584	79.3803	72.813	79.9586	80	80	79.975	72.8131	72.8131	72.8541
P^3	79.4294	73.6879	83.9842	78.1128	79.4341	81.14	83.5594	78.105	78.1122	78.2511	78.1185
P^4	80.6875	91.3554	86.5942	83.6088	85.0000	84.216	84.6031	97.119	83.6785	83.5412	83.6085
P^5	136.8551	134.0522	144.4386	137.2432	142.1063	138.3377	146.5632	152.74	137.2455	137.1542	137.1522
P^6	172.6393	174.9504	165.7756	172.9188	166.5670	167.5086	169.2481	163.08	172.9145	172.8214	172.9128
P^7	283.8233	289.4350	283.2122	287.2023	292.8749	296.8338	300	258.61	287.2154	287.2514	287.2154
P^8	316.3407	314.0556	312.7709	326.4023	313.2387	311.5824	317.3496	302.22	326.4153	324.4985	326.4469
P^9	448.5923	455.6978	440.1135	448.8814	441.1775	420.3363	412.9183	433.21	448.8824	449.8785	448.8258
P^{10}	436.4287	431.8054	432.6783	423.9025	428.6306	449.1598	434.3133	466.07	423.9045	425.9125	423.9025
Fuel cost*10^5 ($)	1.13484	1.13539	1.1351	1.1352	1.1349	1.1342	1.13445	1.1337	1.1351	1.1349	**1.1335**
Emission (lb)	4124.9	4130.2	4111.4	4109.1	4111.4	4120.1	4113.98	3997.7	3998.7452	3996.0222	**3994.2514**
Loses (MW)	84.33	84.25	83.9	84.1	83.9869	84.1736	83.56	84.3	84.103	84.274	**84.012**
h	8.2877	8.2398	8.9955	8.23567	8.54322	8.96788	8.99078	8.99967	8. 22284	8.21887	8. 21255
CPU(s)	3.82	6.02	4.23	7.53	3.68	3.65	2.90	2.23	3.56	3.25	**2.22**

Table 10 Simulation results for UTS-6

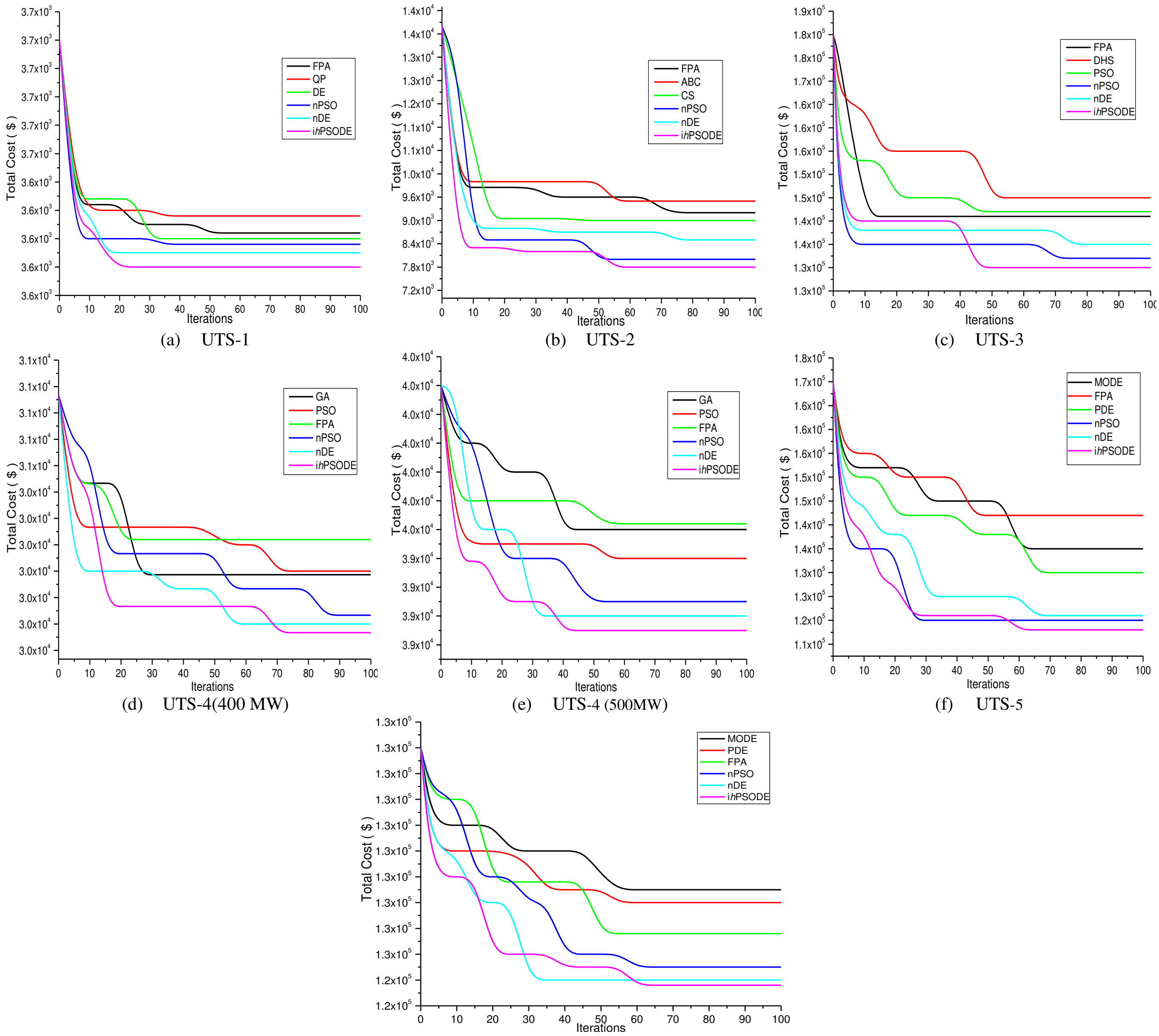
Outputs (MW)	MODE	PDE	NSGA-II	SPEA-2	GSA	MABC/D/Cat	WOA	MABC/D/Log	FPA	GA-WOA	nPSO	nDE	i*h*PSODE
P^1	113.5295	112.1549	113.8685	113.9694	113.9989	110.7998	103.5252	110.7998	43.405	103.42601	113.1254	113.2581	113.5214
P^2	114	113.9431	113.6381	114	113.9896	110.7998	102.8142	110.7998	113.95	102.9235	113.4851	113.7854	113.9158
P^3	120	120	120	119.8719	119.9995	97.3999	93.399	97.3999	105.86	92.01605	119.9999	119.1524	119.5562
P^4	179.8015	180.2647	180.7887	179.9284	179.7857	174.5504	181.2391	174.5486	169.65	183.828	179.8547	179.3584	179.7211
P^5	96.7716	97	97	97	97	87.7999	88.6654	97	96.659	84.01458	97.0000	97.2514	97.0000
P^6	139.2760	140	140	139.2721	139.0128	105.3999	123.93	105.3999	139.02	123.9824	139.1145	139.1542	139.1452
P^7	300	299.8829	300	300	299.9885	259.5996	258.589	259.5996	273.28	259.9376	299.7896	299.7854	299.9885
P^8	298.9193	300	299.0084	298.2706	300	284.5996	282.982	284.5996	285.17	294.4696	300.0000	300.0000	300.0000
P^9	290.7737	289.8915	288.8890	290.5228	296.2025	284.5996	288.395	284.5996	241.96	289.805	296.2845	296.7541	296.2514
P^{10}	130.9025	130.5725	131.6132	131.4832	130.3850	130	221.951	130	131.26	248.272	130.358	130.3581	130.3854
P^{11}	244.7349	244.1003	246.5128	244.6704	245.4775	318.1921	188.5985	318.2129	312.13	198.1655	245.4859	245.7854	245.1526
P^{12}	317.8218	318.2840	318.8748	317.2003	318.2101	243.5996	128.7994	243.5996	362.58	123.955	318.2052	318.7845	318.2055
P^{13}	395.3846	394.7833	395.7224	394.7357	394.6257	394.2793	451.763	394.2793	346.24	441.354	394.6025	394.6985	394.6569
P^{14}	394.4692	394.2187	394.1369	394.6223	395.2016	394.2793	394.28	394.2793	306.06	398.9032	395.2154	395.1458	395.2115
P^{15}	305.8104	305.9616	305.5781	304.7271	306.0014	394.2793	354.081	394.2793	358.78	382.0418	306.0015	306.0008	306.0078
P^{16}	394.8229	394.1321	394.6968	394.7289	395.1005	394.2793	334.28	394.2793	260.68	342.2716	395.1007	395.1485	395.1005
P^{17}	487.9872	489.3040	489.4234	487.9857	489.2569	399.5195	429.2798	399.5195	415.19	469.7822	489.2567	489.2569	489.3588
P^{18}	489.1751	489.6419	488.2701	488.5321	488.7598	399.5195	489.128	399.5195	423.94	445.1378	489.1752	489.1751	489.1158
P^{19}	500.5265	499.9835	500.8	501.1683	499.2320	506.1985	451.2306	506.1716	549.12	462.328	500.5268	500.7854	500.5562
P^{20}	457.0072	455.4160	455.2006	456.4324	455.2821	506.1985	461.88	506.2206	496.7	482.812	457.0021	457.3598	457.0152
P^{21}	434.6068	435.2845	434.6639	434.7887	433.4520	514.1472	523.2803	514.1105	539.17	513.198	434.6845	434.2541	434.6085
P^{22}	434.5310	433.7311	434.15	434.3937	433.8125	514.1455	485.28	514.1472	546.46	499.9768	434.5584	434.2541	434.5521
P^{23}	444.6732	446.2496	445.8385	445.0772	445.5136	514.5237	521.458	514.5664	540.06	528.032	444.6485	444.6732	444.6895
P^{24}	452.0332	451.8828	450.7509	451.8970	452.0547	514.5386	493.311	514.4868	514.5	469.3765	452.4851	445.0732	452.4851
P^{25}	492.7831	493.2259	491.2745	492.3946	492.8864	433.5196	523.28	433.5195	453.46	499.1242	492.7877	492.7131	492.2514
P^{26}	436.3347	434.7492	436.3418	436.9926	433.3695	433.5195	481.2801	433.5196	517.31	499.8282	436.3377	438.3847	436.3589
P^{27}	10	11.8064	11.2457	10.7784	10.0026	10	12.33	10	14.881	15.7668	10.0000	10.0000	10.0000
P^{28}	10.3901	10.7536	10	10.2955	10.0246	10	21	10	18.79	18.06882	10.3901	10.3971	10.3988
P^{29}	12.3149	10.3053	12.0714	13.7018	10.0125	10	14.17	10	26.611	12.98864	12.3158	12.1349	12.3188
P^{30}	96.9050	97	97	96.2431	96.9125	97	88.7611	87.8042	59.58	88.05685	96.1548	96.9108	96.9888
P^{31}	189.7727	189.4826	190.0000	189.4826	190.0000	189.9689	190	159.733	183.48	162.1605	189.7154	189.7727	184.7757
P^{32}	174.2324	175.3065	174.7971	174.2163	175	159.733	190	159.7331	183.39	188.2118	173.7524	174.2898	174.2154
P^{33}	190	190	189.2845	190	189.0181	159.733	190	159.733	189.02	181.089	186.4856	189.4589	189.1544
P^{34}	199.6506	200	200	200	200	200	161.834	200	198.73	174.1796	199.3355	199.7895	199.4851
P^{35}	199.8662	200	199.9138	200	200	200	163.891	200	198.77	195.0134	199.1548	199.1485	199.1593
P^{36}	200	200	199.5066	200	199.9978	200	169.8054	200	182.23	194.884	198.6348	197.8952	199.0541
P^{37}	110	109.9412	108.3061	110	109.9969	89.1141	109.389	89.1141	39.673	98.4301	110.0000	110.0000	110.0000
P^{38}	109.9454	109.8823	110	109.6912	109.0126	89.1141	110.12	89.1141	81.596	109.5671	109.1648	109.9458	109.5841
P^{39}	108.1786	108.9686	109.7899	108.5560	109.4560	89.1141	110.91	89.1141	42.96	108.8450	108.6586	108.1214	108.1658
P^{40}	422.0628	421.3778	421.3778	421.5609	421.8521	421.9987	511.28	506.1879	537.17	505.7928	421.1528	421.7845	421.8855
Total cost*10^5 ($)	1.2579	1.2573	1.2583	1.2581	1.2578	1.2449090	1.23644	1.24491161	1.23170	1.22862	1.2365	1.2236	**1.2225**
Emission*10^5(ton)	2.1119	2.1177	2.1095	2.1110	2.1093	2.5656026	2.1324	2.56560267	2.0846	2.06850	2.1094	2.1025	**2.0985**
CPU(s)	5.39	6.15	7.32	8.57	5.69	6.25	5.02	5.65	4.92	5.51	4.55	4.25	**3.25**

As specified in these tables the best cost created by- (i) presented nPSO for UTS-1, UTS-2 and UTS-3 of ELD problems are 3619.88 ($/hr), 8368.8401($/hr) and 1.2121×10^5($/hr) respectively and for UTS-4 (400 and 500 MW load demand), UTS-5 and UTS-6 of CEED problems are 29559.6854 ($/hr), 39209.125($/hr), 1.1351×10^5 ($/hr) and 1.2365×10^5 ($/hr) separately. (ii) proposed nDE for UTS-1, UTS-2 and UTS-3 of ELD problems are 3619.55($/hr), 8359.2141($/hr), and 1.2117×10^5($/hr) respectively and for UTS-4 (400 and 500 MW load demand), UTS-5 and UTS-6 of CEED problems are 29558.6500 ($/hr), 39210.168 ($/hr), 1.1349×10^5($/hr) and 1.2236×10^5($/hr) correspondingly. (iii) proposed i*h*PSODE for UTS-1, UTS-2, and UTS-3 of ELD problems are 3619.45($/hr), 8356.1545($/hr) and

1.2104×10^5($/hr) respectively and for UTS-4 (400 and 500 MW load demand), UTS-5 and UTS-6 of CEED problems are 29557.3589 ($/hr), 39208.181($/hr), 1.1335×10^5($/hr) and 1.2225×10^5($/hr) individually.

According to the reported cost results, the proposed nPSO, nDE, and i*h*PSODE algorithms have the lowest fuel cost and emission when compared to other compared algorithms for all unit test systems. Furthermore, CPU average times for each unit test system are noted in the associated tables, demonstrating that the proposed algorithms produce better solutions in less time than others. As a result, the proposed algorithms outperform and outlast other compared algorithms in terms of reducing total cost in the shortest amount of time. This indicates that the presented algorithm has higher reliability/robustness, stability and convergence when compared to other algorithms.

The convergence curves of presented and other algorithms are plotted in Fig. 7(a-g) for UTS-1, UTS-2, UTS-3, UTS-4, UTS-5 and UTS-6 in terms of total cost versus iterations. These figures show that presented algorithms (nPSO, nDE, and i*h*PSODE) has better convergence performance.



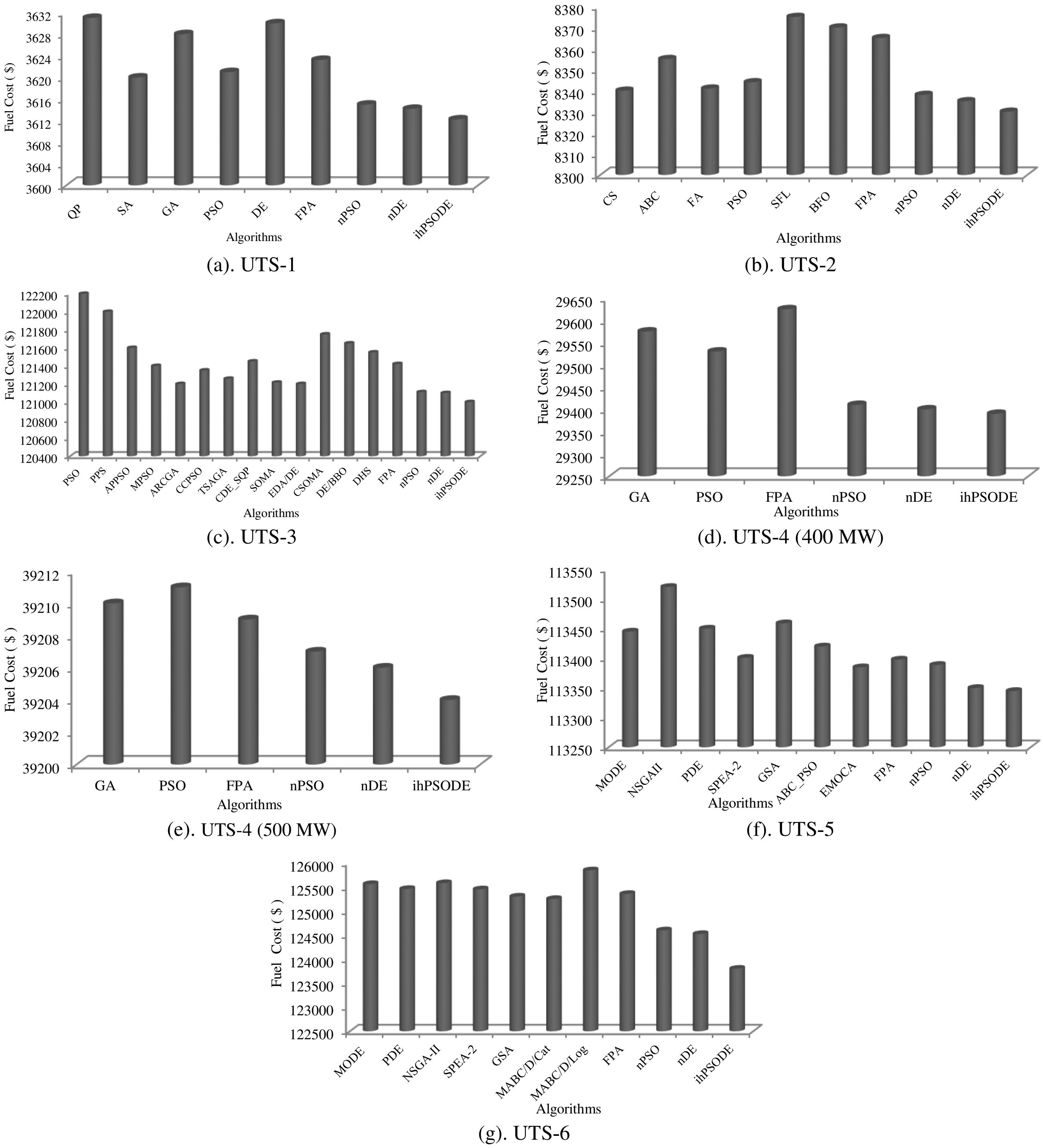
(a) UTS-1 (b) UTS-2 (c) UTS-3

(d) UTS-4(400 MW) (e) UTS-4 (500MW) (f) UTS-5

(g) UTS-6

Fig. 7(a-g) Cost convergence characteristic for different test systems

Moreover, fuel cost variations for all test systems presented in Fig. 8 (a-g). It shows and confirmed effectiveness of the proposed nPSO, nDE and i*h*PSODE for decreasing the fuel cost. Also, these figures demonstrate that the supremacy of the presented algorithm in attaining minimum fuel cost compared to others with different demands. Therefore, presented algorithms are economically competent. At large, it can be stating that (from the all above result investigation) presented algorithms (nPSO, nDE and i*h*PSODE) are performing better and/or similar with others. Still, between three presented algorithms hybrid algorithm i.e. i*h*PSODE have greater capability.



(a). UTS-1

(b). UTS-2

(c). UTS-3

(d). UTS-4 (400 MW)

(e). UTS-4 (500 MW)

(f). UTS-5

(g). UTS-6

Fig. 8(a-g) Fuel cost variations for different unit test systems

4.2 Complexity analysis

In this section some complexity examination of the presented algorithms is specified as below.

i). time complexity

Presented i*h*PSODE has the following time complexity (according to the steps).

a). np-population initialization needs $O(np.D)$ time.

b). evaluation and sorting (as per fitness function values) population wants $O(t_{max} \times np)$ time.

c). partition of population (pop_1 and pop_2) requires $O(t_{max} \times np)$ time.

d). calculation of pop_1 (by nDE) and pop_2 (by nPSO) takes $O(t_{max} \times \frac{np}{2} \times \frac{np}{2}) = O(t_{max} \times \frac{np^2}{4})$ time.

e). population integration and execution of algorithm involves $O(t_{max} \times np \times np)$= $O(t_{max} \times np^2)$ time.

So, for maximum number of iterations the total time complexity of ihPSODE is-

$$O(np.D)+ O(t_{max} \times np) + O(t_{max} \times np)+ O(t_{max} \times \frac{np^2}{4})+ O(t_{max} \times np^2)= O(t_{max} \times np^2 \times D)$$

ii). space complexity

The space complexity is the maximum volume of space which is used by presented ihPSODE algorithm. Thus, the total space complexity of proposed ihPSODE algorithm is $O(max(np, np, np, \frac{np^2}{4}, np^2) \times D)$=$O(np^2 \times D)$.

5. Conclusion with future perspectives

In this paper, to promote the performance of PSO and DE algorithm, a novel PSO (namely nPSO), novel DE (called nDE) and their innovative hybrid (titled ihPSODE) is presented for solving combined economic and emission dispatch (CEED) problems. Presented nPSO has a new acceleration coefficient, inertia weight and position improve equation (to alleviate the stagnation) as well as nDE has a new mutation approach and crossover rate (to prevent premature convergence). After population evaluation (in ihPSODE) best half member has been recognized and nPSO employed (which enhanced local and global search capacity) then nDE (which ensures bring solutions with higher quality), in each iteration process. In addition, because of suitable implementation of nPSO and nDE, particle can learn not only from the globally based individuals, but also from the best individuals of each problem in ihPSODE. Altogether, quality of 'memorizing (by nPSO)' and 'diversity maintaining (by nDE)' brands ihPSODE more robust. Likewise, related novel presented control parameters of nPSO and nDE makes extra support for the success of ihPSODE.

The presented algorithms (nPSO, nDE and ihPSODE) have been tested over 23 unconstrained benchmark functions then applied to solve two large scale power engineering optimization problem namely economic load dispatch (ELD) and combined economic emission dispatch (CEED) problem. These problem include 3 test systems (3, 6 and 40-unit test system) of ELD and 3 test systems (3, 10 and 40-unit test system) of CEED problem. The performance of presented methods compared with the classical DE and PSO with their existed variants and hybrids plus other state-of-the-art methods.

The simulation results prove that presented algorithms are more effective than or at least competitive to the compared algorithms in case of unconstrained benchmark functions. Moreover, presented algorithms are successfully used to solve ELD and CEED power system engineering optimization problems. The optimization results confirm that presented algorithms can achieve better solutions than other compared methods in case of power system engineering optimization problems. Therefore, presented algorithms are economically competent. All in all, it can be summarized that the proposed algorithms (nPSO, nDE and ihPSODE) can be seen as an effective algorithm to solve power system engineering optimization problems. Lastly, between three presented algorithms hybrid algorithm i.e. ihPSODE have greater capability.

Furthermore, the presented algorithms do have higher time complexity than some PSO, DE and hybrid variants. The matrix operation evaluation is the primary cause of the presented algorithms' time-consuming nature. This operation is repeated for each individual on each iteration, which increases the algorithm's running time in some extent. Besides, the presented algorithms may not be appropriate for all complex optimization problems.

Some novel parameters will be designed for the presented nDE, nPSO, and ihPSODE as part of our future work for finding more precise solutions and falling time complexity. Finally, this paper is expected to devote in a fruitful analysis i.e. complete mathematical convergence analysis of the presented algorithm which may done in the coming paper with inspecting how to advance the strength for multifaceted optimization problems.

Acknowledgments
Authors, heartfelt thanks to the Editor and the Reviewers for their constructive suggestions.

Declaration of competing interest
There are no conflicts of interest.

Data availability statement
Authors declare that all data analyzed or generated throughout this study are involved in the submitted paper. Similarly, availability of data and materials are cited in references of the paper. Additionally, the data of this study are available from the corresponding author upon reasonable request wich support the findings. No additional data archiving is required.

References

Abdelaziz AY, Ali ES, Abd Elazim SM (2016) Combined economic and emission dispatch solution using Flower Pollination Algorithm. Electrical Power and Energy Systems 80:264–274

Ajayi O, Heymann R (2021) Day-Ahead Combined Economic and Emission Dispatch with Spinning Reserve Consideration using Moth Swarm Algorithm for a Data Centre Load Heliyon, 7(9):e08054

Amjady N, Rad HN (2010) Solution of nonconvex and nonsmooth economic dispatch by a new adaptive real coded genetic algorithm. Expert System Applications 37(7):5239–5245

Ang KM, Lim WH, Isa NAM, Tiang SS, Wong CH (2020) A Constrained Multi-swarm Particle Swarm Optimization without Velocity for Constrained Optimization Problems. Expert Systems with Applications 140:1-23.

Basu M (2011) Economic environmental dispatch using multi-objective differential evolution, Applied Soft Computing 11:2845–2853

Ben GN (2020) An accelerated differential evolution algorithm with new operators for multi-damage detection in plate-like structures. Applied Mathematical Modelling 80:366-383.

Betar A, Awadallah MA, Krishan MM (2020). A non-convex economic load dispatch problem with valve loading effect using a hybrid grey wolf optimizer. Neural Computing and Applications 32:12127–12154

Bhattacharya A, Chattopadhyay PK (2010) Hybrid differential evolution with biogeography-based optimization for solution of economic load dispatch. IEEE Transaction Power System 25(4):1955–1964

Bibi H, Ahmad A, Aadil F, Kim M, Muhammad K (2020) A Solution to Combined Economic Emission Dispatch (CEED) problem using Grasshopper Optimization Algorithm (GOA). International Conference on Computational Science and Computational Intelligence 712-718

Chegini SN, Bagheri A, Najafi F (2018) A new hybrid PSO based on sine cosine algorithm and Levy flight for solving optimization problems. Applied Soft Computing 73:697-726

Chen CH, Yeh SN (2006) Particle swarm optimization for economic power dispatch with valve-point effects. In: IEEE PES transmission and distribution conference and exposition Latin America, Venezuela, pp 1–5

Chen Y, Li L, Xiao J, Yang Y, Liang J, Li T (2018) Particle swarm optimizer with crossover operation. Engineering Applications of Artificial Intelligence 70:59–169

Civicioglu P (2013) Backtracking Search Optimization Algorithm for numerical optimization problems. Applied Mathematics and Computation 219:8121–8144

Coelho LDS, Mariani VC (2006) Combining of chaotic differential evolution and quadratic programming for economic dispatch optimization with valve-point effect. IEEE Transaction Power System 21(2):989–996

Coelho LDS, Mariani VC (2010) An efficient cultural self-organizing migrating strategy for economic dispatch optimization with valve-point effect. Journal Energy Conversion and Management 51(12):2580–2587

Das KN, Parouha RP (2015) An ideal tri-population approach for unconstrained optimization and applications. Applied Mathematics and Computation 256:666–701

Dash J, Dam B, Swain R (2020) Design and implementation of sharp edge FIR filters using hybrid differential evolution particle swarm optimization. AEU - International Journal of Electronics and Communications 114:1-61

Davis L (1991) Handbook of Genetic Algorithms

Deb K (1995) Optimization for Engineering Design: Algorithms and Examples. Prentice-Hall of India, New Delhi

Devi AL, Krishna OV (2008) Combined economic and emission dispatch using evolutionary algorithms – a case study. ARPN Journal of Engineering and Applied Sciences 3(6):28–35

Edwin Selva Rex CR, Marsaline Beno M, Annrose J (2019) A Solution for Combined Economic and Emission Dispatch Problem using Hybrid Optimization Techniques. Journal of Electrical Engineering & Technology doi.org/10.1007/s42835-019-00192-z

Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm–A novel metaheuristic optimization method for solving constrained engineering optimization problems. Computers & Structures 110-111:151-166

Espitia HE, Sofrony, JI (2018) Statistical analysis for vortex particle swarm optimization. Applied Soft Computing 67:370–386

Famelis IT, Alexandridis A, Tsitouras C (2017) A highly accurate differential evolution–particle swarm optimization algorithm for the construction of initial value problem solvers. Engineering Optimization 50(8):1364–1379

Faramarzi A, Heidarinejad M, Stephens B. Mirjalili S (2019) Equilibrium optimizer: A novel optimization algorithm. Knowledge-Based Systems 191:1-34

Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Communications in Nonlinear Science and Numerical Simulation 17(12):4831-4845

Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60-68

Goudarzi A, Li Y, Xiang J (2020) A hybrid non-linear time-varying double-weighted particle swarm optimization for solving non-convex combined environmental economic dispatch problem. Applied Soft Computing 86:1-54

Gui L, Xia X, Yu F, Wu H, Wu R, Wei B, He G (2019) A multi-role based differential evolution. Swarm and Evolutionary Computation 50:1-15

Güvenç U, Sonmez Y, Duman S, Yorükeren N (2012) Combined economic and emission dispatch solution using gravitational search algorithm. Computers & Electrical Engineering 19(6):1754–1762

Hardiansyah (2013) A modified particle swarm optimization technique for economic load dispatch with valve-point effect. International Journal of Intelligent Systems and Applications 7:32–41

Hardiansyah, Junaidi, Yohannes MS (2013) An efficient simulated annealing algorithm for economic load dispatch problems. Telecommunication, Computing, Electronics and Control 11(1):37–46

Hassan BA, Rashid TA (2019a) Operational framework for recent advances in backtracking search optimisation algorithm: A systematic review and performance evaluation. Applied Mathematics and Computation https://doi.org/10.1016/j.amc.2019.124919

Hassan BA, Rashid TA (2019b) Datasets on statistical analysis and performance evaluation of backtracking search optimization algorithm compared with its counterpart algorithms. Data in brief 28:105046

Hassan MH, Kamel S, Salih SQ, Khurshaid T, Ebeed M (2021) Developing Chaotic Artificial Ecosystem-Based Optimization Algorithm for Combined Economic Emission Dispatch. In IEEE Access, 9:51146-51165, doi: 10.1109/ACCESS.2021.3066914

Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: Algorithm and applications. Future Generation Computer Systems 97:849-872

Hosseini SA, Hajipour A, Tavakoli H (2019) Design and optimization of a CMOS power amplifier using innovative fractional-order particle swarm optimization. Applied Soft Computing 85:1-10

Hu L, Hua W, Lei W, Xiantian Z (2020) A modified Boltzmann Annealing Differential Evolution algorithm for inversion of directional resistivity logging-while-drilling measurements. Journal of Petroleum Science and Engineering 180:1-10

Huang H, Jiang L, Yu X, Xie D (2018) Hypercube-Based Crowding Differential Evolution with Neighborhood Mutation for Multimodal Optimization. International Journal of Swarm Intelligence Research 9(2):15–27

Isiet M, Gadala M (2019) Self-adapting control parameters in particle swarm optimization. Applied Soft Computing 83:1-24

Jiang S, Zhang C, Wu W, Chen S (2019) Combined Economic and Emission Dispatch Problem of Wind-Thermal Power System Using Gravitational Particle Swarm Optimization Algorithm. Mathematical Problems in Engineering 2019:1–19.

Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm. Journal of Global Optimization 39(3):459–471

Kennedy J, Eberhart RC (1995) Particle Swarm Optimization. In: Proceeding of IEEE International Conference on Neural Networks, pp 1942–1948

Khajeh A, Ghasemi MR, Arab HG (2019) Modified particle swarm optimization with novel population initialization. Journal of Information and Optimization Sciences 40:(6) 1167-1179

Khatsu S, Srivastava A, Das DK (2020) Solving Combined Economic Emission Dispatch for Microgrid using Time Varying Phasor Particle Swarm Optimization, 6th International Conference on Advanced Computing and Communication Systems (ICACCS) 411-415

Kumar C, Alwarsamy T (2012) Solution of economic dispatch problem using differential evolution algorithm. International Journal of Soft Computing and Engineering 1(6):236–241

Lanlan K, Ruey SC, Wenliang C, Yeh C (2020) Non-inertial opposition-based particle swarm optimization and its theoretical analysis for deep learning applications. Applied Soft Computing 88:1-10

Li S, Gu Q, Gong W, Ning B (2020) An enhanced adaptive differential evolution algorithm for parameter extraction of photovoltaic models. Energy Conversion and Management 205: 1-16

Liu ZG, Ji XH, Yang Y (2019) Hierarchical Differential Evolution Algorithm Combined with Multi-Cross Operation. Expert Systems with Applications 130:276-292

Mahdi FP, Vasant P, Abdullah-Al-Wadud M, Kallimani V, Watada J (2019) Quantum-behaved bat algorithm for many-objective combined economic emission dispatch problem using cubic criterion function. Neural Computing and Applications 31:5857–5869

Mahmoodabadi MJ, Mottaghi ZS, Bagheri A (2014) High Exploration Particle Swarm Optimization. Journal of Information Science 273:101-111

Mansor MH et al. (2018) A hybrid optimization technique for solving economic dispatch problem. Journal of Physics: Conf. Series 1049:1-7

Manteaw ED, Odero NA (2012) Combined economic and emission dispatch solution using ABC_PSO hybrid algorithm with valve point loading effect. International Journal of Scientific and Research Publication 2(12):1–9

Mao B, Xie Z, Wang Y, Handroos H, Wu H (2018) A Hybrid Strategy of Differential Evolution and Modified Particle Swarm Optimization for Numerical Solution of a Parallel Manipulator. Mathematical Problems in Engineering 1–9

Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete and multi-objective problems. Neural Computing and Applications 27(4):1053-1073

Mirjalili S, Lewis A (2016) The Whale optimization algorithm. Advances in Engineering Software 95:51–67

Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Advances in Engineering Software 69:46-61

Park JB, Jeong YW, Shin JR, Lee KY (2010) An improved particle swarm optimization for nonconvex economic dispatch problems. IEEE Transaction Power System 25(1):156–166

Parouha RP, Das KN (2015) An Efficient Hybrid Technique for Numerical Optimization and Applications. Computers & Industrial Engineering 83:193–216

Parouha RP, Das KN (2016a) A Robust Memory Based Hybrid Differential Evolution for Continuous Optimization Problem. Knowledge-Based Systems 103:118-131

Parouha RP, Das KN, (2016b) An intelligent parallel hybrid algorithm for economic load dispatch problems with various practical constraints. Expert Systems with Applications 63:295–309

Parouha RP, Verma P (2021) An innovative hybrid algorithm for bound-unconstrained optimization problems and applications. Journal of Intelligent Manufacturing doi.org/10.1007/s10845-020-01691-x

Parouha RP; Verma P (2020) An innovative hybrid algorithm to solve nonconvex economic load dispatch problem with or without valve point effects. International Transactions on Electrical Energy Systems 34(1):1-67

Qiu X, Xu JX, Xu Y, Tan KC (2018) A New Differential Evolution Algorithm for Minimax Optimization in Robust Design, IEEE Transactions on Cybernetics 48(5):1355–1368.

Rao RV, Savsani VJ, Vakharia DP (2011) Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Computer-Aided Design 43(3):303–315

Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) A Gravitational Search Algorithm. Information Sciences 179(13):2232–2248

Rashid HA, Mohammed KA, Firas MFF (2020) A New Enhancement on PSO Algorithm for Combined Economic-Emission Load Dispatch Issues. international journal of intelligent engineering & system 13(1):77-85

Rezaie H, Kazemi-Rahbar MH, Vahidi B, Rastegar H (2018) Solution of combined economic and emission dispatch problem using a novel chaotic improved harmony search algorithm. Journal of Computational Design and Engineering 6(3):447-467

Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Applied Soft Computation 13(5):2592–2612

Sakthivel VP, Suman M, Sathya PD (2021) Combined economic and emission power dispatch problems through multi-objective squirrel search algorithm. Applied Soft Computing. 100, doi:10.1016/j.asoc.2020.106950

Secui DC (2015) A new modified artificial bee colony algorithm for the economic dispatch problem. Energy Conversion and Management 89:43–62

Serapião ABS (2009) Fundamentos de otimizaçãopor inteligência de enxames: uma visão geral. Revista SBA Controle and Automação 20(3):271–304

Serapiao ABS (2013) Cuckoo search for solving economic dispatch load problem. Intelligent Control Automat 4:385–390

Seyedmahmoudian M. et al. (2015) Simulation and hardware implementation of new maximum power point tracking technique for partially shaded PV system using hybrid DEPSO method. Transactions on Sustainable Energy 6(3):850-862

Simpson AR, Dandy GC, Murphy LJ (1994) Genetic algorithms compared to other techniques for pipe optimization. Journal of Water Resources Planning and Management 20:423–443

Storn R, Price K (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization 11:341–359

Subbaraj P, Rengaraj R, Salivahanan S (2011) Enhancement of Self-adaptive real coded genetic algorithm using Taguchi method for economic dispatch problem. Applied Soft Computing 11(1):83–92

Tanabe R, Fukunaga A (2013) Success-history based parameter adaptation for Differential Evolution. In IEEE Congress on Evolutionary Computation 71–78

Tang B, Xiang K, Pang M (2018) An integrated particle swarm optimization approach hybridizing a new self-adaptive particle swarm optimization with a modified differential evolution. Neural Computing and Applications 1-35

Tang B, Zhu Z, Luo J (2016) Hybridizing Particle Swarm Optimization and Differential Evolution for the Mobile Robot Global Path Planning. International Journal of Advanced Robotic Systems 13(3):1-17

Too J, Abdullah, Saad NM (2019) Hybrid Binary Particle Swarm Optimization Differential Evolution-Based AR Feature Selection for EMG Signals Classification. Axioms 8(3):1-17

Verma P, Parouha RP (2021) Non-convex Dynamic Economic Dispatch Using an Innovative Hybrid Algorithm. Journal of Electrical Engineering & Technology doi.org/10.1007/s42835-021-00926-y

Wang L, Li LP (2013) An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems. International Journal of Electric Power Energy System 44:832–843
Wang Y, Li B, Weise T (2010) Estimation of distribution and differential evolution cooperation for large scale economic load dispatch optimization of power systems. Information Sciences 180(12):2405–2420
Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Transactions on Evolutionary Computation 1(1):67–82
Xia X, Gui L, He G, Xie C, Wei B, Xing Y, Tang Y (2018) A hybrid optimizer based on firefly algorithm and particle swarm optimization algorithm. Journal of Computational Science 26:488–500
Xiong H, Qiu B, Liu J (2020) An Improved Multi-swarm Particle Swarm Optimizer for Optimizing the Electric Field Distribution of Multichannel Transcranial Magnetic Stimulation. Artificial Intelligence In Medicine 104:1-14
Yan B, Zhao Z, Zhou Y, Yuan W, Li J, Wu J, Cheng D (2017) A Particle Swarm Optimization Algorithm with Random Learning Mechanism and Levy Flight for Optimization of Atomic Clusters. Computer Physics Communication 219:79-86
Yang X, Li J, Peng X (2019) An improved differential evolution algorithm for learning high-fidelity quantum controls. Science Bulletin 64(19):1402-1408
Yang XS, Deb S (2009) Cuckoo Search via Lévy flights. In proceedings of World Congress on Nature & Biologically Inspired Computing, Coimbatore, India, pp 210-214
Yu H, Tan Y, Zeng J, Sun C, Jin Y (2018) Surrogate-assisted hierarchical particle swarm optimization. Information Sciences 454-455:59–72
Zhang H, Li X (2018) Enhanced differential evolution with modified parent selection technique for numerical optimization. International Journal of Computational Science and Engineering 17(1): 98-108
Zhang J, Sanderson C (2009) JADE: Adaptive Differential Evolution with optional external archive. IEEE Transactions on Evolutionary Computation 13(5):945–958
Zhang R, Zhou J, Mo L, Ouyang S, Liao X (2013) Economic environmental dispatch using an enhanced multi-objective cultural algorithm. Electric Power Systems Research 99:18–29