Applied Intelligence (2022) 52:10448–10492
https://doi.org/10.1007/s10489-021-02803-7

A systematic overview of developments in differential evolution and particle swarm optimization with their advanced suggestion

Raghav Prasad Parouha[1] · Pooja Verma[1]

Accepted: 26 August 2021 / Published online: 14 January 2022

Abstract

An efficient survey of numerous traditional metaheuristic algorithms (MAs) has been investigated in this paper. Among successful MAs, differential evolution (DE) and particle swarm optimization (PSO) have been widely recognized to solve complex optimization problems and received much attention from many researchers. Therefore, DE and PSO are chosen in the present study and an extensive survey of their recent-past variants with hybrids has been inspected again. After this an advanced DE (ADE) and PSO (APSO) with their hybrid (AHDEPSO) are proposed for unconstrained optimization problems. In ADE a novel mutation strategy, crossover probability and random nature selection scheme (to avoid premature convergence) as well as in APSO novel gradually varying parameters (to avoid stagnation) are introduced. Hence, ADE and APSO affords different convergence characteristics to the solution space. Also to balance between exploration and exploitation, in AHDEPSO population is divided (multi-population approach) and merged with others in a pre-defined way. Thus, AHDEPSO achieves better solutions and it is expected to obtain productive solutions with an increasing success rate at each cycle. To verify the performance of all 3 proposed algorithms i.e. ADE, APSO, and AHDEPSO applied to solve 23 basic, 30 IEEE CEC 2017 unconstrained benchmark functions and 3 real-world problems. There are several numerical and graphical analyses have been done to verify the performances of the proposed algorithms robustly. Additionally, statistical and comparative analysis confirms the superiority of the proposed algorithms among traditional DE and PSO with their recent variants and hybrids as well as over many state-of-the-art algorithms. Finally, between 3 proposed algorithms the best one i.e. AHDEPSO is recommended to solve unconstrained optimization problems.

Keywords Metaheuristic algorithm · Particle swarm optimization · Differential evolution · Hybrid algorithm · Unconstrained optimization

1 Introduction

Many optimization methods have been developed to solve multifaceted optimization problems. However, conventional optimization methods have certain inherent drawbacks like high computational complexity, local optimal stagnation, and derivation of the search space [1]. Also, it is difficult to find the optimal solution in the solving process. Presently, to overcome the drawbacks of conventional optimization methods, a bunch of optimization methods known as metaheuristics algorithms (MAs) have been introduced. The mechanisms to develop MAs are simple as well as based on natural practices and become yet very efficient in solving complex global optimization problems. According to the mechanical differences, the MAs can be categorized into four groups as follows- swarm intelligence algorithms, evolutionary algorithms, physics-based algorithms, and human behavior based algorithms. Some recent past instances of these algorithms are depicted in Fig. 1 and reviewed briefly as follows.

Genetic algorithm (GA), the foundation for many evolutionary algorithms, was defined by David Goldberg in 1989 [2]. It is taken as the search algorithms build on the mechanics of natural genetics and selection, simulated by the natural process of evolution like selection, mutation, and crossover. Later in 1995, inspired by the flocking of bird Eberhart and Kennedy proposed particle swarm optimization (PSO) [3]. PSO has acquired immense popularity amongst researchers

✉ Raghav Prasad Parouha
rparouha@gmail.com

Pooja Verma
pooja.v0903@gmail.com

[1] Department of Mathematics, Indira Gandhi National Tribal University, Amarkantak, MP, India

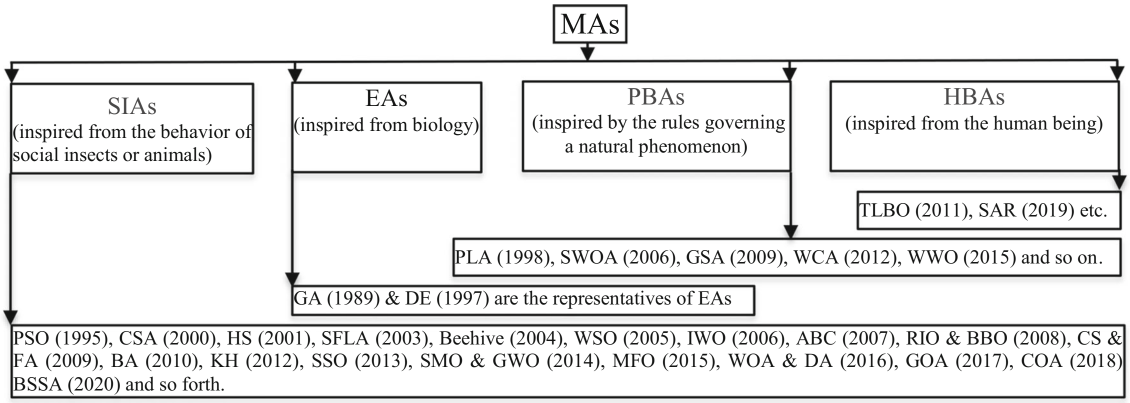

MAs
SIAs
(inspired from the behavior of social insects or animals)
EAs
(inspired from biology)
PBAs
(inspired by the rules governing a natural phenomenon)
HBAs
(inspired from the human being)
TLBO (2011), SAR (2019) etc.
PLA (1998), SWOA (2006), GSA (2009), WCA (2012), WWO (2015) and so on.
GA (1989) & DE (1997) are the representatives of EAs
PSO (1995), CSA (2000), HS (2001), SFLA (2003), Beehive (2004), WSO (2005), IWO (2006), ABC (2007), RIO & BBO (2008), CS & FA (2009), BA (2010), KH (2012), SSO (2013), SMO & GWO (2014), MFO (2015), WOA & DA (2016), GOA (2017), COA (2018) BSSA (2020) and so forth.

Fig. 1 Traditional metaheuristics

because of its simplicity and effectiveness in plenty of scientific and industrial applications. In PSO particles get updated by themselves with the initial velocity, also all the individuals learn from the others and adapt themselves by trying to emulate the behavior of the fittest individuals. On the other hand, differential evolution (DE) is one of the most popular algorithms in evolutionary metaheuristics algorithm discovered by Storn and Price in 1997 [4]. DE work on two phases; initialization and evolution. In first phase, population is generated randomly and in the second phase, the same generated population undergo mutation, crossover and selection processes. Due some advantages like easy implementation and strong global search capability, DE has come out to be a well-liked choice among researchers for solving various optimization problems in different sectors. Then in 1998, photosynthetic learning algorithm (PLA) came into existence by Murase and Wadano [5]. It utilizes the rules governing the conversion of carbon molecules from one substance to another in the Benson-Calvin cycle (comprises of three phases i.e. carhoxylation, reduction & regeneration of the acceptor) and Photorespiration reactions. Further in 2000, de Castro and Zuben developed clonal selection algorithm (CSA) [6] which is related to artificial immune system that described a general learning strategy. The principal immune aspects of CSA taken into consider were: maintenance of the memory cells, selection and cloning of the most stimulated cells, death of non-stimulated cells, affinity maturation, and re-selection of the clones.

With the help of analogy of the music performance process, Geem et al. [7] devised harmony search (HS) algorithm. In this algorithm a new vector is produced after taking all existing vectors instead of considering only two (parents) as in the genetic algorithm, and HS does not need the setting of initial values of decision variables and it performed well in many combinatorial or continuous problems. In 2003, Eusuff and Lansey proposed shuffled frog leaping algorithm (SFLA) [8]. It mimics the cooperative behavior of frogs displayed while they search for food in a swamp and uses memetic evolution in the form of infection of ideas from one individual to another in a local search. Also, a shuffling technique permits for the exchange of information between local searches to move toward a global optimum. Again, in 2004 BeeHive algorithm (Beehive) devised by Wedde et al. [9], it is inspired by the communicative and evaluative procedures of honey bees. In this algorithm, bee agents pass through network regions called foraging zones and their information on network state is delivered for updating the local routing tables. Further, Pinto et al. in 2005 proposed wasp swarm optimization (WSO) [10]. It is bidding algorithm where the wasps take the role of bidders trying to acquire finite resources and had been designed specifically with reference to logistics system optimization. Then, Du et al. used theories of small-world phenomenon and constructed SWOA small-world optimization algorithm (SWOA) [11] algorithm in 2006 where local shortcuts search and random long-range search operator are employed to solve optimization problems. Later in 2006, Mehrabian and Lucas proposed a general-purpose optimization algorithm that is inspired by weed colonization named as invasive weed optimization (IWO) [12]. In this algorithm, invasive weed reproduces quickly by building seeds and increase their population. Moreover, their behavior becomes different with time as the colony becomes dense leaving lesser opportunity of life for the ones with lesser fitness. Again in 2007, Karaboga and Basturk presented artificial bee colony (ABC) [13] which contains three groups of bees: employed bees, onlookers and scouts with three steps cycle: employed bees send through the food sources then their nectar amounts are measured; onlookers select the food sources after sharing the information of employed bees then determining the nectar amount of the foods and lastly determining the scout bees then direct them onto possible food sources.

Moving in the year of 2008, Havens et al. invented roach infestation optimization (RIO) [14] whose main aim is to look into the effect of adapting the PSO with the social behavior of cockroaches. In this algorithm cockroaches makes an effort to search the darkest place, fitness of a cockroach is proportional to the darkness of its location. They communicated with each other with a predefined probability. At a certain point of time cockroaches become hungry and leave the darkness to search for food. Another population-based metaheuristic algorithm, biogeography based optimization (BBO) was proposed by Simon in 2008 [15]. It is inspired by geographical distribution of biological organisms and enclosed two main operator's migration (support to update each individual by sharing the characteristic of individuals) and mutation (helps to enhance diversity and the changes for a good solution). Later in 2009, Yang & Deb devised cuckoo search (CS) algorithm [16] in which the brood parasitic behavior of cuckoo species simulated along with the lévy flight action of birds and fruit flies. There are three steps in CS, (i) one egg laid at a time, and cuckoo leaves its egg in an any random nest, (ii)high-quality eggs nests having can survive, (iii) the number of host nests is constant, and the egg can be noticed by the host bird with a probability (p_a) belongs to [0, 1]. Further, in the area of swarm intelligence Yang discovered firefly algorithm (FA) in 2009 [17], inspired by the behavior of flashing features of fireflies. The two main purposes of such flashes are attracting mating partners and warning against predators. The flashing can be formulated as an objective function which needs to be optimized. Also in same year, Rashedi et al. proposed gravitational search algorithm (GSA) [18]. It is based on the law of Gravity in which agents are considered as objects and their performance is measured by their masses. Each agent has four properties (determined by fitness function value) position, inertial mass, active gravitational mass, and passive gravitational. Then in 2010, bat algorithm (BA) was proposed by Yang [19]. Its main motivation is the echolocation behavior of bats, where bats can find their prey and differentiate different kinds of insects even in complete darkness.

Further, Rao et al. introduced teaching-learning based optimization (TLBO) in the year 2011 [20]. It is training based model in which population comprises of two phase i.e. Teacher phase (learning from the teacher) and learner phase (learning by the interaction between learners). Then in the year 2012, water cycle algorithm (WCA) was discovered by Eskandar et al. [21]. Their elementary concepts are inspired from nature and rely on the observation of water cycle process. In WCA, raindrop is considering as initial population, best raindrop (best individual) chosen as sea, good raindrops taken as river and left raindrop chosen as streams which flow to the rivers and sea. Again in 2012, Gandomi and Alavi devised an algorithm named as krill herd (KH) [22], which is based on the herding behavior of krill. Its three main actions are: movement influence by the presence of other individuals, foraging activity and random diffusion, that's determined by the time-dependent position of an individuals. Later in 2013, social spider optimization (SSO) was introduced by Cuevas et al. [23], which is inspired by simulation of cooperative behavior of social-spiders. In this algorithm, individual spiders (solutions) are simulated by the biological laws of the cooperative colony. Also, males and females are two search agents (spiders) in SSO, depending on which each individual is conducted by different evolutionary operators. Again, spider monkey optimization (SMO) was discovered by Bansal et al. in 2014 [24], which is inspired by intelligent foraging behavior of fission–fusion social structure based animals. It has two control parameters i.e. global leader limit and Local leader limit that helps local and global leaders to take appropriate decisions. In same year 2014, Mirjalili et al. proposed grey wolf optimizer (GWO) [25], it mimics the leadership hierarchy and hunting approach of grey wolves. It employed four types of grey wolves (Alpha, beta, delta, and omega) for simulating the leadership hierarchy and three steps (searching for prey, encircling prey, and attacking prey) for hunting. Later in 2015, inspired by the shallow water wave theory, Zheng devised an algorithm called as water wave optimization (WWO) [26] in which three operators; propagation (helps to make high fitness waves search small areas and low fitness waves explore), refraction (helps waves to escape search stagnation) and breaking (enables an intensive search around a promising area.) have been implemented depending upon phenomena of water flow. Then, Mirjalili proposed moth- flame optimization (MFO) [27] in same year 2015, inspired by the navigation technique (transverse orientation) of moths. In MFO, a mathematical model of spiral flying path of moths around artificial lights (flames) is developed in which moths are considered as candidate solutions and problem's variables are the position of moths in the space.

In the year 2016, Mirjalili and Lewisa proposed whale optimization algorithm (WOA) [28]. This metaheuristic inspired by the social behavior of humpback whales and it consists of three steps; encircling prey, bubble-net attacking method and search for prey. Later in 2016, a swarm intelligence based technique, dragonfly algorithm (DA) proposed by Mirjalili [29]. It is inspired by static and dynamic swarming behaviours of dragonflies and it has two phases: exploration and exploitation, which modeled while dragonflies search food, navigate and avoid enemies in a swarm. Further in 2017, grasshopper optimization algorithm (GOA) was modeled mathematically by Saremi et al. [30], it mimics the behavior of grasshopper swarms and designed to simulate repulsion force (help to explore the search space) and attraction forces (help to exploit promising regions) in between the grasshoppers. Later in 2018, Pierezan and Dos inspired by the *Canis latrans* species and introduced a population based metaheuristic named as coyote optimization algorithm (COA) [31]. It especially focuses on the social structure and experiences exchange by the coyotes and has two parameters: number of packs and the number of coyotes per pack. In recent year 2019, search and rescue optimization (SAR) was developed by Shabani et al. [32], inspired by search and rescue operations of the humans being and it consists of two phases: social phase (selects the search direction based on the position of clue) and individual phase (searches around the best clue). Then early in 2020, based on sense of smell and movement mechanism of bear, Marzbali presented bear smell search algorithm (BSSA) [33]. It has two mechanisms: olfactory bulb mechanism that connected to the brain, employed mesh mechanism to move the next position.

The primary advantage of these algorithms is their use of the "trial-and-error" principle in searching for solutions. Thus, these algorithms were successfully applied to solve global optimization problems. Among successful MAs, DE and PSO have been widely recognized to solve complex optimization problems and received much attention of many researchers [34–39]. Therefore, DE and PSO are chosen in the present study. However, some shortcomings of DE and PSO cause limitation in applying them in complex optimization environments. Therefore, avoiding shortcomings of these algorithms many variants and their hybridization are introduced in the literature.

Although, as claimed on No-Free-Lunch(NFL) theorem [40], a large number of MAs are introduced in the literature but they couldn't able to solve variety of problems. Moreover, a method may have suitable results for some problems, but not for others. Thus, there is a need to introduce some effective algorithms to solve a wider range of problems. This is the motivation of this study to present novel variants of DE and PSO with their hybridization.

Moreover, after extensive vigorous literature review on different variants of DE and PSO with their hybridization (section 2 related work), following points are analyzed and motivated from them.

(i). In DE mutation and crossover strategy with their associate control parameters utilized to produce the global best solution which is beneficial for improving the convergence behavior. Therefore, in DE most appropriate strategies and their associated parameter values are considered a vital research study.

(ii). The performance of PSO greatly depends on its parameters like acceleration coefficients and inertia weight which guide particles to the optimum and balancing diversity respectively. Hence, many researchers have tried to modify the control parameter of PSO to achieve better accuracy and higher speed.

(iii). Hybrid algorithms (by combining the advantages of different algorithms) have aroused interest of the researchers due to its effectiveness for complex optimization problems. Since DE and PSO have complementary properties therefore their hybrids have gained prominence recently. To best of our knowledge, finding ways to combine DE and PSO is still an open problem.

Inspired/motivated by the above observations and our survey of literature, the following plans of action (major contributions) have been outlined for solving complex unconstrained optimization problems.

(i). Developed an advanced differential evolution (ADE) where novel mutation strategy and crossover probability along with slightly changed selection scheme are familiarized.

(ii). Suggested an advanced particle swarm optimization (APSO) which consists of novel gradually varying (decreasing and/or increasing) parameters.

(iii). Designed an advanced hybrid algorithm (AHDEPSO) by hybridizing advanced DE and PSO as well as based on multi-population approach.

The rest of this paper is organized as follows: Section 2 reviews the related work on different and hybrid variants of DE and PSO. Section 3 describes the proposed algorithms. The proposed algorithms are verified on a wide set of benchmark functions and real-world engineering applications in Section 4. Section 5 concludes this study with future works.

2 Related work

During past decade, development of several powerful MAs to solve high-dimensional optimization problems has become a popular study area. Among successful MAs, DE and PSO have been widely recognized to solve complex optimization problems and received much attention of many researchers. The basics of original DE and PSO are presented as follows.

2.1 Differential evolution (DE)

DE is an evolutionary approach proposed by Storn and Price [4]. The key idea behind DE is to use vector differences for perturbing vector population. In *D*-dimensional search space it initializes a population randomly of *np* individuals within the lower and upper boundaries (x_l, x_u). After initialization, the DE is conducted by the three main operations defined as follows.

Mutation: for each target vector $(x_{i,j}^t)$ a mutant vector $(v_{i,j}^t)$ at iteration t is generated as follows.

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

where $r_1, r_2, r_3 \in \{1, 2, ..., np\}$ are randomly chosen integers with $r_1 \neq r_2 \neq, r_3 \neq i$, and F denotes the scaling vector that employed to control the amplification of differential variation.

Crossover: a trial vector $\left(u_{i,j}^{t}\right)$ produce by combining target ($x_{i,j}^{t}$) and mutant ($v_{i,j}^{t}$) vector as follows.

$$u_{i,j}^{t}=\begin{cases} v_{i,j}^{t}; \text{if } rand\leq C_r \\ x_{i,j}^{t}\ \ ; \text{otherwise}\end{cases} \tag{2}$$

where $i\in[1,np]$, $j\in[1,D]$ and $rand\in \mathrm{U}[0,1]$ (uniformly distributed random number between 0 & 1), $C_r\in[0,1]$ denotes crossover rate that controls how many components are inherited from the mutant vector.

$$\text{Selection}: \qquad x_{i,j}^{t+1}=\begin{cases} u_{i,j}^{t}; \text{if } f\left(u_{i,j}^{t}\right)\leq f\left(x_{i,j}^{t}\right) \\ x_{i,j}^{t}; \text{Otherwise}\end{cases} \tag{3}$$

Again mutation, crossover and selection operators allowed for offspring repeatedly up-to predefined stopped criteria.

2.2 Particle swarm optimization (PSO)

PSO was originally proposed by Eberhart and Kennedy in 1995 [3]. It simulates social and/or group behaviors in animals, insects and humans. In classical PSO, swarm flies in the D-dimensional search space to seek for global optimum. Each of the i^{th} swarm particles has its own position ($x_i=(x_{i,\ 1},x_{i,\ 2},\ldots,x_{i,\ D})$) and velocity $v_i=(v_{i,\ 1},v_{i,\ 2},\ldots,v_{i,\ D})$. During the evolution, each particle tracks its individual best $pbest_{i\text{-}}=(pbest_{i,\ 1},pbest_{i,\ 2},\ldots,pbest_{i,\ D})$ and global best $gbest_{j\text{-}}=(gbest_{1},gbest_{2},\ldots,gbest_{D})$, velocity and position of the i^{th} particle are updated as follows at each iteration.

$$v_{i,j}^{t+1}=wv_{i,j}^{t}+c_1r_1\left(pbest_{i,j}-x_{i,j}^{t}\right)+c_2r_2\left(gbest_{j}-x_{i,j}^{t}\right) \tag{4}$$

$$x_{i,j}^{t+1}=x_{i,j}^{t}+v_{i,j}^{t+1} \tag{5}$$

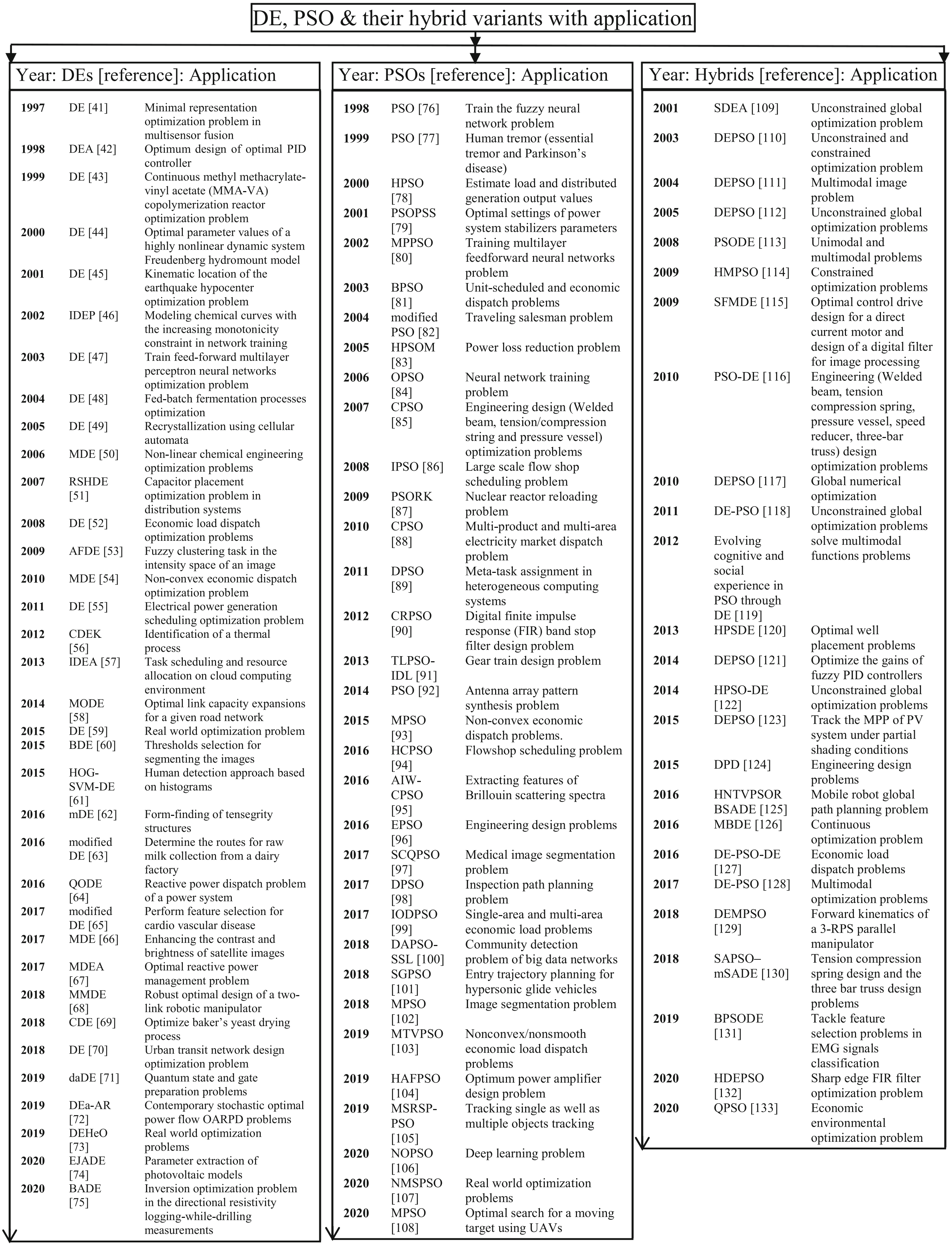
where t is iteration index, $v_{i,j}^{t}$ is velocity of i^{th} particle in D-dimension at t^{th} iteration, c_1 is cognitive acceleration coefficient, c_2 is social acceleration coefficient, r_1, r_2 are two uniform random numbers in the range between [0, 1] and w is the inertia weight.

Due to some deficiencies of DE (low local exploitation ability and loss of diversity) and PSO (easily get stuck at a local optimal solution region and low convergence rate) cause limitation in applying them in typical optimization problems. Therefore, avoiding shortcomings of these algorithms many variants and their hybridization are introduced in the literature. Some recent past variants of DE and PSO as well as their hybrids are reviewed effectively as follows and illustrated in Fig. 2.

2.3 DE variants

DE has remarkable performance and become a powerful optimizer in the field of real world problems. However, it has few issues such as convergence rate and local exploitation ability. In order to overcome its shortcomings, lots of robust and effective DE has been designed in the literature. A detailed survey of DE variants can be found in [34, 35]. Moreover, a briefed survey of significant DE variants has been summarized as follows.

As DE is effective in solving in difficult search problem, Joshi and Sanderson in 1997 [41] applied DE approach to solve the minimal representation problem in multisensor fusion. In 1998, Cheng and Hwang developed DE algorithm (DEA) [42], it represents the continuous parameters by floating-point numbers rather than by binary bit-strings. Moreover, it is applied to the design of optimal PID controller. Later in 1999, Lee et al. proposed modified DE (MDE) [43], it employed a local search to improve computational efficiency as well as modified heuristic constraints to lessen the search space size. And it is applied to the continuous methyl methacrylate-vinyl acetate (MMA-VA) copolymerization reactor problem. Then in 2000, Kyprianou et al. [44] employed DE to identify the optimal parameter values of a highly nonlinear dynamic system Freudenberg hydromount model. Further, Ruzek and Kvasnicka have pointed out the practical applicability of the DE in the problem of the kinematic location of the earthquake hypocenter in 2001 [45]. It is found that the sensitivity of the DE essentially retains the favorable properties over most of the admissible range. Then Chen et al. proposed improved differential evolution (IDEP) [46] in 2002. It employed flip operation (to adjust the prior-knowledge-violating networks), also Levenberg–Marquardt descent and random perturbation strategy are adopted to speed up the convergence of DE and prevent from being locally trapped. Also, it is applied in the modeling chemical curves with the increasing monotonicity constraint in network training. Later in 2003 Ilonen et al. [47] used DE to analyzed the train feedforward multilayer perceptron neural networks for reaching its optima. Because DE has no major restrictions apply to the error function as well as on the regularization methods. Proceeding in 2004, Kapadi and Gudi [48] analysed the computational aspect of DE with augmented Lagrangian including the dynamic penalty method. Then it applied on fed-batch fermentation processes involving multiple feeds as it provides a path to obtain the feasible optimal environment in fermentation broth and avoid inhibition. In 2005, Rane et al. [49] worked on the process of recrystallization using cellular automata (CA). Where DE is employed to search for the value of nucleation rate, providing an acceptable matching between the theoretical and experimentally observed values of fraction-recrystallized. Further, Babu and Angira in 2006 developed Modified DE (MDE) [50]. It accomplished by taking single



DE, PSO & their hybrid variants with application

Year: DEs [reference]: Application

1997	DE [41]	Minimal representation optimization problem in multisensor fusion
1998	DEA [42]	Optimum design of optimal PID controller
1999	DE [43]	Continuous methyl methacrylate-vinyl acetate (MMA-VA) copolymerization reactor optimization problem
2000	DE [44]	Optimal parameter values of a highly nonlinear dynamic system Freudenberg hydromount model
2001	DE [45]	Kinematic location of the earthquake hypocenter optimization problem
2002	IDEP [46]	Modeling chemical curves with the increasing monotonicity constraint in network training
2003	DE [47]	Train feed-forward multilayer perceptron neural networks optimization problem
2004	DE [48]	Fed-batch fermentation processes optimization
2005	DE [49]	Recrystallization using cellular automata
2006	MDE [50]	Non-linear chemical engineering optimization problems
2007	RSHDE [51]	Capacitor placement optimization problem in distribution systems
2008	DE [52]	Economic load dispatch optimization problems
2009	AFDE [53]	Fuzzy clustering task in the intensity space of an image
2010	MDE [54]	Non-convex economic dispatch optimization problem
2011	DE [55]	Electrical power generation scheduling optimization problem
2012	CDEK [56]	Identification of a thermal process
2013	IDEA [57]	Task scheduling and resource allocation on cloud computing environment
2014	MODE [58]	Optimal link capacity expansions for a given road network
2015	DE [59]	Real world optimization problem
2015	BDE [60]	Thresholds selection for segmenting the images
2015	HOG-SVM-DE [61]	Human detection approach based on histograms
2016	mDE [62]	Form-finding of tensegrity structures
2016	modified DE [63]	Determine the routes for raw milk collection from a dairy factory
2016	QODE [64]	Reactive power dispatch problem of a power system
2017	modified DE [65]	Perform feature selection for cardio vascular disease
2017	MDE [66]	Enhancing the contrast and brightness of satellite images
2017	MDEA [67]	Optimal reactive power management problem
2018	MMDE [68]	Robust optimal design of a two-link robotic manipulator
2018	CDE [69]	Optimize baker's yeast drying process
2018	DE [70]	Urban transit network design optimization problem
2019	daDE [71]	Quantum state and gate preparation problems
2019	DEa-AR [72]	Contemporary stochastic optimal power flow OARPD problems
2019	DEHeO [73]	Real world optimization problems
2020	EJADE [74]	Parameter extraction of photovoltaic models
2020	BADE [75]	Inversion optimization problem in the directional resistivity logging-while-drilling measurements

Year: PSOs [reference]: Application

1998	PSO [76]	Train the fuzzy neural network problem
1999	PSO [77]	Human tremor (essential tremor and Parkinson's disease)
2000	HPSO [78]	Estimate load and distributed generation output values
2001	PSOPSS [79]	Optimal settings of power system stabilizers parameters
2002	MPPSO [80]	Training multilayer feedforward neural networks problem
2003	BPSO [81]	Unit-scheduled and economic dispatch problems
2004	modified PSO [82]	Traveling salesman problem
2005	HPSOM [83]	Power loss reduction problem
2006	OPSO [84]	Neural network training problem
2007	CPSO [85]	Engineering design (Welded beam, tension/compression string and pressure vessel) optimization problems
2008	IPSO [86]	Large scale flow shop scheduling problem
2009	PSORK [87]	Nuclear reactor reloading problem
2010	CPSO [88]	Multi-product and multi-area electricity market dispatch problem
2011	DPSO [89]	Meta-task assignment in heterogeneous computing systems
2012	CRPSO [90]	Digital finite impulse response (FIR) band stop filter design problem
2013	TLPSO-IDL [91]	Gear train design problem
2014	PSO [92]	Antenna array pattern synthesis problem
2015	MPSO [93]	Non-convex economic dispatch problems.
2016	HCPSO [94]	Flowshop scheduling problem
2016	AIW-CPSO [95]	Extracting features of Brillouin scattering spectra
2016	EPSO [96]	Engineering design problems
2017	SCQPSO [97]	Medical image segmentation problem
2017	DPSO [98]	Inspection path planning problem
2017	IODPSO [99]	Single-area and multi-area economic load problems
2018	DAPSO-SSL [100]	Community detection problem of big data networks
2018	SGPSO [101]	Entry trajectory planning for hypersonic glide vehicles
2018	MPSO [102]	Image segmentation problem
2019	MTVPSO [103]	Nonconvex/nonsmooth economic load dispatch problems
2019	HAFPSO [104]	Optimum power amplifier design problem
2019	MSRSP-PSO [105]	Tracking single as well as multiple objects tracking
2020	NOPSO [106]	Deep learning problem
2020	NMSPSO [107]	Real world optimization problems
2020	MPSO [108]	Optimal search for a moving target using UAVs

Year: Hybrids [reference]: Application

2001	SDEA [109]	Unconstrained global optimization problem
2003	DEPSO [110]	Unconstrained and constrained optimization problem
2004	DEPSO [111]	Multimodal image problem
2005	DEPSO [112]	Unconstrained global optimization problems
2008	PSODE [113]	Unimodal and multimodal problems
2009	HMPSO [114]	Constrained optimization problems
2009	SFMDE [115]	Optimal control drive design for a direct current motor and design of a digital filter for image processing
2010	PSO-DE [116]	Engineering (Welded beam, tension compression spring, pressure vessel, speed reducer, three-bar truss) design optimization problems
2010	DEPSO [117]	Global numerical optimization
2011	DE-PSO [118]	Unconstrained global optimization problems
2012	Evolving cognitive and social experience in PSO through DE [119]	solve multimodal functions problems
2013	HPSDE [120]	Optimal well placement problems
2014	DEPSO [121]	Optimize the gains of fuzzy PID controllers
2014	HPSO-DE [122]	Unconstrained global optimization problems
2015	DEPSO [123]	Track the MPP of PV system under partial shading conditions
2015	DPD [124]	Engineering design problems
2016	HNTVPSOR BSADE [125]	Mobile robot global path planning problem
2016	MBDE [126]	Continuous optimization problem
2016	DE-PSO-DE [127]	Economic load dispatch problems
2017	DE-PSO [128]	Multimodal optimization problems
2018	DEMPSO [129]	Forward kinematics of a 3-RPS parallel manipulator
2018	SAPSO–mSADE [130]	Tension compression spring design and the three bar truss design problems
2019	BPSODE [131]	Tackle feature selection problems in EMG signals classification
2020	HDEPSO [132]	Sharp edge FIR filter optimization problem
2020	QPSO [133]	Economic environmental optimization problem

Fig. 2 DE, PSO & their hybrid variants with application

array in MDE as compared to two arrays in traditional DE, which reduces the memory and computational efforts. It is applied to solve the non-linear chemical engineering problems. Next year in 2007, Chang et al. proposed robust searching hybrid DE (RSHDE) [51]. It comprises of two schemes; multi-direction search and search space reduction scheme to enhance the search ability in the initial stages. RSHDE is used to solve the capacitor placement problem in distribution systems.

In 2008, Noman and Iba [52] have investigated the potential of the DE for solving economic load dispatch (ELD) problems in power systems. Where DE enhances the way of satisfying the power balance constraint and other boundary constrained by using a reflection mechanism. Further, Das and Konar presented automatic fuzzy clustering DE (AFDE) in 2009 [53]. It incorporates the two novel parameter tuning strategies to escape from stagnation and/or premature convergence. Moreover, it is applied to the fuzzy clustering task in the intensity space of an image. Later in the year 2010, Amjady and Sharifzadeh invented modified DE (MDE) [54]. This framework generated by owning a new mutation operator and selection mechanism which is inspired from GA, PSO and simulated annealing (SA). It is applied on non-convex economic dispatch problem considering valve loading effect. Then in 2011 Uyar et al. [55] proposed a novel way of employing DE to short-term electrical power generation scheduling problem. This problem is divided into two sub problem, where DE is applied with binary decision variable in a way that it lower the cost scheduling of power generators, satisfying some operational constraints. Proceeding in year 2012, Santos et al. [56] designed a new chaotic DE optimization approach based on Ikeda map (CDEK) to tune the crossover rate and mutation factor. It is applied to the identification of a thermal process. Later in 2013, Tsai et al. presented improved DE algorithm (IDEA) [57] based on the cost and time models on cloud computing environment. It combined the Taguchi method (to exploit the better individuals on micro space) and DE (has powerful global exploration capability). Moreover, IDEA used to optimize the task scheduling and resource allocation on cloud computing environment. Further, Baskan and Ceylan proposed Modified DE (MODE) [58] in 2014. It developed mutation strategy consideration rate (MSCR) and local search operator that enhanced the convergence rate of DE without being trapped in bad local optimum. This algorithm deals with determining the optimal link capacity expansions for a given road network. Guo and Yang [59] developed DE utilizing eigenvector-based crossover operator in 2015. It introduced a rotationally invariant crossover (based on eigenvectors of covariance matrix) and a new parameter P (eigenvector ratio) to control the ratio between the binomial & eigenvector-based crossover and to preserve the population diversity respectively. Moreover, it is used to apply in real world optimization problem. Ayala et al. invented beta DE (BDE) in 2015 [60]. It applies beta probability distribution in tuning F and CR parameters as it is flexible for modelling data. BDE is employed to select the thresholds for segmenting the images. Further in 2015, Chen et al. [61] employed DE in human detection approach based on histograms of oriented gradients (HOG) feature and termed as HOG-SVM-DE. Here DE is applied instead of scanning the detection windows in sliding fashion so as to achieve fast and accurate detection.

Further in 2016, Do et al. devised a modified DE (mDE) [62]. In this framework, best individual based mutation and elitist selection techniques instigated with modified scale and crossover factor to escalate exploitation ability and/or convergence speed of DE. Then it is applied for form-finding of tensegrity structures. In the same year Sethanan and Pitakaso proposed modified DE algorithms [63], where two additional steps included i.e. reincarnation and survival process in order to improve the solution quality. This algorithm is then used to determine the routes for raw milk collection from a dairy factory. Then, Basu developed quasi-oppositional DE (QODE) [64] in 2016. It adds quasi-oppositional based learning (QOBL) for population initialization and also for generation jumping. Moreover, QODE is used to solve reactive power dispatch problem of a power system. Proceeding in 2017, Vivekanandan and Iyengar proposed modified DE [65], where mutation strategy (DE/rand/2-wt/exp) is employed. Also, modified-DE-based feature selection is adapted to perform feature selection for cardio vascular disease. Further, Suresh and Lal invented Modified DE (MDE) [66] in 2017, it excesses DE for exploration phase whereas cuckoo search (CS) for exploitation phase as it ensured an increase in the convergence rate and to avoiding premature convergence. And MDE is employed for enhancing the contrast and brightness of satellite images. Later Sakr et al. proposed modified DE algorithm (MDEA) [67] in 2017, it comprises of self-adaptive scaling factor that dynamically adopts global and local searches to eliminate local optima trapping. It is implemented for solving optimal reactive power management (ORPM) problem.

Then in 2018, Qiu et al. designed minimax DE (MMDE) [68] where a novel bottom-boosting mechanism is introduced to maintain the reliability and identified the promising solutions. Also it applied partial-regeneration strategy and mutation operator (DE/current/1) to provide in-depth exploration over solution space. Then, it is applied to the robust optimal design of a two-link robotic manipulator. Continuing in same year, Yuzgec and Eser proposed another DE variant named as Chaotic based DE (CDE) [69]. To maintain the diversity in the initial population it includes four chaotic systems like as Lorenz, Rossler, Chua and Mackey-Glass functions for selection of the candidates from population in the mutation, crossover operations. Also it is used for the optimization of baker's yeast drying process. Again in 2018, Buba and Lee [70] proposed DE approach to optimize the urban transit network design problem (UTNDP). Identical point mutation and uniform route crossover with 0/1 crossover masks are used in this algorithm to increase the diversity and noisy random vectors. Early in 2019, a new DE variant direction averaged DE (daDE) is proposed by Yang et al. [71]. It created a modified mutation rule which utilizing the information of the current and the former individuals altogether. daDE is employed to solve the quantum state and gate preparation problems. Further in 2019, Awad et al. proposed a new DE algorithm named as DEa-AR [72]. It uses arithmetic recombination crossover and scaling factor (based on Laplace distribution). Additionally, an archive strategy is incorporated to consider the inferior individual's information to find new good solutions. DEa-AR is propounded to solve the contemporary stochastic optimal power flow OARPD problems. Later in 2019, DE with biological based mutation operator (DEHeO) is proposed by Prabha and Yadav [73]. In this algorithm, a mutation operator (hemostatic operator influenced by hemostasis biological phenomenon) is introduced which gives promising solutions and helps in enhancing the diversity in earlier stages thereby avoiding stagnation during later stages. It is used to apply in real world optimization problems. Recently in 2020, Li et al. developed enhanced adaptive DE (EJADE) [74]. In EJADE, a sorting mechanism is evolved to rationally assign CR values for each individual according to their fitness values. Moreover, a dynamic population reduction strategy is employed to speed up the convergence rate and maintain the diversity. It is used to apply in Photovoltaic systems optimization. Further, Hu et al. proposed Boltzmann annealing DE (BADE) [75] in 2020. It introduced an annealing strategy into the DE algorithm that allows exploring more searching space. Additionally, different strategies are employed at different stages of annealing (high and low temperature) to accelerate the convergence. Also, it is employed to optimize the inversion problem in the directional resistivity logging-while-drilling(DRLWD) measurements.

2.4 PSO variants

PSO has attracted the attention to solve many complex optimization problems due to its efficient search ability and simplicity. However, the main drawback of the PSO is that it may easily get stuck at a local optimal solution region. Therefore, accelerating the convergence speed and avoiding the local optimal solutions are the two critical issues in PSO. To overcome such issues various different modifications of the PSO has been proposed in the literature. A comprehensive survey on the variants of PSO can be found in [36, 37]. Furthermore, briefed reviews of some noteworthy PSO variants are summarized as follows.

Zhenya et al. proposed a modified version of PSO in 1998 [76], where each particle's uses the best current performance of its neighbours to replace the best previous one. Also to accelerate the search procedure, a not accumulative rate of change replaces to the accumulative one. And it is applied to train the fuzzy neural network problem. Then in 1999, Eberhart and Hu used PSO to analyze the human tremor (essential tremor and Parkinson's disease) [77]. It is used as to evolve a neural network weights and to evolve the network structure indirectly. Further, Naka and Fukuyama proposed hybrid PSO (HPSO) in 2000 [78]. It replaced the agent position of low evaluation values by high evaluation values using the selection procedure of evolutionary algorithm. Moreover, HPSO can estimate the load and distributed generation output values at each node considering nonlinear characteristics of distribution systems. Abido developed PSO based power system stabilizers (PSOPSS) in 2001 [79]. It incorporated an annealing procedure to make uniform and local search in the initial and later stages respectively. Additionally, feasibility check procedure also imposed in order to prevent the particles not to go outside the feasible search space. It is used as in order to search for optimal settings of PSS parameters. Later in 2002, Al-kazemi and Mohan developed multi-phase PSO (MPPSO) [80]. It evolves multiple groups of particles to change the direction of the search in the different phases that helps to explore the search space, enhancing population diversity, and preventing premature convergence. Also, MPPSO is employed for training multilayer feedforward neural networks (MFNN) problem. Further, Gaing proposed binary PSO (BPSO) in 2003 [81]. In BPSO the trajectories are changes with the probability in which a coordinate will take on binary value (0 or 1). Also, it combined with the lambda-iteration method for solving unit commitment (unit-scheduled and economic dispatch (ED)) problems. Proceeding in year 2004 [82], Pang et al. developed modified PSO, where position and velocity of the particles are presented by fuzzy matrices and operators of the classical PSO are redefined. It is used to solve traveling salesman problem (TSP). Then in 2005, Esmin et al. invented hybrid PSO with mutation (HPSOM) [83]. It incorporated mutation process of GA into PSO which helps to allow the search to escape from local optima and search in different zones of the search space. HPSOM is then applied to solve the power loss reduction problem. Meissner et al. proposed optimized PSO (OPSO) [84] in 2006. It comprises of subswarms and superswarm. The subswarms is used to find a solution to a given optimization problem, while the superswarm is employed to optimize their parameters. This algorithm is applied to resolve the neural network training problem. Later in year 2007, He and Wang invented co-evolutionary PSO (CPSO) [85]. It consists of multiple swarm (for searching the good solutions) and single swarm (for evolving the suitable penalty factors). Furthermore, CPSO is implement in parallel and applied to solve the welded beam design, tension/compression string design and pressure vessel design problems. Zhang et al. proposed improved PSO (IPSO) in 2008 [86], it combines the PSO with the two-point crossover (to redefine the model of the original PSO) and shift mutation operators (to search neighbourhood when a particle gets stagnate). Also, a fast fitness computation method based on matrix is devised to improve the algorithm speed. Moreover, it is used to solve the large scale flow shop scheduling problem. Further in 2009 [87], Meneses et al. presented PSO with random keys (PSORK). In which the position vector's information is decoded by the random keys (RK), so that positions need not to be rounded or truncated. And, it is employed to solve nuclear reactor reloading problem. Proceeding in year 2010, Azadani et al. proposed constrained PSO (CPSO) [88]. It initializes and updates the particles under uniform distribution for the faster convergence. Also, CPSO is applied to the multi-product and multi-area electricity market dispatch problem. In 2011, Kang and He proposed a novel discrete PSO (DPSO) [89]. It utilizes the characteristics of discrete variable to update the position. Moreover, variable neighbourhood descent algorithm and migration mechanisms are submerged in DPSO to speed up the convergence and maintain diversity. It is employed to solve meta-task assignment in heterogeneous computing systems. Later in 2012, Kar et al. proposed craziness based PSO (CRPSO) [90]. It introduced a craziness operator to make sure that the particle has a predefined craziness probability in order to maintain its diversity. CRPSO is designed to solve the digital finite impulse response (FIR) band stop filter design problem. Proceeding in year 2013, Lim and Isa developed the two-layer PSO with intelligent division of labor (TLPSO-IDL) [91]. It performes the evolutions sequentially on the current and memory swarm. A new learning mechanism is also proposed for current swarm to improve its exploration. Meanwhile an intelligent division of labor (IDL) module is invented for memory swarm to evolve adaptively by allocating different tasks to each swarm member. Additionally, an elitist-based perturbation (EBP) module is considered to prevent stagnation in local optima. Further, it is applied to solve the gear train design problem. Then in 2014, based on particle positions Zhang et al. proposed a novel parameter mechanism for classical PSO. [92]. It used the concept of overshoot and the peak time of a transition process, which provides new way to analyses the particle trajectories. Moreover, this algorithm is applied to solve the antenna array pattern synthesis problem. Basu developed modified PSO (MPSO) in 2015 [93]. The Gaussian random variables are introduced in velocity term to improve the search efficiency and obtaining the global optimum without impairing the convergence speed and the structure of PSO. Also, MPSO is used to apply in solve non-convex economic dispatch problems.

Then in 2016, Eddaly et al. [94] proposed hybrid combinatorial PSO (HCPSO). It introduced an iterative local search algorithm based on probabilistic perturbation, sequentially to PSO for enhancing the solutions quality. This algorithm is used for solving flowshop scheduling problem. Proceeding in same year, Zhang et al. [95] proposed adaptive inertia weight-chaos PSO (AIW-CPSO). It introduced an adaptive inertia weight so as to enhance the local optimization ability. Additionally, the logical self-mapping chaotic search is carried out in order to make the PSO to jump out of local optima. Further, it is employed for extracting the features of Brillouin scattering spectra. Also in 2016, Ngo et al. proposed extraordinariness PSO (EPSO) [96]. It contained extraordinary motion concept (movement strategy) for particles where they can proceed to a target that may be global best, local bests, or even the worst individual. EPSO is used to apply in engineering design problems. Later in 2017, Li et al. developed partitioned and cooperative quantum-behaved PSO (SCQPSO) [97]. It introduced auxiliary swarms and partitioned search space to enhance the population diversity. Also the cooperative theory is considered to improve the particle global search ability. Moreover, it is used to apply in medical image segmentation problem. Proceeding in same year, Phung et al. proposed discrete PSO (DPSO) [98]. It includes three techniques deterministic initialization, random mutation (to avoid the collapse situation and keep the balance between exploration and exploitation), and edge exchange (to compare each valid combination of the swapping mechanism for edges) to improve the accuracy of DPSO. Also, this algorithm solved the inspection path planning (IPP) problem. Qin et al. developed improved orthogonal design PSO (IODPSO) in 2017 [99]. It employed the tent chaotic map for the acceleration coefficients adaptation to improve global search capability. Further, IODPSO is used to solve the single-area and multi-area economic load problems.

Later in year 2018, direction aware PSO algorithm with sensitive swarm leader (DAPSO-SSL) is proposed by Mishra et al. [100]. It incorporated the basic human nature qualities like awareness, maturity, relationship and leadership to swarm leader and individual particles, so that particles can utilize previous knowledge of the best and current fittest individuals. DAPSO-SSL is applied to the community detection problem of big data networks. Proceeding to year 2018, Li et al. proposed stochastic gradient PSO (SGPSO) [101]. It combines stochastic gradient with the randomness of particle swarm search to overcome the problems of premature convergence and poor accuracy of standard PSO. It is used to generate a feasible solution of entry trajectory planning for hypersonic glide vehicles. Then Tian and Shi developed modified PSO (MPSO) in 2018 [102]. It utilized logistic map to distribute the particle uniformly so as to improve to initial population quality. Sigmoid like inertia weight and wavelet mutation is considered to achieve better swarm diversity. Additionally, an auxiliary velocity-position update mechanism is employed to global best particle in order to guarantee the convergence. It is applied to the image segmentation problem. In 2019 Parouha proposed modified time-varying PSO (MTVPSO) [103], It introduced a linearly decreased inertia weight and novel acceleration coefficients which improve the global search capability and diversity of the population. MTVPSO is used to apply in nonconvex/nonsmooth economic load dispatch problems. Further, Hosseini et al. [104] proposed hunter-Attack fractional-order PSO (HAFPSO) in 2019, where fractional-order derivatives and hunter-attack strategy are used to accelerate convergence and avoid stagnation respectively. It is used to apply in Optimum power amplifier design problem. Early in 2019, Dash and Patra developed mutation-based self-regulating and self-perception PSO (MSRSP-PSO) [105]. It incorporates self-regulation and self-perception behaviour of the global particle and dynamic adaption in learning. Also mutation operator is performed in global particle. This algorithm is suitably applied for tracking single as well as multiple objects tracking. Further, Non-inertial opposition-based PSO (NOPSO) [106] is proposed by Lanlan et al. in 2020. It has a non-inertial velocity update formula and opposition based learning strategy (to accelerate the convergence speed) also an adaptive elite mutation strategy (to avoid trapping into local optimum) is introduced. NOPSO is applied in the deep learning problem. Proceeding in same year, novel multi-swam PSO (NMSPSO) [107] is suggested by Xiong et al. It developed the three scheme i.e. 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). Also, it is used to solve the real world applications problems. Then, Motion-encoded PSO (MPSO) [108] is proposed early in 2020 by Phung and Ha. The motion-encoded approach preserves the important qualities of the swarm including the cognitive and social coherence so as to obtain better solution. Moreover, it is used to solve the problem of optimal search for a moving target using UAVs.

2.5 DE and PSO hybrid variants

Hybrid strategy is one of the main research directions to improve the performance of single algorithm. Different optimization algorithms have different search behaviors and advantages. Yet, to overcome individual shortcomings, such as premature convergence or stacking at some local optima, hybrid techniques are now more favored over their individual effort. Therefore, in order to enhance the performance of DE and PSO, lots of their hybrid algorithms are presented in the literature. A systematic survey on hybrid variants of DE and PSO can be found in [38, 39]. Likewise, a briefed review of some notable hybrids of DE and PSO are summarized as follows.

Hendtlass in 2001 proposed SDEA [109], where each individual follows traditional PSO and at the time of searching, individual move from a poorer region to a better region with help of DE time to time. Also, it is applied on unconstrained global optimization problems. Further in 2003, Zhang and Xie devised a hybrid algorithm DEPSO [110]. It has a bell-shaped DE mutation to control population diversity and retain the self-organized particle swarm dynamics. Moreover, it is solved through unconstrained and constrained optimization problems. Then Talbi and Batouche developed DEPSO [111] in 2004. This framework follows the iterative scheme as; DE is employed on even iteration (to enhance diversity) and PSO is applied on odd iteration (to extricate the swarm from unwanted fluctuations). Also, it is applied to solve multimodal image problem. Another hybridization of PSO and DE (DEPSO) was delivered by Hao et al. in 2005 [112], in which position of particle is updated partially through DE (extracting its differential information) and partially by PSO (extracting its memory data) due to which swarm's diversity and enhancement of local and global search ability is maintained. It is applied to solve the unconstrained global optimization problems. Later in 2008, Niu and Li utilized the concept of parallel mechanism and proposed PSODE [113], in which one population's individual upgraded by PSO whereas other evolved by DE, this interaction of two populations leads to maintain the diversity. To check its effectiveness applied to solve unimodal and multimodal problems. In 2009, Wang and Cai proposed a hybrid multi-swarm particle swarm optimization (HMPSO) [114], this hybrid model splits the swarm into several sub-swarms and PSO used as search engine for each sub-swarm as well as personal best of each particle is improved by applying DE. Moreover, it is used to solve on constrained optimization problems. Caponio et al. developed a super-fit memetic differential evolution (SFMDE) [115] in 2009. This framework synergized DE with PSO (for super-fit individual) as well as used nelder mead and rosenbrock algorithm as local searcher for measuring quality of super fit individual compared to other individual. Moreover, it is employed for solving optimal control drive design for a direct current (DC) motor and design of a digital filter for image processing purposes. Further, Liu et al. incorporated DE (as it has strong search ability) in PSO (to overcome the stagnation and speed up the convergence) and proposed PSO-DE [116] in 2010. Additionally, it is used to solve welded beam design, tension compression spring design, pressure vessel design, speed reducer design, three-bar truss design problems. Xin et al. proposed DEPSO [117] in year 2010. This hybrid model adopted statistical learning strategy for an individual who leads to the adaptation of evolution methods according to the relative success ratio of alternative methods (DE and PSO). And then applied to solve global numerical optimization problems.

In 2011, Pant et al. developed DE-PSO [118]. It utilized the strength of both algorithm (DE and PSO) and executed through alternating phases i.e. initiated by DE if trial vector better than the corresponding point, then it is added in population otherwise enters the PSO phase to generate a candidate solution. DEPSO is further applied to solve unconstrained global optimization problems. Epitropakis et al. in the 2012 [119] framed a hybrid algorithm in which after each evolution step of PSO, the social and cognitive experience evolved with DE that helps to enhance convergence. Further, this hybrid is employed to solve multimodal functions problems. Later in 2013, Nwankwor et al. [120] developed hybrid particle swarm differential evolution (HPSODE). It initiated with DE till the trial vector is generated otherwise PSO is activated further to generate a new candidate solution. Also, HPSODE is used to apply on optimal well placement problems.

Then Sahu et al. [121] in 2014 utilized the advantages of DE (maintaining diversity) and PSO (memory mechanism) and proposed DEPSO. This hybrid algorithm is used to optimize the gains of fuzzy PID controllers employed in the control areas. Further, in year 2014, HPSO-DE initiated by Yu et al. [122]. It has an adaptive mutation to improve the current population position from local optima and balanced diversity efficiently. Then this hybrid algorithm is used to solve unconstrained global optimization problems. Later in 2015, Seyedmahmoudian et al. [123] proposed DEPSO by employing DE that adds on diversity to traditional PSO. Also detrimental effects of the random coefficients are reduced by DE in parallel with PSO. It is a reliable and system independent technique to track the MPP of PV system under partial shading conditions. Parouha and Das proposed DPD [124] in 2015. It is based on tri-population scheme (inferior, mid and superior group), in which DE is executed in the inferior and superior groups, while PSO is employed in the mid-group. Moreover, elitism and non-redundant search concept are included in DPD cycle to maintain diversity and escape local optima effectively. This hybrid is investigated on the engineering design problems. In year 2016, Tang et al. [125] proposed hybrid of DE and PSO namely HNTVPSORBSADE, where a nonlinear time varying PSO (to update the velocities and positions of particles) and a ranking based self-adaptive DE (to avoid stagnation) are introduced which resulted the exploration and exploitation dynamically. This framework is used for solving mobile robot global path planning problem. Further, Memory based DE (MBDE) is proposed by Parouha and Das [126] in 2016. It employed two operators: swarm mutation and swarm crossover (based on concept of PSO) for DE to direct knowledge and improve the solution quality. Also MBDE is applied on continuous optimization problems. Then, DE-PSO-DE is proposed in 2016 by Parouha and Das [127], in which the population is divided into three groups (A, B, & C) and executed in parallel manner. Additionally, elitism (to retain the best obtained values) and non-redundant search (to improve the solution quality) are evolved in DE-PSO-DE. It is employed for solving economic load dispatch problems. A year later in 2017, Famelis et al. devised DE-PSO [128] where to enhance diversity, DE is merged with a velocity-update rule of PSO in DE-PSO. It is applied on Multimodal optimization problems. Mao et al. in 2018 designed DEMPSO [129], 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. DEMPSO is performed to solve the numerical solution of the forward kinematics of a 3-RPS parallel manipulator. Tang et al. also proposed SAPSO–mSADE [130] in 2018. It integrated self-adaptive PSO (SAPSO) to balance global and local search ability of particles and modified self-adaptive DE (mSADE) to evolve the personal best positions and reduce potential stagnation. It is applied to solve the tension compression spring design and three bar truss design problems. Early in 2019, Too et al. developed binary particle swarm optimization differential evolution (BPSODE) [131]. It inherits the strength of binary PSO (BPSO) and binary DE (BDE) which is computed on sequence. Additionally, the dynamic inertia weight and dynamic crossover rate are introduced to track the optimal solution and balance diversity. And this hybrid algorithm is used to tackle feature selection problems in EMG signals classification. Recently in 2020, Dash et al. proposed HDEPSO [132]. In HDEPSO, three DE operations (mutation, modified crossover and selection) are fused with the best particles of PSO for enhancing global searching ability. Moreover, it is developed to solve the effectiveness of the sharp edge FIR filter (SEFIRF) problem. Since diversity of quantum PSO (QPSO) decline rapidly this shows its inadequacy so Zhao et al. developed improved QPSO [133] in 2020, in which DE is familiarized to improve its diversity and convergence rate. It is employed to solve economic environmental (EED) problem of the microgrid.

3 Proposed methodology

In this section proposed advanced differential evolution (ADE), advanced particle swarm optimization (APSO) and advanced hybrid DEPSO (AHDEPSO) have been described in detail as follows.

3.1 Advanced differential evolution (ADE)

As per earlier extensive studies performance of the DE mostly depends on the following features.

(i). mutation strategy: as it improves the local search ability and convergence rate.
(ii). crossover operator: since it increases the population diversity.
(iii). control parameters (F and C_r): as they improve the search and/or exploration and exploitation ability.

Moreover, above mentioned features are very essential for efficiency of DE because they determine cooperation mechanism among the different individuals. Thus, new DE mutation strategies and crossover operators as well as adjusting control parameters will surly helpful to improve its robustness. Motivated by above observations and to refrain from shortcomings, advanced DE **(**ADE) is proposed in this paper. Where modified mutation strategy and crossover rate as well as altered selection scheme are introduced like so.

$$\textbf{Mutation: } v_{i,j}^{t} = x_{i,j}^{t} + F \times rand(\mathbf{0},\mathbf{1}) \times \left(best_j - x_{i,j}^{t} \right) \tag{6}$$

where $x_{i,j}^{t}$: target vector, $v_{i,j}^{t}$: mutant vector, $rand\,(0, 1)$: uniformly distributed random number between 0 & 1, $best_j$: best vector and F: Scalar factor and it is given as follows.

$$F = F_{min} + (F_{max} - F_{min}) \left(\frac{t_{max} - t}{t_{max}} \right), F_{min} \epsilon\ [0.1, 0.5], F_{max} = 1.$$

when F composed of a higher values series, then it is beneficial to the global search and when F composed of a lower values series, then it is beneficial to the local search.

$$\textbf{Crossover: } u_{i,j}^{t} = \begin{cases} v_{i,j}^{t}\textbf{;} \text{ if } rand(\mathbf{0},\mathbf{1}) \leq C_r \text{ (crossover rate)} \\ x_{i,j}^{t}\ \textbf{;} \quad \text{otherwise} \end{cases} \tag{7}$$

In Eq. (7) C_r is set as $2 - exp\left(\left(\frac{t_{max} - t}{t_{max} - 0.6t} \right)^{10} \right)$. It assured of individual diversity in early stage which improves global search ability. Further reduce degree of difference among individuals which accelerate convergence rate in later stage.

$$\textbf{Selection: } x_{i,j}^{t+1} = \begin{cases} x_{i,j}^{t}; \text{ if } f\left(u_{i,j}^{t}\right) > f\left(x_{i,j}^{t}\right) \text{ and } rand\ (0, 1) < p \\ u_{i,j}^{t}\ \text{; otherwise} \end{cases} \tag{8}$$

where $f\,(\cdot)$: fitness function values and p: random value in (0, 1]. In this selection each pioneer vectors gets chance to survive and share its observed information with other vectors in the next steps. It implies searching capabilities are more enriched. Moreover, it is advantageous for stabilizing essential exploration and exploitation trends to encourage ADE for converging to better quality solutions. The pseudocode of the proposed ADE is presented below.

```
Begin
initialize population $x_{i,j}^{t}$; $\forall\ i = 1,2, \dots, np$ randomly
for $t = 1$
    compute F = $F_{min} + (F_{max} - F_{min}) \left(\frac{t_{max}-t}{t_{max}}\right)$ and $C_r = 2 - exp\left(\left(\frac{t_{max}-t}{t_{max}-0.6t}\right)^{10}\right)$
    for $i$ = 1 to $np$
        ///mutation
        randomly choose one target vectors $x_{i,j}^{t}$ and one best target vectors $best_j$
        generate a mutant (donor) vector $v_{ij}^{t}$ as
        $v_{i,j}^{t} = x_{i,j}^{t} + F \times rand(0,1) \times (best_j - x_{i,j}^{t})$
        ///crossover
        generate a trial vector $u_{i,j}^{t}$
        for $j$ = 1 to $D$
            if $rand$ (0,1) $\leq C_r$ then
                $u_{i,j}^{t} = v_{i,j}^{t}$
            else
                $u_{i,j}^{t} = x_{i,j}^{t}$
            end
        end
        ///selection
        generate $p = rand(0,1]$
        if $f(u_{i,j}^{t}) > f(x_{i,j}^{t})$ and $rand$ (0,1) $< p$ then
            $x_{i,j}^{t+1} = x_{i,j}^{t}$
        else
            $x_{i,j}^{t+1} = u_{i,j}^{t}$
        end
    end
end
$t = t + 1$
End
```

3.2 Advanced particle swarm optimization (APSO)

Based on pros and cons along with existing assessments of PSO, it is essential to strike a good trade-off between global and local search to find an optimal solution. Preferably, PSO needs strong exploration ability (particles can rove entire search space instead of clustering around the current best solution) and boost exploitation capability (particles can explore in a local region) at early and later phase of the evolution respectively. In velocity update equation of the PSO, inertia weight (w) and acceleration coefficient ($c_1$and c_2) are important factors to satisfy the above requirement with the following concerning concept.

(i). a large and small values of w assists exploration and exploitation respectively.
(ii). c_1& c_2 values facilitate exploitation and exploration of the search area based on ensuing strategies.

acceleration coefficient		tactics
c_1(cognitive acceleration coefficient)	c_2 (social acceleration coefficient)	
large	small	exploration
slightly large	slightly small	exploitation
slightly small	slightly large	convergence
small	large	jumping out

Considering all of the concerns like advantages, disadvantages and parameter influences of the PSO, an advanced particle swarm optimization (APSO) is introduced in this study. It relies on novel gradually varying (decreasing and/or increasing) parameters (w, $c_1$and c_2) stated as follows.

$$w=\left(\frac{w_f-w_i}{1-t_{max}}\right)(t-t_{max})+w_i;C_1=\left(\frac{C_{1f}-C_{1i}}{1-t_{max}}\right)(t-t_{max})+C_{1i}\text{ and }C_2=\left(\frac{C_{2f}-C_{2i}}{1-t_{max}}\right)(t-t_{max})+C_{2i}\tag{9}$$

where, w_i and w_f: initial and final values of w; c_{1i} and c_{1f}: initial and final values of c_1; c_{2i} and c_{2f}: initial and final values of c_2; t and t_{max}: iteration index and maximum number of iteration. Hence the velocity and position of the i^{th} particle are updated by the following equations in the proposed APSO.

$$v_{i,j}^{t+1}=\left(\left(\frac{w_f-w_i}{1-t_{max}}\right)(t-t_{max})+w_i\right)v_{i,j}^t+\left(\left(\frac{C_{1f}-C_{1i}}{1-t_{max}}\right)(t-t_{max})+C_{1i}\right)r_1\left(p_{best\ i,j}^t-x_{i,j}^t\right)+\left(\left(\frac{C_{2f}-C_{2i}}{1-t_{max}}\right)(t-t_{max})+C_{2i}\right)r_2\left(g_{bestj}^t-x_{i,j}^t\right)\tag{10}$$

$$x_{i,j}^{t+1}=x_{i,j}^t+v_{i,j}^{t+1}\tag{11}$$

The pseudocode of the proposed APSO is presented below.

```
Begin
initialize: $np$(population randomly), $w_i$, $w_f$, $c_{1i}$, $c_{1f}$, $c_{2i}$ & $c_{2f}$
for $t = 1$
    evaluate fitness function value
    for $i = 1$ to $np$
        find pbest & gbest
        for $j = 1$ to $D$
            update velocity of particles by eq. (10)
            update position of particles by eq. (11)
        end
    end
end
$t = t + 1$
End
```

3.3 Advanced hybrid DEPSO (AHDEPSO)

Introductory reviews and results showed that hybrid algorithm improve the performances of DE & PSO, because they have complimentary properties. Therefore, an advanced hybrid algorithm of advanced DE and PSO (AHDEPSO) are proposed to further improve the solution quality. Basically, AHDEPSO is based on relating superior capability of the proposed ADE and APSO.

In AHDEPSO, entire population is sorted according to the fitness function value and divided into two sub-populations i.e. pop_1 (best half) and pop_2 (rest half). Since pop_1 and pop_2 contains best and rest half of the main population which implies good global and local search capability respectively. In order to maintain local and global search capability, applying proposed ADE (due to its good local search ability) and APSO (because of its virtuous global search capability) on the respective sub-population (pop_1 and pop_2). Evaluating both sub-population then better solution obtained in pop_1 (by using ADE) and pop_2 (by using APSO) are named as *best* and *gbest* separately. If *best* is less than *gbest* then pop_2 is merged with pop_1 thereafter merged population evaluated by ADE (as it mitigate the potential stagnation). Otherwise, pop_1 is merged with pop_2 afterward merged population evaluated by APSO (as it established to guide better movements). Finally, reporting the optimal solution, if stoppings criteria met then stop otherwise returns to sorting process of population. Continue this whole process until get desire optimal solution. The flowchart of AHDEPSO is demonstrated in Fig. 3 and the pseudocode described below.

```
Begin
generate a random initial $np$- population
for $t = 1$
    for $i = 1$ to $np$
        evaluate and sort population according to fitness function values
        divide population into $pop_1$ and $pop_2$ according to best and worst solution
        evaluate $pop_1$ (by ADE) and $pop_2$ (by APSO)
        record best and gbest
        if best < gbest then
            merge $pop_2$ with $pop_1$ & apply ADE
        else
            merge $pop_1$ with $pop_2$ & apply APSO
        end
        report the optimal solution
        if termination criteria are met then
            stop the process
        else
            repeat the process
        end
    end
end
$t = t + 1$
End
```

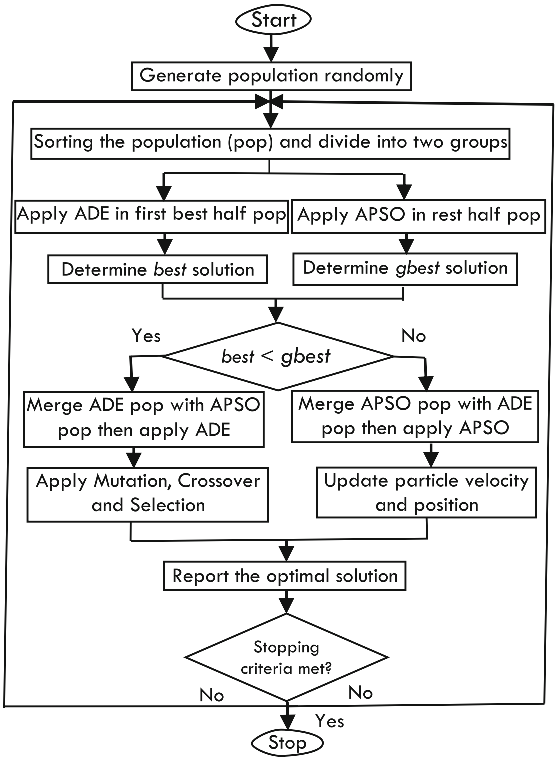

Start
Generate population randomly
Sorting the population (pop) and divide into two groups
Apply ADE in first best half pop
Apply APSO in rest half pop
Determine best solution
Determine gbest solution
Yes
best < gbest
No
Merge ADE pop with APSO pop then apply ADE
Merge APSO pop with ADE pop then apply APSO
Apply Mutation, Crossover and Selection
Update particle velocity and position
Report the optimal solution
Stopping criteria met?
No
No
Yes
Stop

Fig. 3 Flowchart of AHDEPSO

4 Experimental results and discussions

In this section considered optimization problems with their experimental results are discussed as follows.

4.1 Test suite (TS) and real world problems (RWPs)

In order to evaluate performance of the proposed ADE, APSO and AHDEPSO algorithm the following unconstrained test suite (TS) and real world problems (RWPs) are considered to solve.

(i). **TS-1**: 23 basic benchmark functions
(ii). **TS-2**: IEEE CEC 2017
(iii). **RWP-1**: Gear train design problem

$$\text{Minimize } f(x) = \left\{\frac{1}{6.931} - \frac{T_d T_b}{T_a T_f}\right\}^2 = \left\{\frac{1}{6.931} - \frac{x_1 x_2}{x_3 x_4}\right\}^2;$$

$$\text{subject to}: 12 \le x_i \le 600, i = 1, 2, 3, 4.$$

(iv). **RWP-2**: Frequency modulation sound parameter identification problem

$$y(t) = a_1 \sin(w_1 t \ \ + a_2 \sin(w_2 t \ \ + a_3 \sin(w_2 t \ \ )))$$

$$y_0(t) = 1.0\sin((5.0)t \ -(1.5)\sin((4.8)t \ + (2.0)\sin((4.9)t \ )))$$

where: $= \frac{2\pi}{100}$, and $-6.4 \le a_i, \quad w_i \le 6.35, \quad i = 1, \quad 2, \quad 3.$
$\text{Minimize } f(a_1, w_1, a_2, w_2, a_3, w_3) = \sum_{t=0}^{100} (y(t) - y_0(t))^2$

(v). **RWP-3**: The spread spectrum radar poly-phase code design problem

$$\text{Minimize } f(x) = \text{Max}\{f_1(X), \ldots, f_{2m}(X)\},$$

$$X = \left\{(x_1, \ldots, x_n) \in R^n \middle| 0 \le x_j \le 2\pi, j = 1, 2, \ldots, n\right\} \& m = 2n - 1$$

$$\text{with: } f_{2i-1}(x) = \sum_{j=1}^{n} cos\left(\sum_{k=|2i-j-1|+1}^{j} x_k\right) \ i = 1, 2, \ldots, n;$$

$$f_{2i}(x) = 0.5 + \sum_{j=i+1}^{n} cos\left(\sum_{k=|2i-j|+1}^{j} x_k\right) \ i$$

$$= 1, 2, \ldots, n-1; f_{m+i}(x) = -f_i(x), i = 1, 2, \ldots, m.$$

The description of TS-1 is listed in Table 1 which consists of three groups' unimodal (f_1-f_7), multimodal (f_8-f_{13}) and fixed-dimension (f_{14}-f_{23}) function. Also, TS-2 cited in Table 2 which consists of unimodal (h_1-h_3), multimodal (h_4-h_{10}), hybrid (h_{11}-h_{20}) and composition (h_{20}-h_{30}) functions. Moreover, the detailed summary of TS-2 and 3 RWPs are given in [134, 135] respectively.

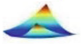
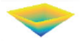
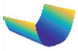
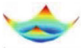
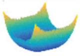
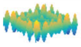
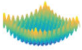
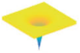
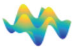
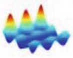
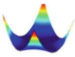
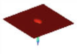
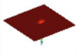
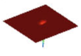
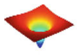
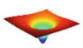
Simulations were conducted on Intel (R) Core (TM) i5–2350 M CPU @ 2.30GHz, RAM: 4.00 GB, Operating System: Microsoft Windows 10, C-free Standard 4.0. An extensive analysis has been carried out to decide the values of parameters w_i, w_f, c_{1i}, c_{1f}, c_{2i} and c_{2f} used in proposed AHDEPSO. For this the values of (w_i, w_f), (c_{1i}, c_{2f}) and (c_{1f}, c_{2i}) varies from (0.1–0.9, 0.1–0.9), (0.1–0.9, 0.1–0.9) and (2.1–2.9, 2.1–2.9) respectively with one step length. The success rate (defined below) of total 81 combinations of these parameters are checked using proposed AHDEPSO with population size (30), stopping criteria (500 iterations) and independent run (30) on test suite TS-1 and TS-2 (30D).

$$\text{Success rate (SR)} = \frac{\text{Number of succesful runs}}{\text{Total nimber of runs}}$$

where a run is declaired as a 'successful run' if $|\boldsymbol{f}(\boldsymbol{x}) - \boldsymbol{f}(\boldsymbol{x}^*)| \le \epsilon$, where $\boldsymbol{f}(\boldsymbol{x})$ is the known global minima and $\boldsymbol{f}(\boldsymbol{x}^*)$ is the obtained minima. In this study ϵ is fixed at 0.0001.

The success rate on best 10 combinations of ($\boldsymbol{w_i}$, $\boldsymbol{w_f}$) and ($\boldsymbol{c_{1i}}$ & $\boldsymbol{c_{2f}}$, $\boldsymbol{c_{1f}}$ & $\boldsymbol{c_{2i}}$) are presented in Figs. 4 and 5 respectively. From these figures, it is clearly noticeable that the highest success rate can be found at (0.4, 0.9) in case of inertia weight and (0.5, 2.5) in case of acceleration coefficients. Hence, $\boldsymbol{w_i}$ = 0.4, $\boldsymbol{w_f}$ = 0.9, $\boldsymbol{c_{1i}}$ = 0.5, $\boldsymbol{c_{1f}}$ = 2.5, $\boldsymbol{c_{2i}}$ = 2.5 and $\boldsymbol{c_{2f}}$ = 0.5 have been recommended to use in proposed AHDEPSO. The overall best values in each table are highlighted with boldface letters of the corresponding algorithms. In all experiments, for fair comparison common parameters such as population size, stopping criteria and independent run are set the same or minimum of comparative algorithms. The results of the comparative

Table 1 Test Suite (TS)-1

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

algorithms are directly taken from the original references. The simulation result analysis on TS-1, TS-2 and RWPs with comparative experiments are presented below.

4.2 Numerical and graphical analysis

The simulation result analysis on TS-1, TS-2 and RWPs with comparative experiments are presented below.

i). On TS-1: 23 basic unconstrained benchmark functions

The produced result by proposed algorithms on TS-1 is compared with traditional algorithms (PSO [3] & DE [4]), DE variants (JADE [136] & SHADE [137]), PSO variants (HEPSO [138] & RPSOLF [139]) and hybrid variants (FAPSO [140] & PSOSCALF [141]). The parameters of all above compared and proposed algorithms are listed in Table 3. The comparative experimental results in terms of

Table 2 Test suite (TS)-2 (IEEE CEC2017 unconstrained benchmark functions)

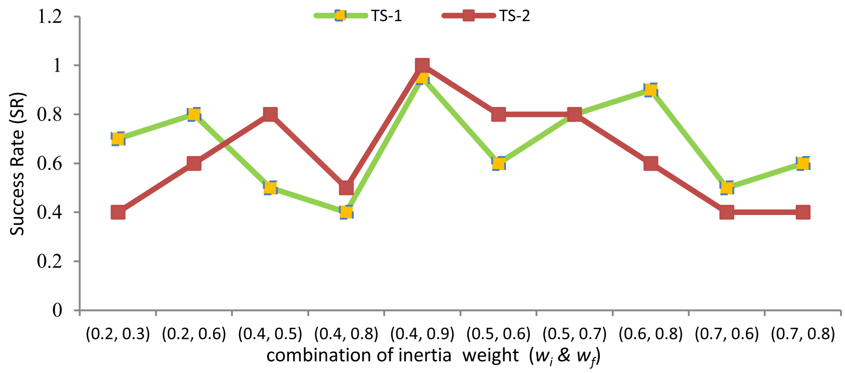
Type	No.	Functions	Optimum
Unimodal	h_1	Shifted and Rotated Bent Cigar Function	100
	h_2	Shifted and Rotated Sum of Different Power Function	200
	h_3	Shifted and Rotated Zakharov Function	300
Multimodal	h_4	Shifted and Rotated Rosenbrock's Function	400
	h_5	Shifted and Rotated Rastrigin's Function	500
	h_6	Shifted and Rotated Expeanded Scaffer's F6 Function	600
	h_7	Shifted and Rotated LunacekBi_Rastrigin's Function	700
	h_8	Shifted and Rotated Non-Continuous Rastrigin's Function	800
	h_9	Shifted and Rotated Levy Function	900
	h_{10}	Shifted and Rotated Schwefel's Function	1000
Hybrid	h_{11}	Hybrid Function 1 (N = 3)	1100
	h_{12}	Hybrid Function 2 (N = 3)	1200
	h_{13}	Hybrid Function 3 (N = 3)	1300
	h_{14}	Hybrid Function 4 (N = 4)	1400
	h_{15}	Hybrid Function 5 ($N = 4$)	1500
	h_{16}	Hybrid Function 6 ($N = 4$)	1600
	h_{17}	Hybrid Function 7 (N = 5)	1700
	h_{18}	Hybrid Function 8 (N = 5)	1800
	h_{19}	Hybrid Function 9 ($N = 5$)	1900
	h_{20}	Hybrid Function 10 (N = 6)	2000
Composition	h_{21}	Composition Function 1 (N = 3)	2100
	h_{22}	Composition Function 2 ($N = 3$)	2200
	h_{23}	Composition Function 2 (N = 4)	2300
	h_{24}	Composition Function 2 (N = 4)	2400
	h_{25}	Composition Function 3 (N = 5)	2500
	h_{26}	Composition Function 4 (N = 5)	2600
	h_{27}	Composition Function 5 (N = 6)	2700
	h_{28}	Composition Function 6 ($N = 6$)	2800
	h_{29}	Composition Function7 (N = 3)	2900
	h_{30}	Composition Function 8 (N = 3)	3000

Search space: $[-100, 100]^D$

mean, std. (standard deviation) and ranking of the objective function values are presented in Table 4 of 30 independent runs.

It should be noted that from Table 4, the mean objective function values of the proposed ADE, APSO and AHDEPSO algorithms are better and/or equal in comparison of above listed traditional algorithms, DE variants, PSO variants and

Fig. 4 Influence of different inertia weights of proposed AHDEPSO on test suite (TS)-1 and TS-2



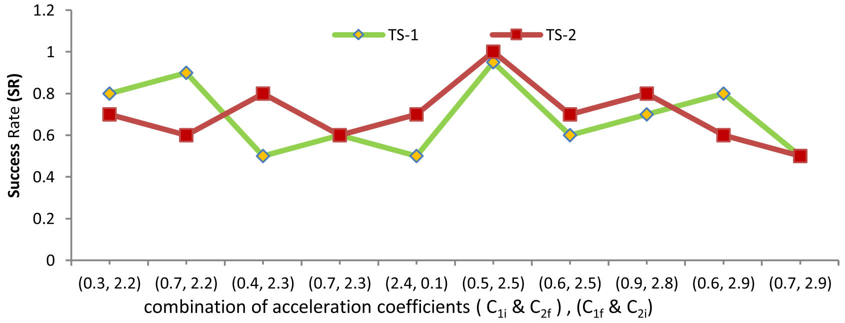


Fig. 5 Influence of different acceleration coefficients of proposed AHDEPSO on test suite (TS)-1 and TS-2

hybrid variants. As per the experimental results shown in Table 4 the following comparison results (among non-proposed algorithms) are summarized as follows for TS-1 cases (i). Unimodal function (f_1-f_7): proposed AHDEPSO obtained better results in all seven functions (f_1-f_7), Suggested ADE and APSO obtained better results for f_1, f_2, f_3 and f_6 functions and marginally similar for the rest functions. (ii). Multimodal function (f_8-f_{13}): proposed AHDEPSO obtained better results for all six function (f_8-f_{13}) and similar for f_8 (on DE, JADE and PSOSCALF), f_9 (on RPSOLF, FAPSO & PSOSCALF) and f_{11} (on RPSOLF & PSOSCALF). Suggested ADE attained better results for f_8, f_9, f_{11} and marginally similar/inferior for the rest functions whereas, APSO obtained better result for f_9 and slightly inferior for the rest. (iii). Fixed-dimension function (f_{14}-f_{23}): proposed AHDEPSO and ADE exhibits best performance on all functions meanwhile APSO obtained marginally better or equal results compared to other algorithms. Moreover, all algorithms are individually ranked (as '1'for the best and '2' for subsequent performer and so on) in Table 4 based on mean result values. From this table it is concluded that AHDEPSO, ADE and APSO ranked 1st, 2nd and 3rd sequentially. Also, average and overall rank of proposed algorithms Vs others are presented in Table 4. It is clear that (from ranking) performances of proposed algorithms are superior to others. Eventually, proposed ADE, APSO and AHDEPSO produce less std. (it may 0.00E+00) for most of the cases on TS-1 which describes their stability. Furthermore, superiority of proposed algorithms is statistically validated over other algorithms through one-tailed*t*-test (with 98 degree of freedom (df) at 5% significance level) and Wilcoxon Signed Rank (WSR) test (at 5% significance level). The details of these tests can be found in [142]. The results of *t*-test and WSR test on TS-1 are reported in Table 5. From Table 5, it can be seen that proposed algorithms has both 'a (significantly better than other)' or 'a^+ (highly significance with other)' sign (in case of t-test) and performs better or equally (in case WSR test) in most of consequence. Also, the *p* values as reported in Table 5 of the proposed algorithms are less with others which conclude that simulations are reliable for the majority of runs.

The convergence speed of proposed and comparative algorithms is compared over 8 (f_1, f_5, f_6, f_7, f_8, f_9, f_{10} and f_{11}) typical 30-*D* TS-1. All plotted convergence graphs (objective function values Vs iterations) are separately presented in

Table 3 Parameter setting for TS-1

Algorithm	Year	Ref.	Control parameter		Population size	Stopping criterion (iterations)	Run
			Term	Values			
PSO	1995	[3]	w, C_1 & C_2	Linear reduction from 0.9 to 0.1, 2 & 2	30	500	30
DE	1997	[4]	F & *CR*	0.5 & 0.5	30	500	30
JADE	2009	[136]	F_i & CR_i,	randc$_i$ (μ_F, 0.1) & randn$_i$ (μ_{CR}, 0.1)	50	1000	30
SHADE	2013	[137]	P*best* & Arc rate	0.1 & 2	30	500	30
HEPSO	2014	[138]	P_C & P_B	0.95 & 0.02	50	500	30
RPSOLF	2017	[139]	w, c_1, c_2, c_3, β & ε	0.55, 1.49, 1.49, 1.5 & 0.99	50	500	30
FAPSO	2018	[140]	–	–	50	5000	30
PSOSCALF	2018	[141]	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
ADE	Proposed		F_{min} & F_{max}	[0.1,0.5] & 1	30	500	30
APSO			w_i, w_f, c_{1i}, c_{1f}, c_{2i} & c_{2f}	0.4, 0.9, 0.5, 2.5, 2.5 & 0.5	30	500	30
AHDEPSO			–	–	30	500	30

Table 4 Simulation results on TS-1

f_n	Criteria	Algorithms										
		Traditional algorithms		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		PSO	DE	JADE	SHADE	HEPSO	RPSOLF	FAPSO	PSOSCALF	ADE	APSO	AHDEPSO
f_1	mean	9.59e-06	3.82e-12	1.87e-31	1.42e-09	16.26772	5.065e-269	2.87e−127	1.11014e-20	**0.00e+000**	**0.00e+000**	**0.00e+000**
	std.	3.35e-05	2.00e-12	6.43e-31	3.09e-09	10.01293	**0.00e+00**	1.76e−127	1.83289e-20	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	8	6	4	7	9	2	3	5	**1**	**1**	**1**
f_2	mean	0.02560	4.40e-08	2.79e-15	0.0087	1.28424	1.000e-134	1.02e-17	4.09460e-11	**0.00e+000**	**0.00e+000**	**0.00e+000**
	std.	0.04595	1.30e-08	9.51e-15	0.0213	0.41611	3.753e-134	1.43e-17	5.68981e-11	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	8	6	4	7	9	2	3	5	**1**	**1**	**1**
f_3	mean	82.2687	2.48e+04	1.10e-03	15.4352	7.423e+03	7.791e-249	1.68e-11	2.16858e-12	**0.00e+000**	0.17e-129	**0.00e+000**
	std.	97.2105	4.04e+03	5.14e-03	9.9489	7.423e+03	0.00e+00	2.49e-11	1.03815e-11	**0.00e+000**	1.67e-131	**0.00e+000**
	rank	8	10	6	7	9	2	5	4	**1**	3	**1**
f_4	mean	4.26128	1.96e+00	1.66e-03	0.9796	23.95145	1.937e-157	4.09e+03	8.47410e-08	3.28e-101	8.35e-098	**0.00e+000**
	std.	0.67730	3.30e-01	1.98e-03	0.7995	7.71460	1.061e-156	6.53e+02	1.23324e-07	7.12e-103	7.08e-098	**0.00e+000**
	rank	9	8	6	7	10	2	11	5	3	4	**1**
f_5	mean	92.4310	4.43e+01	1.18e+01	24.4743	2.380e+03	27.42672	6.55e-11	21.97646	1.25e-021	4.85e-012	**2.21e-033**
	std.	74.4794	2.21e+01	1.57e+01	11.2080	1.852e+03	0.24848	1.99e-11	0.54774	1.85e-023	3.21e-012	**3.72e-037**
	rank	10	9	5	7	11	8	4	6	2	3	**1**
f_6	mean	8.89e-06	4.52e-12	4.59e-31	5.31e-10	21.55405	2.98244	2.37e-12	7.13998e-12	**0.00e+000**	1.75e-032	**0.00e+000**
	std.	9.91e-06	1.86e-12	1.65e-30	6.35e-10	9.33263	0.23250	1.84e-13	3.65884e-11	**0.00e+000**	9.16e-035	**0.00e+000**
	rank	8	5	3	7	10	9	6	4	**1**	2	**1**
f_7	mean	0.02724	2.50e-02	6.49e-03	0.0235	0.12982	0.00104	**0.00e+00**	0.00012	2.19e-003	5.11e-002	6.47e-003
	std.	0.00804	4.35e-03	2.48e-03	0.0088	0.09727	7.644e-04	**0.00e+00**	0.00010	1.10e-004	1.70e-002	1.40e-005
	rank	8	9	2	10	11	6	**1**	5	4	7	3
f_8	mean	−6075.85	**−1.25e+04**	**−1.24e+04**	−11,713.1	−2.139e+03	−3.254e+03	2.48e-11	**−12,569.48**	**−1.25e+004**	−6.37e+003	**−1.25e+004**
	std.	754.632	2.09e+02	1.27e+02	230.49	8.282e+02	2.860e+02	6.44e-12	2.39996e-07	1.07e-017	2.10e-001	**0.00e+000**
	rank	4	**1**	**1**	2	6	5	7	**1**	**1**	3	**1**
f_9	mean	52.8322	5.92e+01	1.71e-04	8.5332	42.00118	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	std.	16.7068	4.90e+00	1.52e-04	2.1959	7.08632	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	5	6	2	3	4	**1**	**1**	**1**	**1**	**1**	**1**
f_{10}	mean	0.00501	5.56e-07	1.31e-14	0.3957	2.83842	4.085e-15	4.86e-15	2.24609e-11	1.44e-015	1.97e-014	**2.88e-016**
	std.	0.01257	1.62e-07	2.46e-14	0.5868	0.66134	1.084e-15	1.74e-15	2.33542e-11	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	9	8	5	10	11	3	4	7	2	6	**1**
f_{11}	mean	0.02381	1.02e-10	2.87e-03	0.0048	1.16858	**0.00e+00**	1.74e-16	**0.00e+00**	**0.00e+000**	3.37e-111	**0.00e+000**
	std.	0.02870	2.02e-10	7.85e-03	0.0077	0.12602	**0.00e+00**	3.60e-16	**0.00e+00**	**0.00e+000**	1.11e-119	**0.00e+000**
	rank	7	4	5	6	8	**1**	3	**1**	**1**	2	**1**
f_{12}	mean	0.02764	5.54e-13	1.73e-02	0.0346	0.47856	0.26157	1.57e-32	8.46465e-14	1.05e-032	3.34e-030	**3.73e-033**
	std.	0.05399	3.57e-13	7.74e-02	0.0875	0.22623	0.03386	**0.00e+00**	2.79106e-13	2.77e-034	1.12e-028	**3.18e-034**
	rank	8	6	7	9	11	10	3	5	2	4	**1**
f_{13}	mean	0.00732	2.38e-12	5.45e-24	7.32e-04	1.85056	2.05282	1.58e-32	0.00399	2.09e-025	9.30e-020	**1.05e-032**
	std.	0.01050	1.61e-12	2.58e-23	0.0028	0.65246	0.16579	**0.00e+00**	0.00928	3.27e-023	3.71e-018	**1.89e-043**
	rank	9	6	4	7	10	11	2	8	3	5	**1**
f_{14}	mean	3.84902	1.03e+00	**9.98e-01**	**0.998004**	**0.99800**	1.54064	**9.98e-001**	1.13027	**9.98e-001**	**9.98e-001**	**9.98e-001**
	std.	3.24864	1.81e-01	**0.00e+00**	5.83e-17	9.219e-17	1.84429	1.27e-08	0.50338	**0.00e+000**	**0.00e+000**	**0.00e+000**

Table 4 (continued)

f_n	Criteria	Algorithms										
		Traditional algorithms		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		PSO	DE	JADE	SHADE	HEPSO	RPSOLF	FAPSO	PSOSCALF	ADE	APSO	AHDEPSO
	rank	5	2	**1**	**1**	**1**	4	**1**	3	**1**	**1**	**1**
f_{15}	mean	0.002434	1.29e-03	3.01e-03	0.002374	6.404e-04	0.00171	3.95e-04	3.13244e-04	3.83e-004	3.99e-004	**3.02e-004**
	std.	0.006081	3.60e-03	6.92e-03	0.0061	2.801e-04	0.00508	6.02e-08	2.17489e-05	9.23e-012	2.96e-007	**2.38e-018**
	rank	10	8	11	9	6	7	4	3	2	5	**1**
f_{16}	mean	**−1.03162**	**−1.03e+00**	**−1.03e+00**	**−1.03162**	**−1.03162**	**−1.03161**	**−1.03e+00**	**−1.0316**	**−1.03e+000**	**−1.03e+000**	**−1.03e+000**
	std.	6.51e-16	6.78e-16	6.78e-16	6.51e-16	3.554e-15	1.650e-05	**0.00e+00**	4.40244e-16	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
f_{17}	mean	0.397887	**3.98e-01**	**3.98e-01**	0.397887	0.39788	**0.39837**	**3.98e-001**	0.39788	**3.98e-001**	**3.98e-001**	**3.98e-001**
	std.	**0.00e+00**	**0.00e+00**	**0.00e+00**	3.24e-16	6.594e-13	5.267e-04	**0.00e+000**	3.66527e-15	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	2	**1**	**1**	2	2	**1**	**1**	2	**1**	**1**	**1**
f_{18}	mean	**3.00e+00**	**3.00e+00**	**3.00e+00**	**3.00e+00**	0.65246	**3.00002**	**3.00e+000**	**3. 00e+00**	**3.00e+000**	**3.00e+000**	**3.00e+000**
	std.	1.97e-15	1.23e-15	1.82e-15	1.87e-15	5.146e-11	1.658e-05	5.31e-016	5.96540e-13	**0.00e+000**	1.33e-018	**0.00e+000**
	rank	**1**	**1**	**1**	**1**	2	**1**	**1**	**1**	**1**	**1**	**1**
f_{19}	mean	−3.86278	**−3.86e+00**	**−3.86e+00**	−3.86278	−3.86278	−3.85923	−3.87e+000	**−3.86278**	**−3.86e+000**	**−3.86e+000**	**−3.86e+000**
	std.	2.65e-15	2.71e-15	2.71e-15	2.69e-15	1.008e-13	0.00283	2.11e-004	8.31755e-15	**0.00e+000**	3.36e-021	**0.00e+000**
	rank	2	**1**	**1**	2	2	3	4	2	**1**	**1**	**1**
f_{20}	mean	−3.26651	**−3.32e+00**	−3.29e+00	−3.27047	−3.31803	−3.10441	−3.29e+000	−3.27168	**−3.32e+000**	−3.27e+000	**−3.32e+000**
	std.	0.06032	7.73e-04	5.11e-02	0.0599	0.02170	0.15760	**0.00e+000**	0.06371	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	7	**1**	3	5	2	8	3	6	**1**	4	**1**
f_{21}	mean	−5.9092	−1.02e+01	−9.14e+00	−9.2343	**−10.15319**	−4.76171	−7.51e+00	**−10.15319**	**−1.01e+001**	−9.87e+001	**−1.01e+001**
	std.	3.59559	1.18e-07	2.06e+00	1.3969	2.680e-05	0.73723	1.21e-01	4.46227e-15	**0.00e+000**	1.07e-011	**0.00e+000**
	rank	8	2	5	4	**1**	7	6	**1**	**1**	3	**1**
f_{22}	mean	−7.3360	**−1.04e+01**	−9.88e+00	−10.2809	−10.39978	−4.81927	−6.04e+01	**−10.40294**	**−1.04e+001**	−9.87e+000	**−1.04e+001**
	std.	3.47381	1.07e-02	1.61e+00	1.3995	0.01728	0.75699	2.14e-01	**1.80672e-15**	2.91e-012	2.08e-002	**1.10e-016**
	rank	6	**1**	4	3	2	8	7	**1**	**1**	5	**1**
f_{23}	mean	−8.7482	**−1.05e+01**	−1.03e+01	63.333	**−10.53640**	−5.06376	−5.64e+00	**−10.53640**	**−1.05e+001**	−9.13e+000	**−1.05e+001**
	std.	2.55743	3.17e-12	1.40e+00	80.872	5.845e-07	0.82968	5.70e-02	4.84794e-15	1.07e-021	2.05e-003	**0.00e+000**
	rank	4	**1**	2	7	**1**	6	5	**1**	**1**	3	**1**
Sum of rank		147	103	84	124	139	108	86	83	34	67	**24**
Average		6.39	4.47	3.65	5.39	6.04	4.69	3.73	3.60	1.47	2.91	**1.04**
Overall rank		11	7	5	9	10	8	6	4	2	3	**1**

Table 5 Statistical comparisons of proposed Vs other algorithms for TS-1

Vs	Criteria	Algorithms									
		Traditional algorithms		DE variants		PSO variants		Hybrid variants		Proposed algorithms	
		PSO	DE	JADE	SHADE	HEPSO	RPSOLF	FAPSO	PSOSCALF	APSO	AHDEPSO
ADE	better	21	15	16	20	19	18	16	15	15	0
	equal	2	8	6	3	4	5	5	8	8	16
	worst	0	0	1	0	0	0	2	0	0	7
	R^+	217	194	220	255	197	150	164	232	154	217
	R^-	59	82	56	21	79	126	112	44	122	59
	p value	5.2e-07	5.6e-09	8.2e-07	5.8e-08	6.2e-04	6.9e-07	4.3e-09	6.2e-08	5.1e-10	6.9e-012
	t-test	a	a	a	a+	a	a	a	a+	a+	a+
	decision	+	+	+	≈	+	+	+	≈	+	+
Vs		PSO	DE	JADE	SHADE	HEPSO	RPSOLF	FAPSO	PSOSCALF	ADE	AHDEPSO
APSO	better	21	14	11	18	17	14	12	15	0	0
	equal	2	4	5	3	2	4	4	2	8	8
	worst	0	5	7	2	4	5	7	6	15	15
	R^+	253	220	181	212	168	192	211	153	150	168
	R^-	23	56	95	64	108	84	65	123	126	108
	p value	5.3e-08	5.1e-07	6.2e-05	4.6e-08	5.7e-07	5.1e-03	5.6e-07	5.8e-07	6.2e-09	5.3e-11
	t-test	a	a	a	a+	a	a+	a+	a+	a	a+
	decision	+	+	+	+	≈	+	+	+	+	+
Vs		PSO	DE	JADE	SHADE	HEPSO	RPSOLF	FAPSO	PSOSCALF	ADE	APSO
AHDEPSO	better	0	0	17	20	19	18	17	15	7	15
	equal	2	8	5	3	4	5	5	8	16	8
	worst	21	15	1	0	0	0	1	0	0	0
	R^+	171	177	145	220	227	217	164	207	156	187
	R^-	105	99	131	56	49	59	112	69	120	89
	p value	5.1e-08	6.2e-09	4.6e-08	5.7e-07	5.1e-07	5.1e-09	5.3e-08	6.2e-09	4.6e-10	5.7e-11
	t-test	a	a	a	a+	a	a+	a+	a	a+	a
	decision	+	+	+	+	+	≈	+	+	+	+

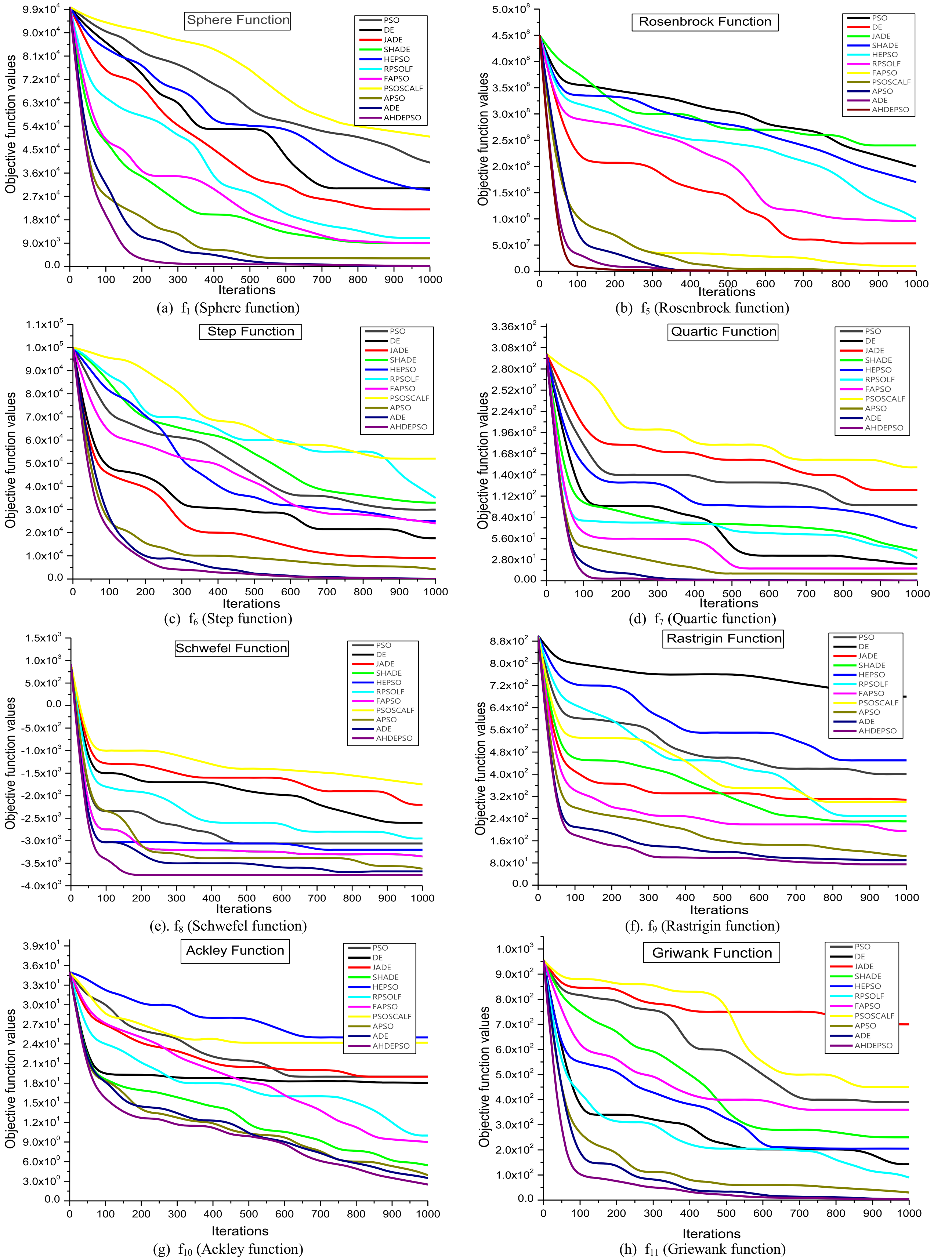
Fig. 6a–h. From this figures it can be concluded that proposed ADE, APSO and AHDEPSO converge much faster than other algorithms in all cases. Also, total of 690 runs (30 runs for each TS-1 with 30 population size) optimal solutions are illustrated in Fig. 7. It confers that the proposed algorithms score the highest optimum solutions. Apart from this the computational time of proposed and compared algorithms on each TS-1 are computed and presented in Fig. 8. From this figure, it can be observed that the proposed algorithms take lesser time to achieve the best value for the entire TS-1.

As a whole, above numerical, statistical and graphical result analysis shows that proposed ADE, APSO and AHDEPSO performs very competitive and/or equally with other compared algorithms. However, among three proposed algorithms AHDEPSO is superior i.e. ranking order of proposed algorithms to solve TS-1 is AHDEPSO>ADE > APSO.

ii). On TS-2: IEEE CEC 2017 unconstrained benchmark functions

Further, the produced result by proposed ADE, APSO and AHDEPSO on TS-2 is compared with DE variants (MPEDE [143] & EFADE [144]), PSO variants (HEPSO [138] & CSPSO [145]) and hybrid variants (HPSODE [120] & PSOJADE [146]). The parameters of all above compared and proposed algorithms are listed in Table 6 for TS-2. The relative experimental results in terms of mean error, std. (standard deviation) and ranking of the objective function values are presented in Tables 7, 8 and 9 (for 10*D*, 30*D* & 50*D* TS-2) of 30 independent runs.

From Tables 7, 8 and 9, it should be noted that the mean values of the proposed ADE, APSO and AHDEPSO algorithms are better and/or equal in comparison of all compared algorithms in test suites. As per the experimental results shown in Table 7 following comparison results are



(a) f_1 (Sphere function)

(b) f_5 (Rosenbrock function)

(c) f_6 (Step function)

(d) f_7 (Quartic function)

(e). f_8 (Schwefel function)

(f). f_9 (Rastrigin function)

(g) f_{10} (Ackley function)

(h) f_{11} (Griewank function)

Fig. 6 **a-h** Convergence graph of compared and proposed algorithms for TS-1

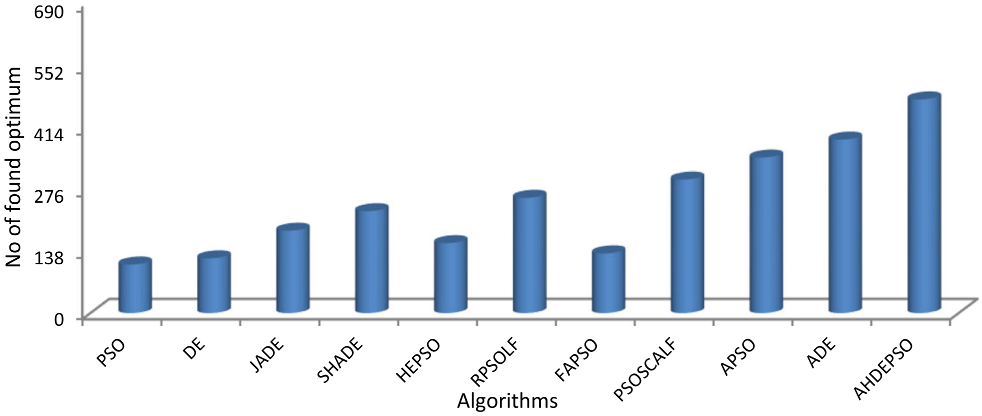


Fig. 7 Comparison of algorithms in finding the global optimal solution out of 690 runs

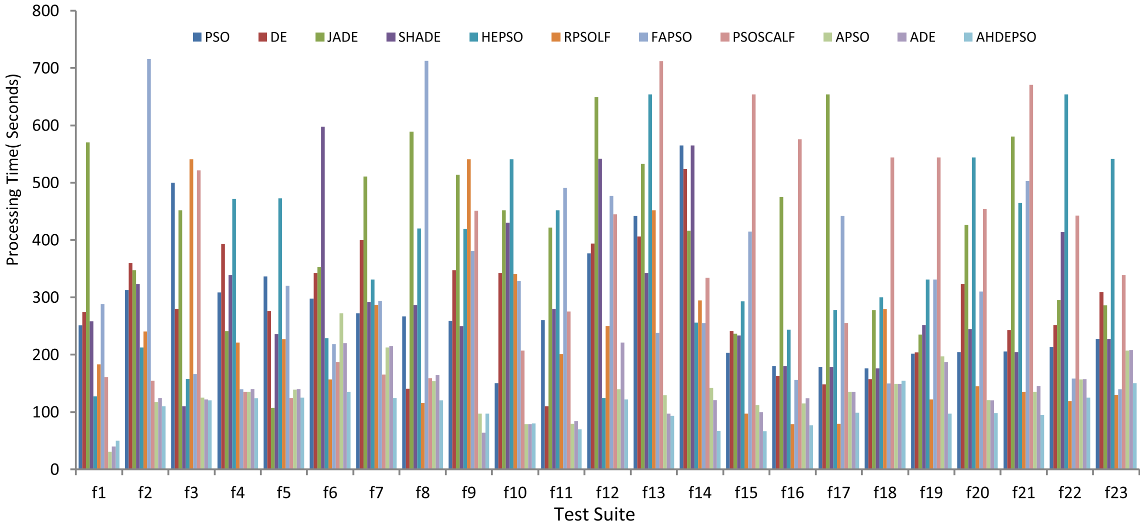


Fig. 8 Processing times of algorithms for TS-1

Table 6 Parameter settings for TS-2

Algorithm	Year	Ref.	Control parameter		Population Size	Stopping criterion (MaxFEs)	Run
			Term	Values			
MPEDE	2016	[143]	λ_1 & ng	(0.1, 0.15, 0.25 & 0.3) & (10, 30, 50, 80)	40	2×10^5	30
EFADE	2018	[144]	F_i, CR_1 & CR_2	rand (0, k_i) where i = 1, 3,2; [0.05, 0.15] & [0.9, 1].	30	2×10^5	51
HEPSO	2014	[138]	P_C & P_B	0.95 & 0.02	40	2×10^5	30
CSPSO	2014	[145]	–	–	50	2×10^5	30
HPSODE	2014	[120]	PSO , DE , &	0.3, 0.01, 1.5 & 0	40	2×10^5	30
PSOJADE	2019	[146]	p	0.5	40	2×10^5	30
ADE	Proposed		F_{min} & F_{max}	[0.1,0.5] & 1	30	2×10^5	30
APSO			w_i, w_f, c_{1i}, c_{1f}, c_{2i} & c_{2f}	0.4, 0.9, 0.5, 2.5, 2.5 & 0.5	30	2×10^5	30
AHDEPSO			–	–	30	2×10^5	30

MaxFEs maximum number of function evaluations,

Table 7 Simulation results on TS-2 (10*D*)

h_n	Criteria	Algorithms								
		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO	AHDEPSO
h_1	mean	**0.00e+000**	**0.00e+00**	5.67e+000	1.29e+002	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	std.	**0.00e+000**	**0.00e+00**	1.66e+001	1.65e+002	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	**1**	**1**	2	3	**1**	**1**	**1**	**1**	**1**
h_2	mean	5.55e-016	**0.00e+00**	1.06e-007	1.28e-006	3.49e-014	1.97e-017	**0.00e+000**	**0.00e+000**	**0.00e+000**
	std.	2.53e-015	**0.00e+00**	1.40e-007	1.11e-006	1.63e-013	4.71e-017	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	3	**1**	5	6	4	2	**1**	**1**	**1**
h_3	mean	**0.00e+000**	**0.00e+00**	2.75e-014	**0.00e+00**	**0.00e+00**	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	std.	**0.00e+000**	**0.00e+00**	4.56e-014	**0.00e+00**	**0.00e+00**	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	**1**	**1**	2	**1**	**1**	**1**	**1**	**1**	**1**
h_4	mean	7.60e-032	**0.00e+00**	7.70e-006	1.78e-005	3.09e-030	**0.00e+00**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	std.	4.16e-031	**0.00e+00**	8.14e-006	6.48e-005	8.85e-030	**0.00e+00**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	2	**1**	4	5	3	**1**	**1**	**1**	**1**
h_5	mean	5.83e+000	6.49e+00	**0.00e+000**	3.42e+000	3.51e+000	3.97e-001	1.66e+000	2.00e+000	**0.00e+000**
	std.	3.35e+000	1.44e+00	**0.00e+000**	9.75e-001	1.53e+000	6.71e-001	2.45e-002	3.70e-002	**0.00e+000**
	rank	7	8	**1**	5	6	2	3	4	**1**
h_6	mean	**0.00e+000**	4.68e-14	2.06e-017	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**	1.40e-007	**0.00e+000**
	std.	**0.00e+000**	5.65e-14	7.14e-017	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**	9.90e-007	**0.00e+000**
	rank	**1**	3	2	**1**	**1**	**1**	**1**	2	**1**
h_7	mean	1.50e+001	1.91e+01	9.04e+000	1.39e+001	1.21e+001	1.07e+001	**0.00e+000**	**1.03e+001**	**0.00e+000**
	std.	1.55e+000	1.71e+00	3.60e+000	1.33e+000	1.15e+000	3.24e-001	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	`7	8	2	6	5	4	**1**	3	**1**
h_8	mean	5.83e+000	7.68e+00	**0.00e+000**	3.24e+000	2.85e+000	3.97e-001	**0.00e+000**	2.38e-003	**0.00e+000**
	std.	2.51e+000	1.99e+00	**0.00e+000**	8.93e-001	1.64e+000	6.18e-001	**0.00e+000**	0.16e+000	**0.00e+000**
	rank	6	**7**	**1**	5	4	3	**1**	2	**1**
h_9	mean	2.01e-029	**0.00e+00**	3.54e-022	1.99e-029	1.98e-029	1.98e-029	**0.00e+000**	1.87e-029	**0.00e+000**
	std.	1.52e-030	**0.00e+00**	1.93e-021	3.18e-031	8.55e-045	8.55e-045	**0.00e+000**	6.68e-032	**0.00e+000**
	rank	3	**1**	7	4	5	5	2	6	**1**
h_{10}	mean	3.36e+002	2.76e+02	2.69e+001	3.80e+001	3.87e+002	9.56e+000	1.30e+001	1.45e+001	**0.00e+000**
	std.	1.65e+002	8.44e+01	2.81e+001	4.65e+001	1.57e+002	3.02e+001	3.82e+009	5.62e+007	**0.00e+000**
	rank	8	7	5	6	9	2	3	4	**1**
h_{11}	mean	1.64e+000	1.40e+00	**1.11e-011**	1.05e+000	4.64e-001	6.63e-002	1.73e-002	2.22e-002	1.07e-003
	std.	1.88e+000	1.05e+00	**1.88e-011**	1.02e+000	6.25e-001	2.52e-001	1.80e-001	7.91e-002	2.21e-002
	rank	9	8	**1**	7	6	5	3	4	2
h_{12}	mean	2.96e+002	**1.62e+00**	4.43e+003	4.92e+003	8.20e+001	7.49e+001	1.90e+002	2.96e+002	**3.13e+001**
	std.	1.44e+002	3.54e+00	4.07e+003	3.43e+003	1.24e+002	1.05e+002	8.30e+002	1.35e+002	**2.42e+003**
	rank	6	**1**	7	8	4	3	5	6	**2**
h_{13}	mean	5.66e+000	3.60e+00	2.14e+000	1.71e+002	4.42e+000	2.23e+000	1.20e+000	3.46e+000	**0.00e+000**
	std.	3.67e+000	2.89e+00	3.26e+000	1.68e+002	1.91e+000	2.10e+000	1.70e+002	8.55e+002	**0.00e+000**
	rank	8	6	3	9	7	4	2	5	**1**
h_{14}	mean	7.02e+000	3.85e-01	1.91e-013	1.19e+000	3.97e-001	**0.00e+000**	3.60e-009	2.04e-008	**0.00e+000**
	std.	8.66e+000	4.52e-01	6.82e-013	9.45e-001	5.60e-001	**0.00e+000**	2.98e-008	7.02e-004	**0.00e+000**
	rank	8	6	2	7	5	1	3	4	**1**
h_{15}	mean	9.32e-001	1.26e-01	2.04e-002	7.84e-001	2.23e-001	**3.21e-003**	5.01e-001	3.25e-001	2.28e-002
	std.	7.86e-001	3.11e-01	3.59e-002	4.84e-001	3.53e-001	**4.23e-003**	5.50e-001	7.91e-001	2.29e-001
	rank	4	9	2	5	8	**1**	6	7	3

Table 7 (continued)

h_n	Criteria	Algorithms								
		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO	AHDEPSO
h_{16}	mean	1.49e+001	**8.59e-01**	2.91e-001	3.45e-001	1.28e+001	1.55e-001	7.50e-001	5.93e-001	8.03e-001
	std.	3.19e+001	3.01e-01	4.40e-001	1.84e-001	2.43e+001	1.80e-001	4.60e-001	5.41e-001	**7.45e-001**
	rank	9	**1**	6	5	8	7	3	4	2
h_{17}	mean	1.33e+001	1.59e+00	2.84e-002	9.72e-001	2.46e+000	1.78e+000	2.77e-002	1.90e-002	**1.18e-003**
	std.	2.30e+001	8.37e-01	3.50e-002	5.41e-001	5.70e+000	5.07e+000	1.13e-002	3.11e-002	**7.86e-002**
	rank	9	6	2	5	8	7	3	4	**1**
h_{18}	mean	9.96e+000	2.45e-01	2.54e-002	1.75e+001	8.81e-001	**7.81e-002**	6.90e-001	5.70e-001	**1.89e-002**
	std.	9.68e+000	3.33e-01	2.16e-002	1.47e+001	3.61e+000	1.58e-001	1.60e-000	4.24e-000	**3.22e-001**
	rank	8	7	2	9	4	**1**	5	6	3
h_{19}	mean	6.43e-001	1.40e-02	2.59e-003	5.47e-002	3.70e-002	6.47e-004	1.70e-004	2.99e-004	**1.06e-005**
	std.	7.73e-001	1.22e-02	6.71e-003	4.20e-002	1.84e-001	3.54e-003	3.80e-001	5.12e-001	**6.16e-004**
	rank	9	8	5	6	7	2	4	3	**1**
h_{20}	mean	3.26e+000	**0.00e+00**	8.95e-014	6.63e-002	8.32e-002	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	std.	6.90e+000	**0.00e+00**	2.53e-013	2.52e-001	1.40e-001	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	rank	5	**1**	2	4	3	**1**	**1**	**1**	**1**
h_{21}	mean	1.65e+002	1.26e+02	5.01e+001	9.38e+001	1.85e+002	1.51e+002	4.70e+001	3.38e+001	**2.30e+001**
	std.	5.43e+001	4.68e+01	4.46e+001	2.34e+001	4.34e+001	5.24e+001	4.96e+000	2.14e+000	**1.81e+001**
	rank	8	6	4	5	9	7	3	2	**1**
h_{22}	mean	9.40e+001	8.06e+01	6.00e+001	8.55e+001	9.48e+001	1.00e+002	4.78e+001	4.83e+001	**3.86e+001**
	std.	2.40e+001	3.89e+01	4.98e+001	3.01e+001	2.12e+001	**0.00e+000**	1.33e+000	2.12e+000	7.22e+001
	rank	7	5	4	6	8	9	2	3	**1**
h_{23}	mean	3.09e+002	3.08e+02	2.20e+002	5.61e+001	3.08e+002	3.01e+002	2.00e+002	2.02e+002	**1.71e+001**
	std.	2.95e+000	2.00e+00	1.34e+002	1.11e+002	3.19e+000	**1.89e+000**	6.72e+005	1.28e+003	3.45e+000
	rank	8	7	5	2	7	6	3	4	**1**
h_{24}	mean	2.74e+002	2.39e+02	1.00e+002	**9.76e+001**	3.39e+002	2.99e+002	2.25e+002	2.29e+002	1.00e+002
	std.	1.06e+002	1.17e+02	**4.78e-014**	1.30e+001	3.92e+000	7.96e+001	4.23e+000	6.82e+000	5.23e+000
	rank	6	5	2	**1**	8	7	3	4	2
h_{25}	mean	4.15e+002	4.00e+02	2.98e+002	4.06e+002	4.21e+002	4.21e+002	2.98e+002	2.96e+002	**2.95e+002**
	std.	2.27e+001	8.92e+00	1.42e+002	1.71e+001	2.39e+001	2.25e+001	8.77e+000	1.73e+001	**2.30e+000**
	rank	6	4	2	5	7	7	2	3	**1**
h_{26}	mean	2.94e+002	3.00e+02	**1.13e+002**	2.59e+002	3.05e+002	2.73e+002	2.94e+002	3.00e+002	2.92e+002
	std.	6.27e+001	0.00e+00	**1.00e+002**	9.26e+001	2.86e+001	6.39e+001	2.45e+001	2.16e+001	6.06e+002
	rank	4	6	**1**	2	7	3	5	6	4
h_{27}	mean	3.90e+002	3.89e+02	**2.78e+002**	3.90e+002	3.89e+002	3.89e+002	3.89e+002	3.90e+002	**3.89e+002**
	std.	1.60e+000	1.96e+00	1.33e+002	2.64e+000	**1.09e+000**	2.11e+000	4.11e+001	6.42e+000	2.38e+002
	rank	3	2	**1**	3	2	2	2	3	**2**
h_{28}	mean	4.67e+002	3.00e+02	**1.20e+002**	2.91e+002	4.58e+002	4.19e+002	2.97e+002	2.98e+002	2.90e+002
	std.	1.48e+002	**0.00e+00**	1.49e+002	4.41e+001	1.48e+002	1.48e+002	4.86e+001	5.78e+001	3.82e+001
	rank	9	6	**1**	3	8	7	4	5	**2**
h_{29}	mean	2.37e+002	2.52e+02	**2.26e+002**	2.40e+002	2.31e+002	2.31e+002	2.27e+002	2.27e+002	**2.26e+002**
	std.	7.28e+000	5.69e+00	2.91e+001	3.45e+000	3.25e+000	**2.58e+000**	5.27e+002	5.27e+001	5.23e+002
	rank	4	6	**1**	5	3	3	2	2	**1**
h_{30}	mean	9.66e+004	3.97e+02	1.04e+003	2.39e+003	1.42e+005	4.50e+002	2.75e+002	2.89e+002	**2.50e+002**
	std.	3.00e+005	**7.46e+00**	2.51e+002	1.31e+003	3.24e+005	5.81e+001	7.90e+001	5.33e+002	1.56e+001
	rank	8	4	6	7	9	5	2	3	**1**

Table 7 (continued)

h$_n$	Criteria	Algorithms								
		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO	AHDEPSO
Sum of rank		177	142	90	146	167	108	78	104	**42**
Average		5.9	4.73	3	4.86	5.56	3.36	2.6	3.46	**1.4**
Overall rank		9	6	3	7	8	4	2	5	**1**

Table 8 Simulation results on TS-2 (30*D*)

h$_n$	Criteria	Algorithms								
		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO	AHDEPSO
h$_1$	mean	5.98e-021	1.50e-14	7.99e+002	1.44e+003	2.80e-023	1.01e-019	7.49e-023	2.78e-022	**4.54e-024**
	std.	1.94e-020	8.24e-15	1.02e+003	2.12e+003	**9.19e-023**	4.67e-019	4.44e-020	7.29e-021	1.38e-020
	rank	5	6	7	8	3	6	2	4	**1**
h$_2$	mean	1.71e-006	**0.00e+00**	5.36e-006	5.06e-005	7.74e-005	1.13e-007	3.13e-008	6.23e-007	**0.00e+000**
	std.	3.07e-006	**0.00e+00**	7.53e-006	1.77e-004	2.35e-004	5.48e-007	6.48e-007	4.28e-006	**0.00e+000**
	rank	6	**1**	5	8	7	4	2	3	**1**
h$_3$	mean	6.58e-028	2.22e-13	8.43e-001	**0.00e+000**	7.49e-028	1.98e-028	**0.00e+000**	7.44e-028	**0.00e+000**
	std.	7.38e-028	2.63e-13	1.82e+000	**0.00e+000**	7.22e-028	2.70e-028	**0.00e+000**	6.12e-027	**0.00e+000**
	rank	4	6	7	**1**	2	5	**1**	3	**1**
h$_4$	mean	2.83e+001	3.74e+01	**2.12e-001**	2.55e+000	5.00e+001	6.47e+000	2.74e+001	2.81e+001	2.62e+001
	std.	3.04e+001	2.91e+01	**7.91e-001**	1.94e+000	2.23e+001	1.61e+001	1.97e+001	8.31e+001	4.17e+001
	rank	7	8	**1**	2	9	3	5	6	4
h$_5$	mean	5.07e+001	4.20e+01	2.23e+001	4.62e+001	3.37e+001	2.69e+001	2.29e+001	2.32e+001	**2.06e+001**
	std.	1.08e+001	1.32e+01	5.68e+000	1.35e+001	6.88e+000	**4.47e+000**	2.45e+002	3.70e+002	5.12e+001
	rank	9	7	2	8	6	5	3	4	**1**
h$_6$	mean	5.62e-004	1.25e-13	1.90e-013	4.91e-008	9.41e-008	**0.00e+000**	**0.00e+000**	1.40e-014	**0.00e+000**
	std.	2.80e-003	3.41e-14	3.42e-013	2.18e-007	3.01e-007	**0.00e+000**	**0.00e+000**	1.30e-012	**0.00e+000**
	rank	7	4	3	6	5	**1**	**1**	2	**1**
h$_7$	mean	7.52e+001	9.51e+01	4.73e+001	6.52e+001	7.53e+001	5.66e+001	**0.00e+000**	1.03e+001	**0.00e+000**
	std.	8.53e+000	2.54e+01	4.05e+000	1.24e+001	9.71e+000	5.02e+000	**0.00e+000**	4.17e+000	**0.00e+000**
	rank	6	8	3	5	7	4	**1**	2	**1**
h$_8$	mean	4.07e+001	4.54e+01	2.15e+001	6.05e+001	3.85e+001	2.85e+001	2.82e+001	2.84e+001	**0.00e+000**
	std.	1.10e+001	1.36e+01	6.61e+000	2.09e+001	1.05e+001	5.98e+000	3.29e+001	5.87e+001	**0.00e+000**
	rank	7	8	2	9	6	5	3	4	**1**
h$_9$	mean	9.60e+000	2.23e-14	5.26e-023	2.35e-001	4.72e-001	1.33e-001	2.28e-025	6.87e-023	**1.78e-026**
	std.	3.05e+001	4.56e-14	9.13e-023	3.53e-001	9.65e-001	4.27e-001	7.77e-023	7.18e-022	**2.16e-024**
	rank	9	5	4	7	6	8	2	3	**1**
h$_{10}$	mean	3.20e+003	3.57e+03	2.14e+003	3.52e+003	2.92e+003	**1.11e+003**	3.10e+003	3.15e+003	2.41e+003
	std.	6.02e+002	7.36e+02	**2.09e+002**	2.86e+002	2.95e+002	2.89e+002	1.71e+004	6.79e+005	7.87e+002
	rank	7	9	2	8	4	**1**	5	6	3
h$_{11}$	mean	9.50e+001	1.30e+01	**5.72e+000**	2.21e+001	2.39e+001	1.87e+001	1.23e+001	1.32e+001	1.17e+001

Table 8 (continued)

hh_n	Criteria	Algorithms								
		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO	AHDEPSO
	std.	5.46e+001	4.99e+00	**3.05e+000**	1.65e+001	2.26e+001	2.13e+001	1.80e+001	7.91e+002	2.21e+002
	rank	9	4	**1**	7	8	6	3	5	2
h_{12}	mean	1.29e+003	7.33e+03	6.39e+004	1.20e+004	9.81e+003	**1.02e+003**	2.30e+003	2.36e+003	1.13e+003
	std.	**3.54e+002**	7.21e+03	1.00e+005	7.55e+003	1.26e+004	4.01e+002	5.17e+002	8.77e+002	3.90e+003
	rank	3	6	9	8	7	**1**	4	5	2
h_{13}	mean	4.61e+002	3.29e+01	5.44e+003	6.40e+003	1.73e+004	5.83e+001	3.27e+001	3.28e+001	**3.24e+001**
	std.	6.26e+002	2.70e+01	3.95e+003	4.70e+003	1.56e+004	3.52e+001	**1.77e+001**	5.77e+001	7.78e+001
	rank	6	4	7	8	9	5	2	3	**1**
h_{14}	mean	1.57e+002	1.11e+01	2.14e+003	7.04e+002	2.91e+001	1.61e+001	1.60e+001	1.61e+001	**1.04e+001**
	std.	7.07e+001	**4.26e+00**	2.89e+003	7.88e+002	9.57e+000	1.13e+001	2.98e+003	7.02e+002	5.12e+002
	rank	6	2	8	7	5	4	3	4	**1**
h_{15}	mean	1.79e+002	**7.91e+00**	2.80e+002	5.40e+002	2.17e+001	6.09e+001	2.16e+001	2.17e+001	**2.14e+001**
	std.	1.22e+002	**3.22e+00**	2.51e+002	7.31e+002	1.46e+001	6.97e+001	4.10e+001	7.92e+001	8.69e+001
	rank	6	**1**	7	8	4	5	3	4	2
h_{16}	mean	7.02e+002	4.47e+02	3.88e+002	2.49e+002	3.56e+002	4.26e+002	2.43e+002	2.45e+002	**2.23e+002**
	std.	3.56e+002	1.76e+02	9.93e+001	1.18e+002	2.05e+002	1.51e+002	4.63e+003	**5.42e+001**	4.15e+003
	rank	9	8	6	4	5	7	2	3	**1**
h_{17}	mean	1.66e+002	6.17e+01	**4.10e+001**	5.95e+001	8.29e+001	4.51e+001	4.20e+001	4.22e+001	4.18e+001
	std.	1.14e+002	3.89e+01	2.43e+001	**1.48e+001**	8.42E+001	4.87e+001	3.23e+001	3.76e+001	2.65e+001
	rank	9	7	**1**	6	8	5	3	4	2
h_{18}	mean	1.03e+002	9.96e+01	1.17e+004	3.50e+004	3.23e+002	3.55e+001	3.21e+001	3.23e+001	**3.20e+001**
	std.	4.78e+001	1.11e+02	8.80e+003	1.35e+004	3.94e+002	**4.36e+001**	6.98e+003	1.68e+004	4.65e+003
	rank	6	5	8	9	7	4	2	3	**1**
h_{19}	mean	8.88e+001	6.35e+00	1.16e+003	1.46e+003	1.17e+001	2.68e+001	1.70e+000	2.99e+000	**1.06e+000**
	std.	5.06e+001	**2.27e+00**	1.21e+003	1.60e+003	5.20e+000	2.32e+001	3.80e+000	4.79e+000	2.72e+000
	rank	7	4	8	9	5	6	2	3	**1**
h_{20}	mean	2.88e+002	1.16e+02	1.22e+002	8.33e+001	1.10e+002	7.87e+001	**0.00e+000**	5.35e+001	**0.00e+000**
	std.	1.80e+002	6.08e+01	4.99e+001	4.72e+001	1.27e+002	5.92e+001	**0.00e+000**	8.92e+001	**0.00e+000**
	rank	8	6	7	4	5	3	**1**	2	**1**
h_{21}	mean	2.41e+002	2.42e+02	1.77e+002	2.39e+002	2.38e+002	2.30e+002	1.45e+002	1.78e+002	**4.33e+001**
	std.	1.35e+001	1.10e+01	6.49e+001	1.28e+001	**7.98e+000**	8.01e+000	3.45e+001	2.44e+002	2.84e+002
	rank	8	9	3	7	6	5	2	4	**1**
h_{22}	mean	3.78e+002	**1.00e+02**	**1.00e+002**	**1.00e+002**	2.62e+003	**1.00e+002**	**1.00e+002**	**1.00e+002**	**1.00e+002**
	std.	9.67e+002	4.49e-13	2.00e-012	**0.00e+000**	**1.32e+003**	**0.00e+000**	2.95e+000	3.87e+003	7.22e+003
	rank	2	**1**	**1**	**1**	3	**1**	**1**	**1**	**1**
h_{23}	mean	3.95e+002	3.92e+02	**3.00e+002**	3.85e+002	3.90e+002	3.70e+002	**3.00e+002**	3.02e+002	**3.00e+002**
	std.	1.18e+001	**2.94e+00**	1.40e+002	1.36e+001	9.65e+000	8.21e+000	5.62e+002	4.88e+003	5.92e+003
	rank	7	6	**1**	4	5	3	**1**	2	**1**
h_{24}	mean	4.69e+002	4.70e+02	2.49e+002	4.49e+002	4.73e+002	4.49e+002	2.31e+002	2.35e+002	**2.09e+002**
	std.	**1.03e+001**	1.35e+01	1.32e+002	1.40e+001	1.37e+001	7.44e+002	1.87e+002	3.89e+002	5.35e+002
	rank	6	7	4	5	8	5	2	3	**1**
h_{25}	mean	3.87e+002	3.87e+02	3.88e+002	3.87e+002	3.87e+002	3.88e+002	3.87e+002	3.88e+002	**3.85e+002**
	std.	**1.24e+000**	3.79e-02	2.19e+000	2.19e+000	1.39e+000	3.04e+000	1.78e+000	2.73e+001	4.87e+002
	rank	2	2	3	2	2	3	2	3	**1**
h_{26}	mean	1.50e+003	1.07e+03	2.44e+002	9.33e+002	1.42e+003	4.25e+002	2.44e+002	2.78e+002	**2.43e+002**

Table 8 (continued)

h_n	Criteria	Algorithms								
		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO	AHDEPSO
	std.	3.82e+002	5.37e+02	5.06e+001	4.43e+002	**1.11e+002**	4.02e+002	3.95e+003	3.89e+003	4.95e+003
	rank	8	6	2	5	**7**	4	2	3	**1**
h_{27}	mean	5.11e+002	4.98e+02	5.98e+002	5.21e+002	5.06e+002	5.03e+002	4.89e+002	4.90e+002	**4.19e+002**
	std.	1.20e+001	7.27e+00	1.15e+001	8.99e+000	**5.77e+000**	8.02e+000	4.11e+001	5.85e+003	5.55e+002
	rank	**7**	4	9	8	6	5	2	3	**1**
h_{28}	mean	3.29e+002	3.14e+02	**3.00e+002**	3.16e+002	3.53e+002	3.14e+002	**3.00e+002**	3.13e+002	**3.00e+002**
	std.	5.51e+001	3.59e+01	**7.82e-013**	3.86e+001	6.36e+001	3.66e+001	4.78e+001	4.98e+001	5.76e+001
	rank	5	3	**1**	4	6	3	**1**	2	**1**
h_{29}	mean	6.39e+002	4.40e+02	6.36e+002	4.95e+002	4.62e+002	4.39e+002	4.39e+002	4.40e+002	**4.36e+002**
	std.	1.23e+002	5.17e+01	3.44e+001	**3.08e+001**	6.31e+001	3.29e+001	3.86e+001	5.56e+001	5.88e+002
	rank	7	3	6	5	4	2	2	3	**1**
h_{30}	mean	2.30e+003	2.27e+03	9.07e+003	3.71e+003	2.14e+003	**5.63e+002**	2.13e+003	2.19e+003	2.10e+003
	std.	3.26e+002	**1.94e+02**	1.95e+003	6.67e+002	2.13e+002	9.81e+001	5.98e+002	4.54e+002	4.14e+002
	rank	7	6	9	8	4	**1**	3	5	2
Sum of rank		195	152	137	181	169	120	68	102	**40**
Average		6.5	5.06	4.56	6.03	5.63	4	2.26	3.4	**1.33**
Overall rank		9	6	5	8	7	4	2	3	**1**

summarized for 10*D*TS-2 cases (i). Unimodal function (h_1-h_3): proposed ADE, APSO and AHDEPSO exhibits the best performance in all three functions (h_1-h_3) with similar for h_1 (on MPEDE, EFADE, HPSODE & PSOJADE), h_2 (on EFADE) and h_3 (on MPEDE, EFADE, CSPSO, HPSODE & PSOJADE). (ii). Multimodal function (h_4-h_{10}): proposed AHDEPSO outperformed for all functions and equally for h_4 (on EFADE & PSOJADE). h_5 (on HEPSO), h_6 (on MPEDE, CSPSO, HPSODE & PSOJADE) and h_9 (on EFADE). Suggested component ADE execute better result in four functions (h_4, h_6, h_7 and h_8) whereas APSO gives best result for f_4 as well as for other functions they perform similar or marginally inferior. (iii). Hybrid function (h_{11}-h_{20}): proposed AHDEPSO execute better result for five functions (h_{13}, h_{14}, h_{17}, h_{19} and h_{20}) similar for h_{15} (on PSOJADE) and marginally inferior with HEPSO (for h_{11}) and EFADE (for h_{12}, h_{16} and h_{20}). At the same time, ADE and APSO obtained better results for h_{20} and marginally similar/inferior for the rest function. (iv). Composition function (h_{21}-h_{30}): proposed AHDEPSO exhibits the best performance on six functions (h_{21}, h_{22}, h_{23}, h_{25}, h_{29} and h_{30}) and equally with HEPSO (for h_{29}) as well as marginally inferior with HEPSO (for h_{26}, h_{27} and h_{28}) and CSPSO (for h_{24}). Meanwhile, ADE and APSO perform marginally better/similar or slightly inferior in few functions with other algorithms.

Further, as per the experimental results shown in Table 8 the following comparison results are précised as follows for 30*D*TS-2 cases (i). Unimodal function (h_1-h_3): proposed AHDEPSO exhibits the best performance in all three functions (h_1-h_3) similar with EFADE (for h_2) and CSPSO (for h_3). Suggested ADE shown the best performance on h_3 function and perform equal or marginally inferior with other algorithms. Whereas, APSO perform slightly inferior in all three functions with other algorithms. (ii). Multimodal function (h_4-h_{10}): proposed AHDEPSO execute better result for five functions (h_5, h_6, h_7, h_8 and h_9) and equally with HEPSO (for h_4) and PSOJADE (for h_6 and h_{10}). Suggested ADE execute better result in two functions (h_6 and h_7) and marginally inferior with others. Whereas, APSO execute similar result or marginally inferior for other functions. (iii). Hybrid function (h_{11}-h_{20}): proposed AHDEPSO execute better result for six functions (h_{13}, h_{14}, h_{16}, h_{18}, h_{19} and h_{20}) marginally inferior with HEPSO (for h_{11} and h_{17}) and PSOJADE (for h_{12}) EFADE (for h_{15}). At the same time, ADE obtained better results for h_{20} and APSO marginally similar/inferior for the rest. (iv). Composition function (h_{21}-h_{30}): proposed AHDEPSO execute the better performance on nine functions (h_{21}, h_{22}, h_{23}, h_{24}, h_{25}, h_{26}, h_{27}, h_{28} and h_{29}) and equally with EFADE, HEPSO, CSPSO and PSOJADE (for h_{22}), HEPSO (for h_{23} and h_{28}) as well as marginally inferior with PSOJADE (for h_{30}). Meanwhile, ADE perform better in three functions (h_{22}, h_{23} and h_{28}) also APSO execute better result in one function (for h_{22}) and marginally better/similar or slightly inferior in few functions with other algorithms.

Table 9 Simulation results on TS-2 (50D)

h_n	Criteria	Algorithms								
		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO	AHDEPSO
h_1	mean	6.85e-017	8.41e-01	1.46e+003	9.26e+002	7.74e+003	2.78e-016	7.49e-028	2.78e-024	**1.77e-029**
	std.	1.34e-016	1.68e+00	2.18e+003	1.61e+003	9.71e+003	1.09e-015	2.24e-026	3.10e-023	**2.38e-025**
	rank	4	6	8	7	9	5	2	3	**1**
h_2	mean	5.21e+09	**0.00e+00**	1.82e+09	6.31e+09	7.35e+008	2.69e-005	2.33e-011	6.74e-009	**0.00e+000**
	std.	2.85e+08	**0.00e+00**	9.96e+09	3.45e+09	4.02e+009	5.51e-005	4.18e-012	4.27e-010	**0.00e+000**
	rank	7	**1**	6	8	5	4	2	3	**1**
h_3	mean	8.00e-027	7.75e-05	7.00e+003	2.18e-027	4.93e-020	2.61e-014	**0.00e+000**	9.25e-027	**0.00e+000**
	std.	7.40e-027	2.28e-04	8.19e+003	1.85e-027	1.10e-019	7.79e-014	**0.00e+000**	7.81e-028	**0.00e+000**
	rank	3	7	8	4	5	6	**1**	2	**1**
h_4	mean	5.32e+001	4.20e+01	4.67e+000	1.58e+001	4.38e+001	2.70e+001	3.74e+000	1.45e+001	**3.69e+000**
	std.	4.53e+001	3.03e+01	**1.73e+001**	3.30e+001	3.60e+001	4.56e+001	2.45e+001	4.31e+001	5.88e+001
	rank	9	7	3	5	8	6	2	4	**1**
h_5	mean	1.10e+002	9.56e+01	1.06e+002	1.05e+002	**8.96e+001**	1.03e+002	1.02e+001	1.10e+001	8.66e+001
	std.	2.97e+001	2.61e+01	1.94e+001	4.21e+001	3.20e+001	**1.87e+001**	2.34e+002	2.82e+002	7.32e+001
	rank	4	3	7	6	**1**	5	2	4	2
h_6	mean	4.71e-002	1.83e-13	3.28e-008	3.05e-006	2.93e-003	**0.00e+000**	**0.00e+000**	7.48e-015	**0.00e+000**
	std.	1.40e-001	5.61e-14	9.08e-008	5.77e-006	1.11e-002	**0.00e+000**	**0.00e+000**	5.10e-013	**0.00e+000**
	rank	7	3	4	5	6	**1**	**1**	2	**1**
h_7	mean	1.73e+002	1.77e+02	1.34e+002	1.39e+002	1.92e+002	1.31e+002	**0.00e+000**	1.13e+002	**0.00e+000**
	std.	2.61e+001	7.00e+01	1.63e+001	4.66e+001	2.50e+001	1.68e+001	**0.00e+000**	2.88e+003	**0.00e+000**
	rank	6	8	4	5	7	3	**1**	2	**1**
h_8	mean	1.15e+002	9.09e+01	1.07e+002	9.25e+001	8.94e+001	9.55e+001	9.02e+001	9.24e+001	**0.00e+000**
	std.	2.32e+001	1.94e+01	2.14e+001	2.87e+001	2.99e+001	2.11e+001	7.19e+001	8.45e+001	**0.00e+000**
	rank	9	4	8	6	2	7	3	5	**1**
h_9	mean	4.48e+002	5.27e-03	4.20e-002	4.08e+000	8.91e+000	3.07e+000	2.28e-010	6.87e-012	3.78e-015
	std.	3.76e+002	2.13e-02	8.81e-002	3.25e+000	2.06e+001	4.55e+001	7.78e-011	7.65e-013	4.14e-016
	rank	9	4	5	7	8	6	3	2	**1**
h_{10}	mean	5.84e+003	8.02e+03	4.52e+003	1.00e+004	7.35e+003	3.08e+003	2.58e+003	2.65e+003	**2.27e+003**
	std.	7.74e+002	1.61e+03	4.51e+002	**2.66e+002**	4.76e+002	5.94e+002	3.78e+004	5.19e+005	3.18e+002
	rank	6	8	5	9	7	4	2	3	**1**
h_{11}	mean	1.91e+002	4.66e+01	**4.55e+000**	5.20e+001	8.23e+001	1.14e+002	4.13e+001	5.38e+001	7.58e+000
	std.	5.00e+001	7.34e+00	**1.99e+000**	1.48e+001	3.11e+001	4.34e+001	3.84e+001	6.92e+002	6.25e+000
	rank	9	4	**1**	5	7	8	3	6	2
h_{12}	mean	9.45e+003	4.83e+04	1.75e+006	7.86e+004	4.20e+004	**2.89e+003**	4.38e+003	5.16e+003	3.65e+003
	std.	1.91e+004	2.97e+04	1.55e+006	8.19e+004	2.46e+004	1.22e+003	6.78e+002	**6.77e+002**	4.91e+003
	rank	5	7	9	8	6	**1**	3	4	2
h_{13}	mean	6.98e+002	2.71e+03	1.99e+003	9.48e+002	7.32e+003	**1.91e+002**	6.98e+002	8.28e+002	6.24e+002
	std.	5.38e+002	2.51e+03	1.27e+003	1.04e+003	9.70e+003	**2.45e+002**	5.87e+002	4.79e+003	3.78e+003
	rank	3	7	6	5	8	**1**	3	4	2
h_{14}	mean	3.85e+002	**4.42e+01**	2.04e+005	1.79e+004	2.77e+002	1.77e+002	1.77e+002	1.88e+002	1.74e+002
	std.	1.13e+002	**1.36e+01**	1.72e+005	1.57e+004	1.98e+002	2.35e+002	3.98e+003	6.78e+002	4.22e+003
	rank	6	**1**	8	7	5	3	3	4	2
h_{15}	mean	4.89e+002	**1.20e+02**	4.89e+003	3.68e+003	8.31e+003	2.38e+002	3.55e+002	2.87e+002	2.27e+002
	std.	1.89e+002	2.36e+02	4.90e+003	2.77e+003	8.53e+003	1.96e+002	7.10e+001	**3.92e+001**	8.69e+001
	rank	6	**1**	7	8	9	3	5	4	2

Table 9 (continued)

h_n	Criteria	Algorithms								
		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO	AHDEPSO
h_{16}	mean	1.31e+003	8.99e+02	1.11e+003	8.34e+002	1.19e+003	8.36e+002	8.24e+002	8.29e+002	**7.85e+002**
	std.	4.18e+002	**2.06e+02**	2.25e+002	3.16e+002	2.83e+002	2.15e+002	6.62e+003	3.82e+003	5.74e+002
	rank	9	6	7	4	8	5	2	3	**1**
h_{17}	mean	8.71e+002	6.02e+02	9.61e+002	7.44e+002	5.67e+002	5.79e+002	4.10e+002	4.18e+002	**4.08e+002**
	std.	2.92e+002	1.38e+02	**1.26e+002**	1.47e+002	1.77e+002	2.26e+002	3.23e+002	3.76e+002	**2.65e+002**
	rank	8	6	9	7	4	5	2	3	**1**
h_{18}	mean	2.36e+002	3.02e+03	3.22e+005	5.94e+005	6.86e+003	2.36e+002	2.21e+002	2.22e+002	**2.20e+002**
	std.	1.45e+002	3.19e+03	2.42e+005	3.91e+005	4.61e+003	**4.09e+001**	5.21e+003	2.61e+003	3.65e+003
	rank	4	5	7	8	6	4	2	3	**1**
h_{19}	mean	2.05e+002	2.17e+01	1.09e+004	1.10e+004	7.77e+003	1.03e+002	2.70e+001	2.78e+001	**2.06e+001**
	std.	6.19e+001	**2.47e+01**	3.61e+003	2.54e+003	9.31e+003	5.08e+001	3.91e+002	3.25e+002	2.72e+002
	rank	6	2	8	9	7	5	3	4	**1**
h_{20}	mean	8.78e+002	3.95e+02	6.20e+002	5.30e+002	5.31e+002	3.41e+002	**0.00e+000**	3.35e+002	**0.00e+000**
	std.	3.92e+002	1.67e+02	1.76e+002	9.35e+001	1.99e+002	1.24e+002	**0.00e+000**	5.12e+002	**0.00e+000**
	rank	8	4	7	5	6	3	**1**	2	**1**
h_{21}	mean	3.02e+002	2.91e+02	3.01e+002	3.02e+002	2.84e+002	2.73e+002	2.65e+002	2.70e+002	2.**56e+002**
	std.	2.12e+001	2.10e+01	5.42e+001	4.56e+001	2.77e+001	**6.47e+000**	3.22e+001	2.46e+002	2.87e+002
	rank	8	6	7	8	5	4	2	3	**1**
h_{22}	mean	6.60e+003	6.44e+03	2.79e+003	**4.04e+002**	7.91e+003	1.94e+003	2.41e+003	2.46e+003	1.89e+003
	std.	1.01e+003	3.78e+03	2.74e+003	1.66e+003	4.23e+002	2.02e+003	**3.89e+002**	4.47e+002	7.32e+003
	rank	8	7	6	**1**	9	3	4	5	2
h_{23}	mean	5.57e+002	5.18e+02	5.90e+002	5.43e+002	5.15e+002	4.91e+002	4.78e+002	4.89e+002	**4.60e+002**
	std.	2.80e+001	2.67e+01	6.17e+001	4.68e+001	1.73e+001	**1.70e+001**	5.62e+002	6.58e+003	6.78e+003
	rank	8	6	9	7	5	4	2	3	**1**
h_{24}	mean	6.10e+002	5.84e+02	8.31e+002	6.10 + 002	5.90e+002	5.64e+002	5.11e+002	5.22e+002	**4.19e+002**
	std.	3.06e+001	1.78e+01	6.65e+001	6.48e+001	**1.76e+001**	1.54e+001	6.44e+002	6.45e+002	5.87e+002
	rank	7	5	8	7	6	4	2	3	**1**
h_{25}	mean	5.34e+002	4.89e+02	4.80e+002	5.43e+002	5.24e+002	5.71e+002	4.80e+002	4.84e+002	**4.29e+002**
	std.	4.25e+001	1.92e+01	3.65e+001	3.95e+001	4.09e+001	2.94e+001	1.78e+000	2.73e+001	5.6e+002
	rank	6	4	2	6	5	7	2	3	**1**
h_{26}	mean	2.45e+003	2.08e+03	1.12e+003	2.26e+003	2.07e+003	4.34e+002	2.34e+002	2.48e+002	**2.13e+002**
	std.	3.25e+002	**2.03e+02**	1.10e+003	4.20e+002	2.31e+002	5.21e+002	1.36e+003	5.19e+003	2.68e+003
	rank	9	7	5	8	6	4	2	3	**1**
h_{27}	mean	7.22e+002	5.29e+02	1.00e+003	7.19e+002	6.44e+002	6.18e+002	4.99e+002	5.10e+002	**4.65e+002**
	std.	9.33e+001	**1.07e+01**	6.26e+001	6.51e+001	5.54e+001	5.14e+001	7.56e+001	6.15e+003	6.23e+002
	rank	8	4	9	7	6	5	2	3	**1**
h_{28}	mean	4.93e+002	**4.59e+02**	4.93e+002	4.84e+002	4.93e+002	4.83e+002	4.62e+002	4.64e+002	4.60e+002
	std.	1.53e+001	**5.31e-13**	1.50e+001	2.09e+001	1.95e+001	1.87e+001	5.92e+001	3.93e+001	3.77e+001
	rank	7	**1**	7	6	7	5	3	4	2
h_{29}	mean	1.07e+003	4.84e+02	1.08e+003	7.66e+002	5.08e+002	**4.61e+002**	4.84e+002	4.82e+002	**4.61e+002**
	std.	3.28e+002	1.13e+02	1.22e+002	1.67e+002	1.19e+002	1.22e+002	**1.86e+001**	3.92e+001	5.21e+002
	rank	6	3	7	5	4	**1**	3	2	**1**
h_{30}	mean	7.14e+005	6.11e+05	7.05e+006	8.01e+005	6.43e+005	6.08e+005	5.98e+005	6.06e+005	**5.83e+005**
	std.	1.10e+005	5.22e+04	1.10e+006	6.38e+004	7.63e+004	3.07e+004	**3.97e+003**	5.87e+005	3.84e+005
	rank	7	5	9	8	6	4	2	3	**1**

Table 9 (continued)

h_n	Criteria	Algorithms								
		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO	AHDEPSO
Sum of rank		202	142	196	186	183	126	70	99	**38**
Average		6.73	4.73	6.55	6.2	6.1	4.2	2.33	3.3	**1.26**
Overall rank		9	5	8	7	6	4	2	3	**1**

Then, according to the experimental results given in Table 9 the following comparison results are précised as follows for 50*D*TS-2 cases (i). Unimodal function (h_1-h_3): proposed AHDEPSO exhibits the best performance in all three functions (h_1-h_3) and similar with EFADE (for h_2). Suggested ADE execute the best performance on h_3 function and gives equal or marginally inferior result with other algorithms. Meanwhile, APSO perform slightly inferior in all three functions with other algorithms. (ii). Multimodal function (h_4-h_{10}): proposed AHDEPSO execute better result for six functions (h_4, h_6, h_7, h_8, h_9 and h_{10}) and equally with PSOJADE (for h_6) as well as marginally inferior with HPSODE (for h_5). ADE executes better result in two functions (for h_6 and h_7). Moreover, APSO execute similar or marginally inferior for

Table 10 Statistical comparisons of proposed Vs other algorithms for TS-2 (10*D*)

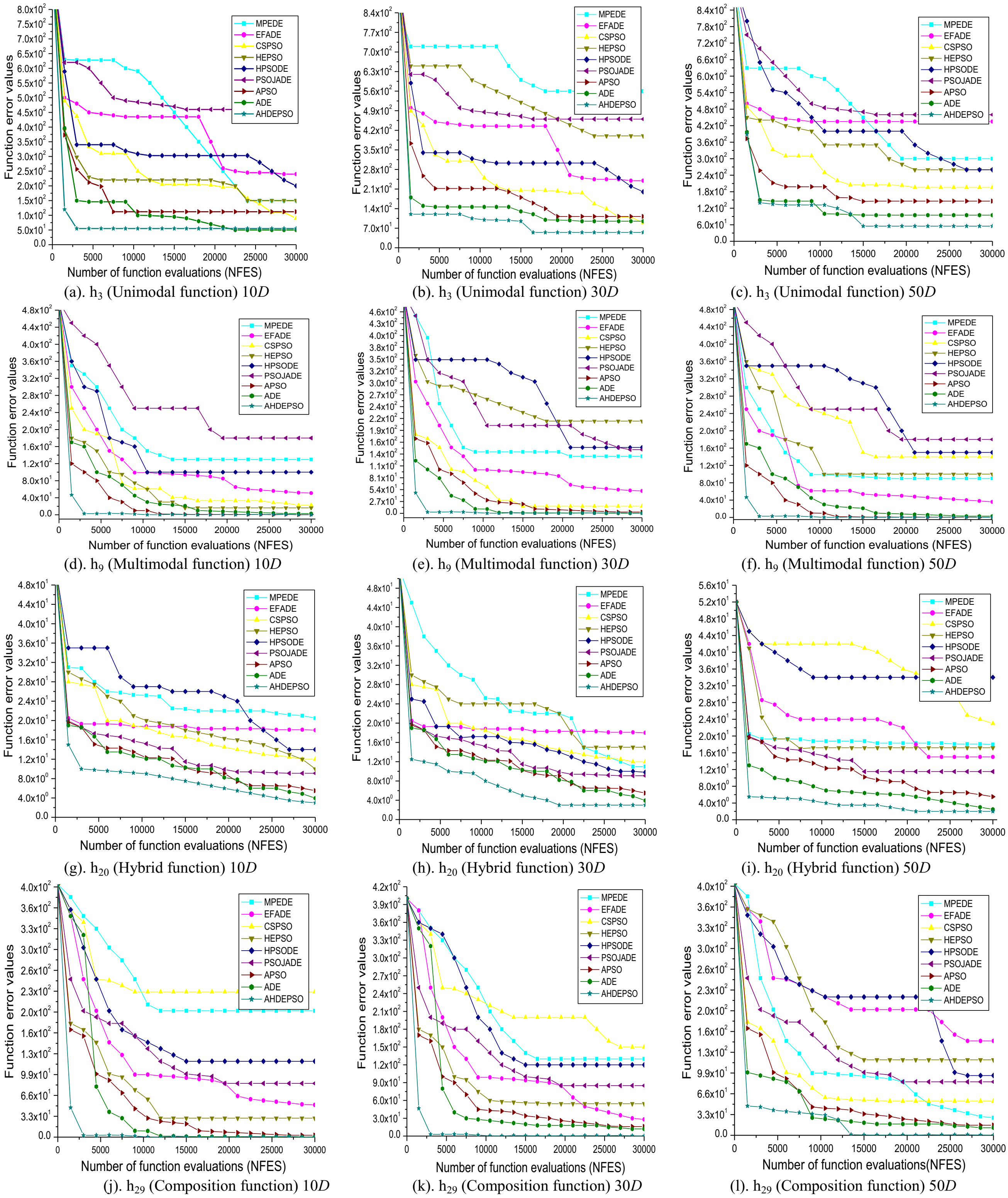
Vs	Criteria	DE variants		PSO variants		Hybrid variants		Praposed algorithms	
		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	APSO	AHDEPSO
ADE	better	25	22	17	24	24	16	23	0
	equal	3	5	2	1	4	6	6	8
	worst	2	3	11	5	2	8	1	22
	R^+	329	367	330	313	377	350	293	323
	R^-	136	98	135	152	88	135	172	142
	p value	5.24e-09	5.14e-10	4.32e-10	6.92e-07	5.15e-10	5.55e-06	6.2e-11	4.41e-10
	t-test	a+	a+	a+	a	a	a	a+	a+
	decision	+	+	+	+	+	+	+	+
Vs		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	AHDEPSO
APSO	better	23	20	15	20	23	22	1	0
	equal	4	6	1	2	2	4	6	5
	worst	3	4	14	8	5	4	23	25
	R^+	345	355	329	465	323	367	377	342
	R^-	120	130	136	79	142	98	88	123
	p value	6.25e-10	4.46e-10	5.74e-10	5.31e-10	5.36e-10	8.32e-10	5.80e-12	6.02e-11
	t-test	a	a	a+	a	a	a+	a+	a+
	decision	+	+	+	+	+	+	+	+
Vs		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO
AHDEPSO	better	27	22	22	26	23	20	22	25
	equal	3	6	4	2	5	7	8	5
	worst	0	2	4	2	2	3	0	0
	R^+	335	305	312	323	382	305	300	350
	R^-	130	160	153	142	83	160	165	115
	p value	3.52e-10	4.52e-10	1.15e-10	6.47e-10	4.71e-10	4.76e-08	3.50e-12	5.28e-11
	t-test	a+	a+	a+	a	a	a	a+	a+
	decision	+	+	+	+	+	+	+	+

Table 11 Statistical comparisons of proposed Vs other algorithms for TS-2 (30*D*)

	Criteria	DE variants		PSO variants		Hybrid variants		Praposed algorithms	
Vs		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	APSO	AHDEPSO
ADE	better	26	23	16	26	23	18	23	0
	equal	3	5	3	2	5	4	6	8
	worst	1	2	11	2	2	8	1	22
	R^+	329	367	330	313	377	313	377	323
	R^-	136	98	135	152	88	152	88	142
	p value	6.26e-09	7.17e-10	4.36e-10	6.69e-07	5.61e-10	5.55e-06	6.25e-11	5.01e-12
	t-test	a+	a	a+	a+	a	a	a+	a+
	decision	+	+	+	+	+	+	+	+
Vs		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	AHDEPSO
APSO	better	22	20	14	21	21	23	1	0
	equal	4	5	1	2	5	3	6	5
	worst	4	5	15	7	4	4	23	25
	R^+	350	293	330	313	377	329	377	350
	R^-	135	172	135	152	88	136	88	135
	p value	6.29e-10	4.60e-10	5.70e-10	5.0e-10	5.46e-10	8.21e-10	4.81e-11	1.2e-13
	t-test	a	a+	a+	a	a	a	a+	a+
	decision	+	+	+	+	+	+	+	+
Vs		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO
AHDEPSO	better	28	22	23	27	24	21	21	24
	equal	2	5	3	2	4	6	9	6
	worst	0	3	4	1	2	3	0	0
	R^+	313	377	329	367	382	313	377	350
	R^-	152	88	136	98	83	152	88	115
	p value	3.55e-10	4.75e-10	1.50e-10	6.04e-10	4.10e-10	4.60e-08	5.5e-11	5.8e-12
	t-test	a+	a+	a	a	a	a	a+	a+
	decision	+	+	+	+	+	+	+	+

Table 12 Statistical comparisons of proposed Vs other algorithms for TS-2 (50*D*)

	Criteria	DE variants		PSO variants		Hybrid variants		Praposed algorithms	
Vs		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	APSO	AHDEPSO
ADE	better	29	24	28	29	28	23	28	0
	equal	1	1	1	0	0	2	0	4
	worst	0	5	1	1	2	5	2	26
	R^+	329	367	330	313	377	350	293	323
	R^-	136	98	135	152	88	135	172	142
	p value	3.23e-09	5.13e-10	4.33e-10	6.19e-07	5.11e-10	5.15e-06	6.02e-11	6.19e-12
	t-test	a+	a	a+	a	a	a	a+	a+
	decision	+	+	+	+	+	+	+	+
Vs		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	AHDEPSO
APSO	better	29	22	27	29	28	24	2	0
	equal	0	0	0	0	0	0	0	0
	worst	1	8	3	1	2	6	28	30
	R^+	345	355	329	465	323	367	377	342
	R^-	120	130	136	79	142	98	88	123
	p value	6.22e-10	5.76e-10	5.73e-10	5.31e-10	5.36e-10	8.23e-10	5.58e-12	6.20e-12
	t-test	a+	a	a+	a	a+	a	a+	a+
	decision	+	+	+	+	+	+	+	+
Vs		MPEDE	EFADE	HEPSO	CSPSO	HPSODE	PSOJADE	ADE	APSO
AHDEPSO	better	30	25	29	29	29	25	26	30
	equal	0	2	0	0	0	2	4	0
	worst	0	3	1	1	1	3	0	0
	R^+	335	305	312	323	382	305	300	350
	R^-	130	160	153	142	83	160	165	115
	p value	3.59e-10	5.50e-10	1.15e-10	6.41e-10	4.11e-10	4.96e-08	5.59e-11	5.8e-12
	t-test	a+	a+	a	a	a	a	a+	a+
	decision	+	+	+	+	+	+	+	+



(a). h_3 (Unimodal function) 10*D*

(b). h_3 (Unimodal function) 30*D*

(c). h_3 (Unimodal function) 50*D*

(d). h_9 (Multimodal function) 10*D*

(e). h_9 (Multimodal function) 30*D*

(f). h_9 (Multimodal function) 50*D*

(g). h_{20} (Hybrid function) 10*D*

(h). h_{20} (Hybrid function) 30*D*

(i). h_{20} (Hybrid function) 50*D*

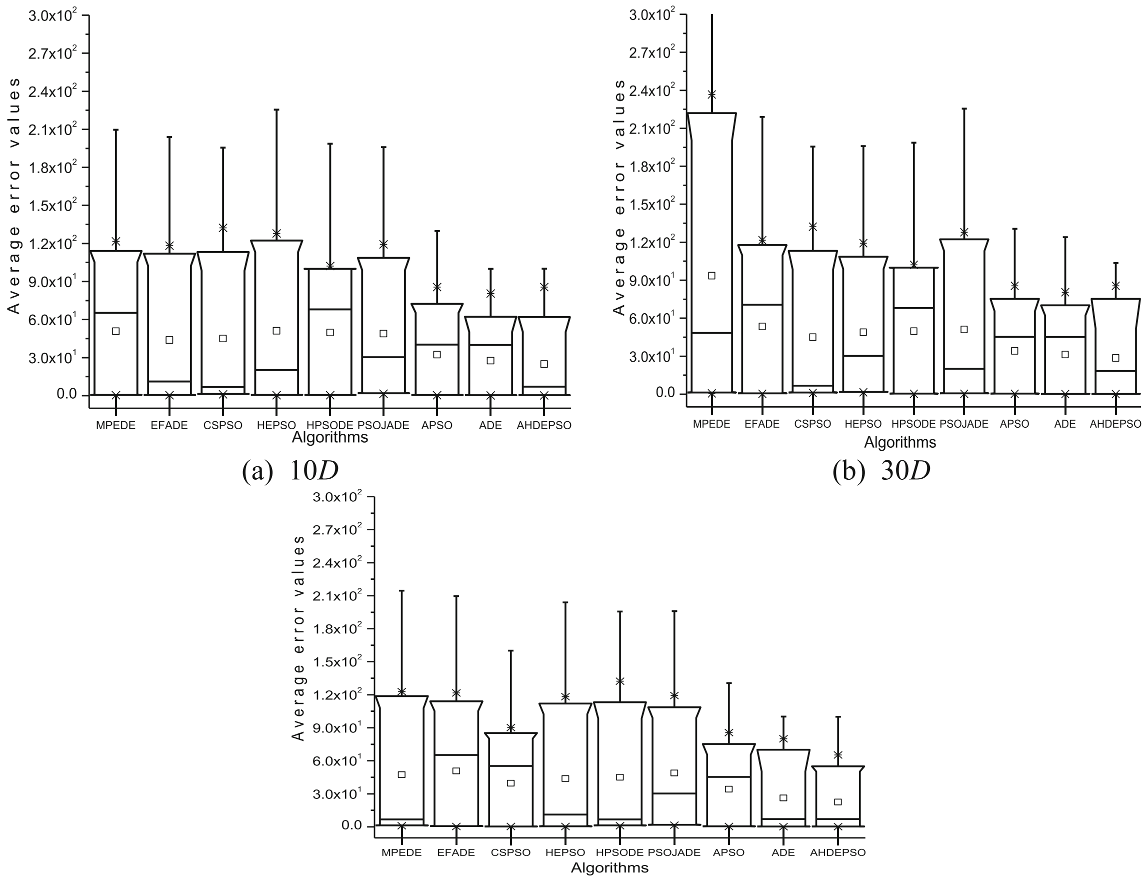
(j). h_{29} (Composition function) 10*D*

(k). h_{29} (Composition function) 30*D*

(l). h_{29} (Composition function) 50*D*

Fig. 9 **a-l** Convergence curves for TS-2 (10*D*, 30*D* & 50*D*)

other functions. (iii). Hybrid function (h_{11}-h_{20}): proposed AHDEPSO outperformed in five functions (h_{16}, h_{17}, h_{18} h_{19} and h_{20}) and marginally inferior with HEPSO (for h_{11}), PSOJADE (for h_{12} and h_{13}) and EFADE (for h_{14} and h_{15}). Meanwhile, ADE obtained better results only for h_{20} and at the same time APSO is marginally similar/inferior for the rest.



(a) 10D

(b) 30D

(c) 50D

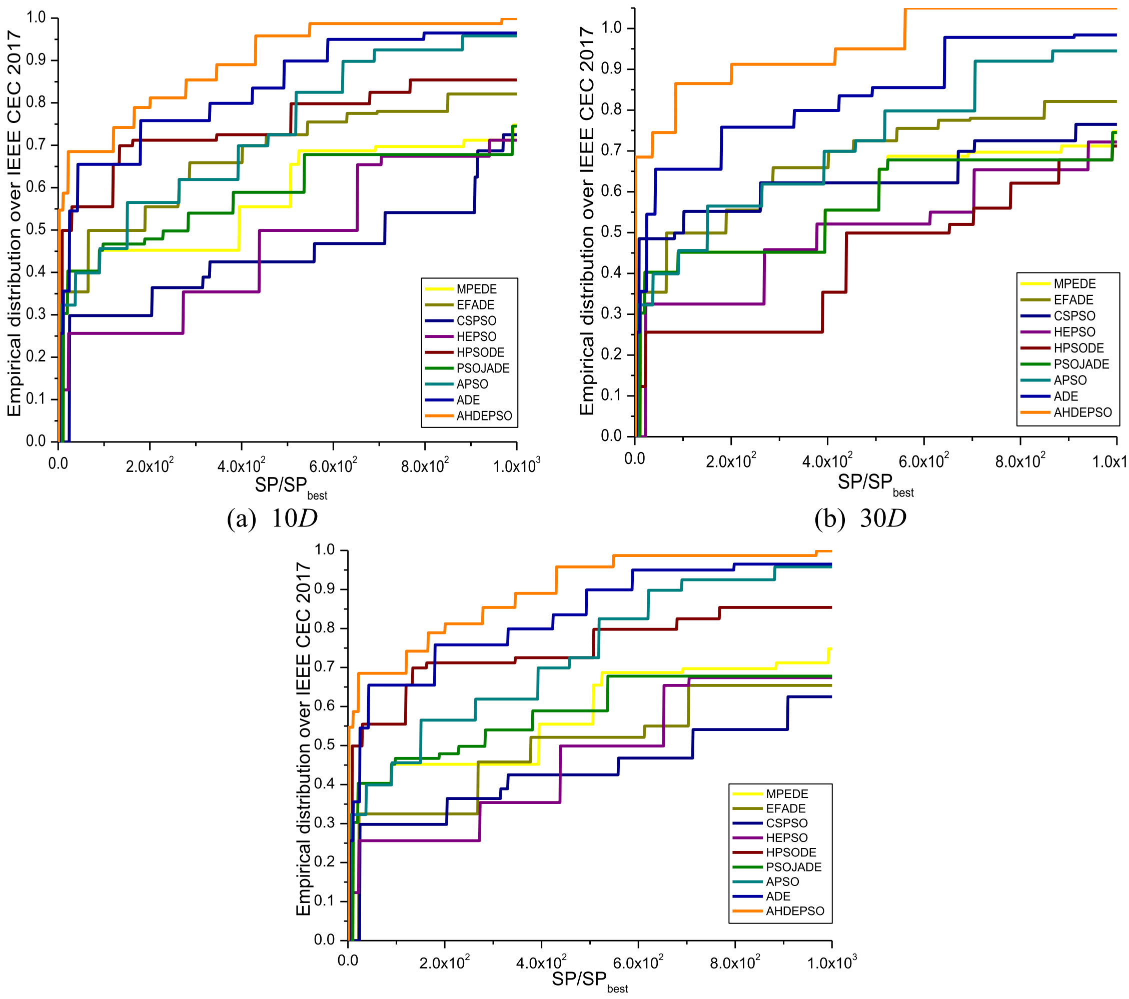
Fig. 10 a-c Average function error values of TS-2 (10D, 30D & 50D)

(iv). Composition function (h_{21}-h_{30}): proposed AHDEPSO exhibit the best performance on eight functions (h_{21}, h_{23}, h_{24}, h_{25}, h_{26}, h_{27}, h_{29} and h_{30}) and marginally inferior with CSPSO (for h_{22}) and EFADE (for h_{28}). Meanwhile, ADE and APSO perform marginally better/similar or slightly inferior in few functions with other algorithms

Moreover, all algorithms are individually ranked in Tables 7, 8 and 9 based on mean error values. From these tables, it is concluded that AHDEPSO, ADE and APSO ranked 1st, 2nd and 5th (in case of 10D TS-2) 1st, 2nd and 3rd (in case of 30D & 50D TS-2) individually. Also, average and overall rank of proposed algorithms Vs others are presented in Tables 7, 8 and 9. It is clear that (from ranking) performances of proposed algorithms are superior to others. Ultimately, proposed ADE, APSO and AHDEPSO produces less std. for most of the functions on TS-2 which defines their stability. Furthermore, superiority of proposed ADE, APSO and AHDEPSO are statistically validated over others through one-tailed$t$-test (with 98 degree of freedom (df) at 5% significance level) and WSR test (at 5% significance level). The results of t-test and WSR test on TS-2 are reported in Table 10(10D TS-2), Table 11(30D TS-2) and Table 12(50D TS-2). From these tables it can be clearly seen that proposed algorithms has both 'a' or 'a^{+}' sign (in case of t-test) and performs better or equally (in case WSR test) in maximum circumstances. As well as, the reported less p-values in Tables 10, 11 and 12 concluded reliable results for the majority of runs of the proposed algorithms in TS-2 case.

The convergence speed of proposed ADE, APSO and AHDEPSO algorithms with others are analyzed on 10, 30 & 50D TS-2. For this one function from each category of TS-2 (h_3, h_9, h_{20} & h_{29}) is taken. The convergence graph for all such function can visualize on Fig. 9a-l for TS-2. It can be observed that from these figures ADE, APSO and AHDEPSO makes better convergence than other algorithms. Apart from this, average error values of proposed algorithms for 10, 30 & 50DTS-2 have been analyzed through box plots. The respective box plots figures presented in Fig. 10a-c. From these figures, it is clearly visible that average error value of proposed algorithms is better than others.

Furthermore, to check the performance of the proposed algorithms ECD (empirical cumulative distribution) test [142] is applied on the set of 10, 30, & 50D TS-2.



(a) 10*D*

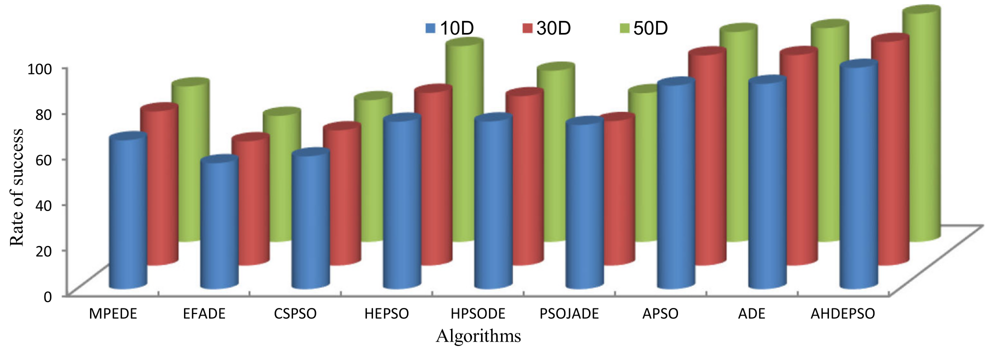
(b) 30*D*

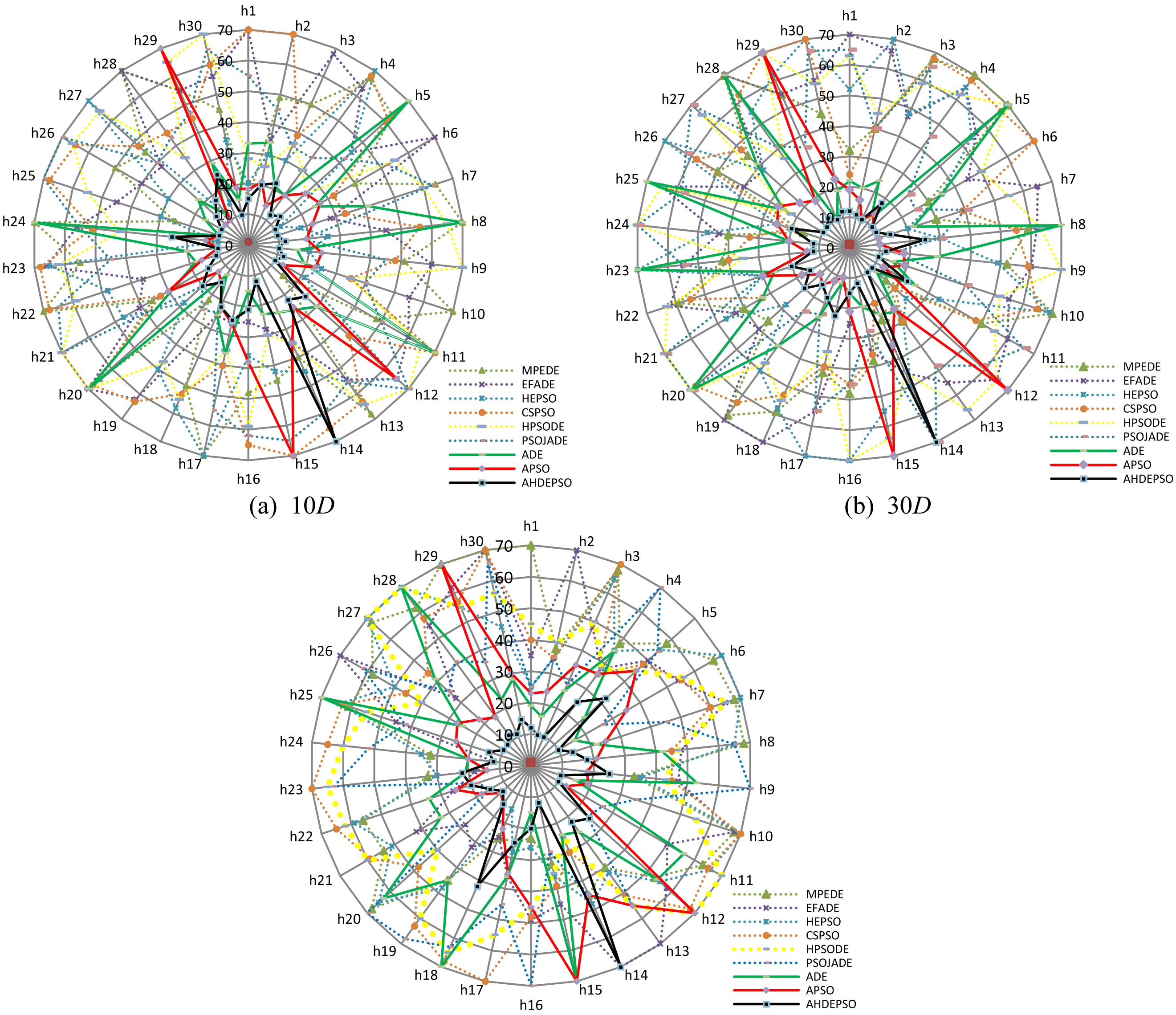
(c) 50*D*

Fig. 11 **a-c** Empirical distribution of normalized success performance on TS-2 (10*D*, 30*D* & 50*D*)

Whereas, SP/SP$_{best}$ Vs empirical distribution graph is plotted in Fig. 11a-c over all TS-2 functions. These figures confirm that the proposed algorithms have higher performance over other comparative algorithms. Besides, success rate distributions for 10, 30 & 50*D*TS-2 are evaluated and presented in Fig. 12. Graphical

Fig. 12 Success performance of the algorithms over TS-2 (10*D*, 30*D* & 50*D*)





(a) 10*D*

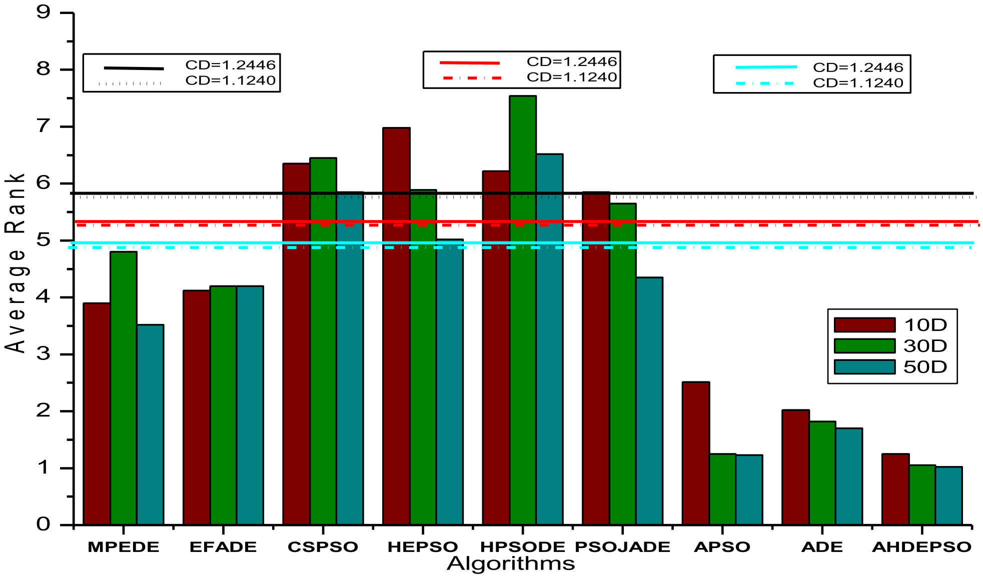
(b) 30*D*

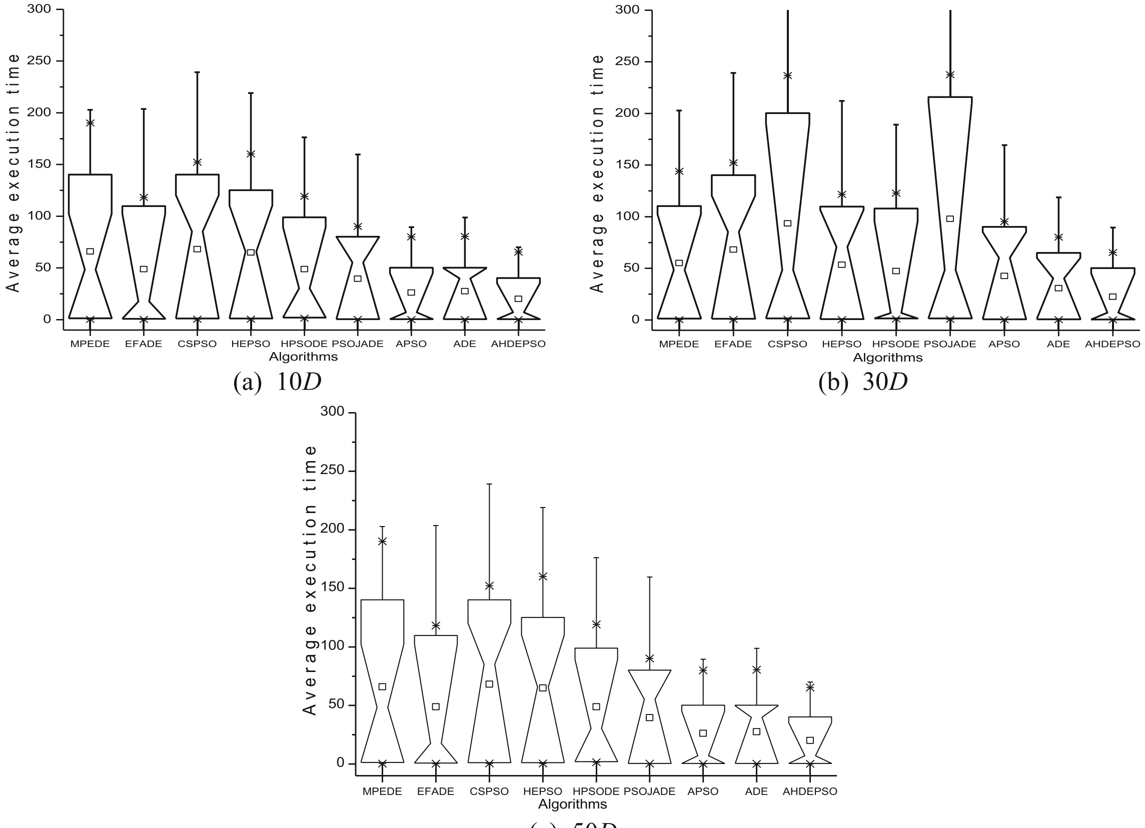
(c) 50*D*

Fig. 13 **a-c** Spider performance charts of different algorithms on TS-2 (10*D*, 30*D* & 50*D*)

representation shows that proposed ADE, APSO and AHDEPSO are competitively higher than others. Moreover, the spider charts (where closer distribution of the algorithm is to the circle's center for better performance of the algorithms) are employed on 10, 30 & 50*D*TS-2 to check the performance differences more intuitively among all algorithms. These charts can visualized in Fig. 13a-c. It can be observed from these figures that the proposed algorithms perform competitively on most of the functions. Additionally, significance level of proposed with other algorithms for 10, 30 & 50*D*TS-2 is evaluated through Bonferroni–Dunn's [147] procedure. It is applied as a post hoc procedure to calculate the critical difference (CD). In this test, solid and dotted lines are representing threshold for the control algorithm (here AHDEPSO) on two prevalent significance levels of 0.05 and 0.1. Also, it demonstrates that performance of two algorithms is significantly different if the difference in average ranking of methods is larger than CD. The Bonferroni–Dunn's bar chart of different algorithms on average rankings (obtained by Friedman test) for TS-2 is reported in Fig. 14. Here it can be observed that among all compared algorithms AHDEPSO significantly outperformed than others at both levels of significance. Also the average execution time of proposed algorithm with others of 30 independent runs for 10, 30 & 50*D*TS-2 has been reported in Fig. 15a-c through box plots. From corresponding figures, it is clearly visible that average execution time of proposed algorithms is less than others.

Fig. 14 Bonferroni Dunn bar chart of different algorithms on average rankings obtained by Friedman test for TS-2 (10*D*, 30*D* & 50*D*)





(a) 10*D*

(b) 30*D*

(c) 50*D*

Fig. 15 Average running time of different algorithms for TS-2 (10*D*, 30*D* & 50*D*)

Table 13 Parameter setting for real world problems

Algorithms	Year	Ref.	Control parameter		Population size	Stopping criterion (iterations)	Run
			Term	Values			
PSO	1995	[3]	w,C_1 & C_2	[0.543254–0.33362], 1.9460 & 1.8180,	50	1000	30
DE	1997	[4]	F & CR	0.12470 & 0.58143	50	1000	30
ABC	2007	[13]	–	–	50	1000	30
JADE	2009	[136]	F & *CR*	0.5 & 0.5	30	10,000	100
SaDE	2009	[148]	F & *CR*	*N*(0.5, 0.3) & *N*(CRm, 0.1)	30	10,000	100
CoDE	2011	[149]	F & *CR*	1.0 & 0.1	30	10,000	100
SLPSO	2012	[150]	*w*	0.9–0.5	30	10,000	100
HCLPSO	2015	[151]	*W* & C	0.9–0.4 & 1.49445	30	10,000	100
MSPSO	2018	[152]	C_1 & C_2	1.49445	30	10,000	100
DEPSO-2S	2012	[135]	*w*, C_1, C_2 & *CR*	0.72,1.19 & 0.5	–	–	30
MBDE	2016	[126]	–	–	100	5000	50
DPD	2016	[142]	F_A, F_C, CR_A & CR_C	0.5, 0.9, 0.9 & 0.9	100	200	30
ADE	Proposed		F_{min} & F_{max}	[0.1,0.5] & 1	30	1000	30
APSO			w_i, w_f, c_{1i}, c_{1f}, c_{2i} & c_{2f}	0.4, 0.9, 0.5, 2.5, 2.5 & 0.5	30	1000	30
AHDEPSO			–	–	30	1000	30

In general, from all above result analysis it can be declare that proposed ADE, APSO and AHDEPSO are performing better and/or equally with others. However, among three proposed algorithms AHDEPSO have larger competence.

iii). On RWPs: Real world problems

The results of proposed ADE, APSO and AHDEPSO algorithms on 3 RWPs are compared with traditional algorithm (PSO [3], DE [4], & ABC [13]), DE variants (JADE [136], SaDE [148], & CoDE [149]), PSO variants (SLPSO [150], HCLPSO [151], & MSPSO [152]), and hybrid variants (MBDE [126], DEPSO-2S [135], & DPD [142]). The parameters of all compared and proposed algorithms are listed in Table 13. The experimental comparative results in terms of best, worst, mean, std. (standard deviation), *p* values and average ranking of the objective function values are presented in Table 4. From Table 14, the proposed ADE, APSO and AHDEPSO produce better and/or equally results in terms of best, worst, and mean case of all RWPs.

Additionally, less std. and *p* values of proposed algorithms on most cases imply their better stability and reliable results for the majority of runs respectively. Also, in order to analyze the performance all algorithms are ranked depending on their mean value and reported in Table 14. From this table, it can be seen that AHDEPSO and ADE ranked 1st on all RWPs. Whereas, APSO secured 1st, 2nd and 4th rank on RWP-1, RWP-2 and RWP-3 successively. Hence, based on ranking results the proposed algorithms perform better than others. Moreover, the convergence graphs of all proposed and compared algorithms is plotted and presented in Fig. 16a-c. In these figures it can be clearly visualized that the proposed algorithms converge faster than others. Hence, proposed algorithms are computationally efficient.

All in all, from all above numerical, statistical and graphical result analysis it can be proclaim that proposed ADE, APSO and AHDEPSO are performing very competitive and/or equally with other compared algorithms. However, among three proposed algorithms AHDEPSO has greater efficiency.

4.3 Complexity analysis

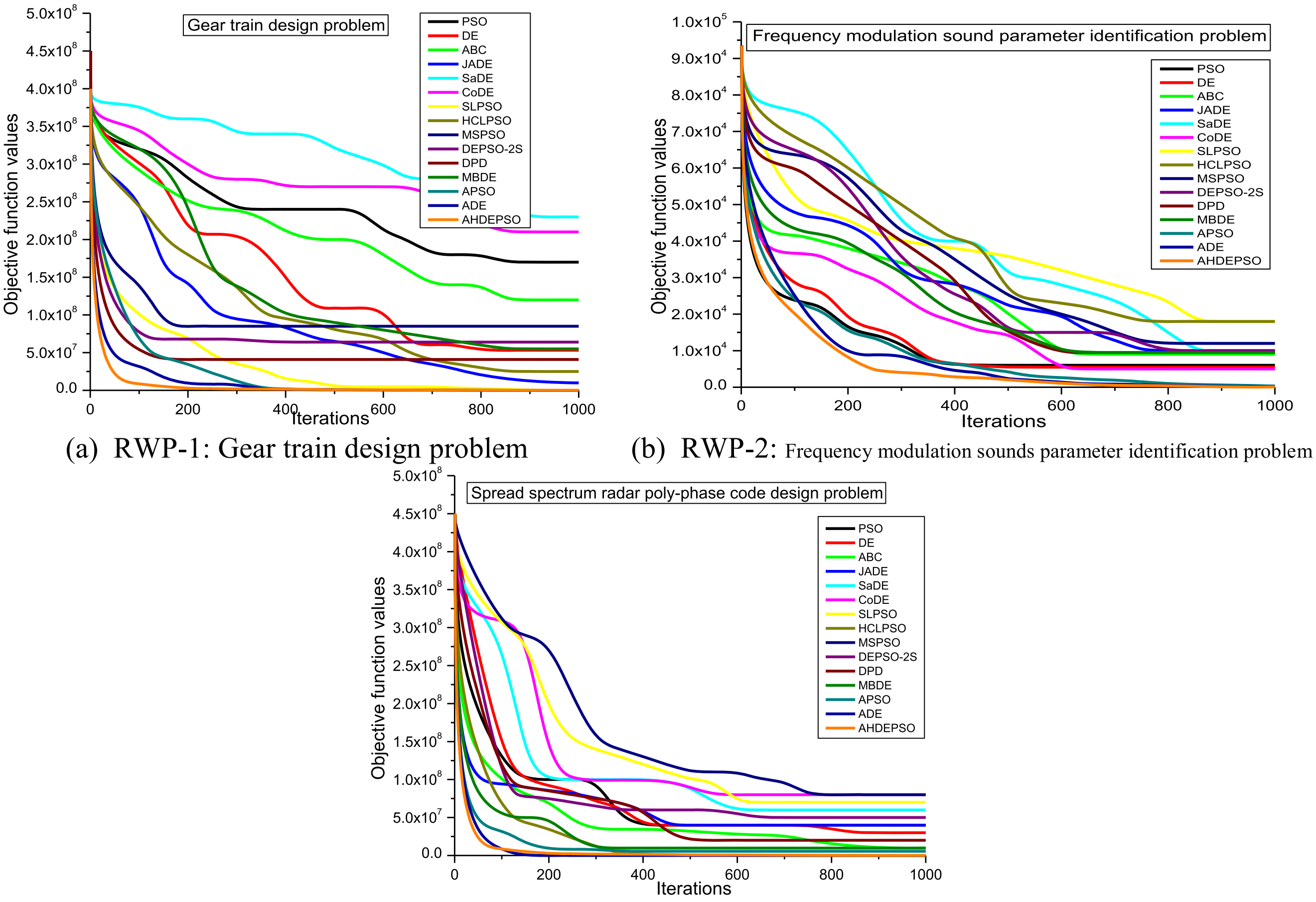
Some complexity analysis of the proposed algorithms is given as follows.

i). Algorithm complexity

According to the guidelines of CEC test suite the algorithm complexity has been investigated on 10, 30 & 50*D* TS-2. Firstly, T_0 time (in seconds) executed through subsequent program.

Table 14 Simulation results on real world problems

RWPs	Criteria	Algorithm														
		Traditional algorithms			DE variants			PSO variants			Hybrid variants			Proposed algorithms		
		PSO	DE	ABC	JADE	SaDE	CoDE	SLPSO	HCLPSO	MSPSO	DEPSO-2S	MBDE	DPD	ADE	APSO	AHDEPSO
RWP-1	best	2.22e−06	1.78e−09	1.33e−10	1.05e−13	**0.00e+00**	**0.00e+00**	4.74e−11	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+00**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	worst	1.35e−06	9.09e−07	4.70e−07	6.34e−05	7.24e−06	7.94e−04	1.17e−01	1.18e−02	3.85e−02	7.17e−03	4.30e−03	5.73e−03	4.70e−014	6.34e−009	**0.24e−016**
	mean	7.17e−03	5.73e−03	4.30e−03	1.87e−01	**0.00e+00**	1.02e+00	9.66e+00	3.32e+00	7.25e+00	2.07e+00	2.06e+00	1.85e+00	**0.00e+000**	**0.00e+000**	**0.00e+000**
	std.	2.06e−03	1.65e−03	1.23e−03	2.06e−03	4.16e−03	5.23e−02	3.56e−03	7.11e−02	6.91e−03	1.67e+000	1.88e-05	3.11e-01	1.65e−009	1.23e−007	**2.06e−011**
	p value	0.9031	0.0527	0.0438	0.0527	0.4094	0.4570	0.0384	0.0582	0.0351	0.0351	0.0489	0.0342	0.0417	0.0327	**0.0338**
	rank	11	10	9	12	**1**	2	8	6	7	5	4	3	**1**	**1**	**1**
RWP-2	best	1.61e−02	8.13e−03	2.50e−02	**0.00e+00**	2.39e−29	**0.00e+00**	**0.00e+00**	4.75e−21	**0.00e+00**	1.98e−29	**0.00e+00**	**0.00e+00**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	worst	1.17e−01	1.18e−02	3.85e−02	1.44e+05	8.14e−04	3.45e−02	8.13e+05	1.88e−07	1.93e−09	3.18e−02	7.48e+05	4.65e−02	1.18e−007	3.18e−006	**1.23e−009**
	mean	1.17e−02	7.96e−03	2.21e−02	2.39e−17	7.67e−18	1.94e−17	1.86e−29	1.49e−15	**0.00e+00**	1.39e-10	4.21e+00	4.55e-12	**0.00e+000**	2.67e−030	**0.00e+000**
	std.	4.43e+05	7.94e−04	3.14e−02	6.33e−06	7.89e−07	6.45e−06	7.16e+00	5.62e+00	1.39e−03	2.65e-10	2.86e-10	7.75e-13	2.65e-010	2.65e-009	**8.65e-016**
	p value	0.0342	0.0412	0.0327	0.0582	0.0527	0.0327	0.0419	0.0327	0.0518	0.0327	0.0477	0.0497	0.0518	0.0327	**0.0321**
	rank	11	10	12	6	4	5	3	7	**1**	9	13	8	**1**	2	**1**
RWP-3	best	1.95e+00	1.88e+00	1.62e+00	1.02e+00	1.11e+00	5.00e+01	1.07e+00	7.59e-01	8.79e−01	1.56e+00	3.14e+00	2.36e+05	**0.00e+000**	**0.00e+000**	**0.00e+000**
	worst	2.89e+00	2.77e+00	2.88e+00	1.44e+05	8.14e+00	3.45e+00	8.13e+05	2.23e+00	3.75e+00	3.80e+00	2.57e+00	1.07e+00	1.03e+000	1.07e+000	**1.57e-005**
	mean	2.42e+00	2.44e+00	2.34e+00	1.17e+00	1.31e+00	6.73e+01	1.26e+00	1.08e+00	1.21e+00	3.00e-01	1.04e-01	1.05e-01	**0.00e+000**	2.56e-001	**0.00e+000**
	std.	0.20e+00	0.22e+00	0.24e+00	1.65e+00	2.54e+00	4.76e+01	1.97e+00	7.59e-01	2.13e+00	7.07e-002	1.25e-04	2.15e-02	8.95e-008	1.85e-007	**2.25e-009**
	p value	0.0351	0.0342	0.0412	0.0518	0.0327	0.0627	0.0127	0.0517	0.0489	0.0417	0.0327	0.0582	0.0498	0.0419	**0.0107**
	rank	13	13	10	7	9	14	8	6	7	5	2	3	**1**	4	**1**



(a) RWP-1: Gear train design problem

(b) RWP-2: Frequency modulation sounds parameter identification problem

(c) RWP-3: The spread spectrum radar poly-phase code design problem

Fig. 16 a-c Convergence graph for real world problems

```
for $i$= 1:1000000 $do$
    $x = 0.55 + (double)i;$
    $x = x + x; x = \frac{x}{2}; x = x * x; x = \sqrt{x}, x = log(x);$
    $x = e^{x}; x = \frac{x}{x+2};$
end
```

After this computing T_1 time on and h_{18} (TS-2 function). Further, T_2 time (evaluated 5 times) for all algorithms to completely execution of the both functions on 200,000 evaluations and store its mean value denoted as $\widehat{T}_2$. Thereafter calculated $\frac{\widehat{T_2 - T_1}}{T_0}$ for each algorithm and reported in Table 15. Also, complexity chart of proposed algorithms with others are presented in Fig. 17. It can be being observed from Table 15 and Fig. 17, the time complexity of proposed algorithms is significantly less than other comparative algorithms.

Table 15 Algorithm complexity results on TS-2

Algorithm	10*D*	30*D*	50*D*
T_0	0.0216	0.0216	0.0216
T_1	0.8971	0.5831	1.3630
MPEDE	94.23	90.88	98.65
EFADE	111.25	99.54	120.88
CSPSO	192.87	167.87	190.23
HEPSO	140.25	130.98	170.54
HPSODE	150.99	140.25	160.23
PSOJADE	140.53	130.55	155.21
ADE	52.32	48.45	53.09
APSO	77.88	75.87	82.98
AHDEPSO	38.98	33.88	40.02

ii). Time complexity

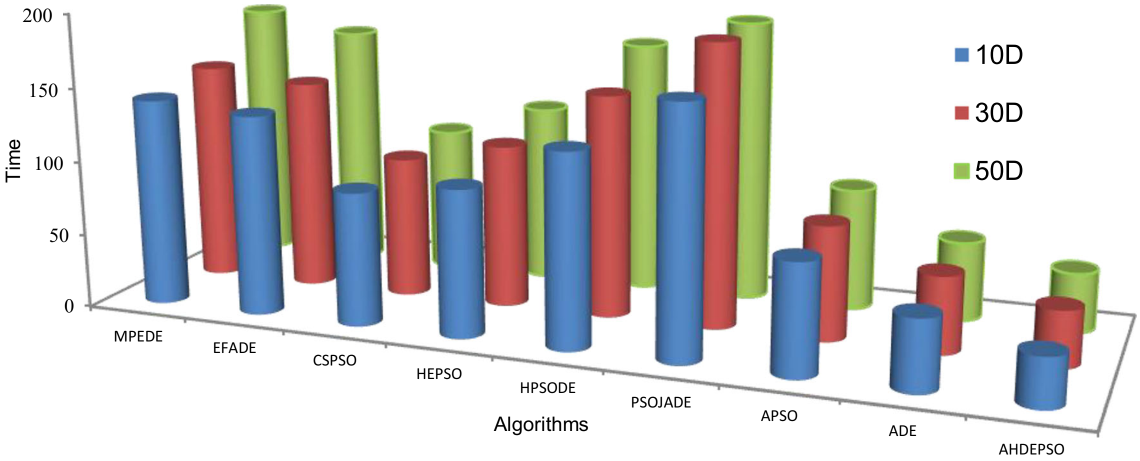


Fig. 17 Algorithm complexity on TS-2 (10*D*, 30*D* & 50*D*)

According to the pseudo-code, AHDEPSO has the following time complexity.

a). Initialization of *np*-population requires *O(np.D*) time.
b). Evaluation and sorting population according to fitness function values needs $O(t_{max} \times np)$ time.
c). Division of population into pop_1 and pop_2 procedure requires $O(t_{max} \times np)$ time.
d). Evaluation of pop_1 (by ADE) and pop_2 (by APSO) takes $O(t_{max} \times \frac{np}{2} \times \frac{np}{2}) = O(t_{max} \times \frac{np^2}{4})$ time.
e). Merging *pop* and implementation of algorithm requires $O(t_{max} \times np \times np)$= $O(t_{max} \times np^2)$ time.

Therefore, the total time complexity of AHDEPSO for maximum number of iterations is

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

iii). Space complexity

The space complexity of proposed AHDEPSO algorithm is the maximum amount of space that used by above algorithm. Thus, the total space complexity of proposed AHDEPSO algorithm is $O(max(np, np, np, \frac{np^2}{4}, np^2) \times D) = O(np^2 \times D)$.

5 Conclusion with future works

In this study, magnificent survey of various recent-past traditional algorithms with DE and PSO variants as well as their hybrids have been examined along with their applied fields. After this, an advanced DE (ADE, to avoid premature convergence) and PSO (APSO, to avoid stagnation) as well as their hybrid (AHDEPSO, to balance between exploration and exploitation) has been proposed for unconstrained optimization problems. The briefed summary of these proposed algorithms are given as follows.

(i). The novel mutation strategy, crossover probability and altered selection schemes of ADE will provide high and low population diversity at start and end of the algorithm respectively.
(ii). The novel gradually varying (decreasing and/or increasing) parameters of APSO can well-balanced exploration and exploitation capabilities and promotes particles to search high quality solution.
(iii). The AHDEPSO yields guaranteed convergence and diversifying solutions due to different convergence characteristics of ADE and APSO as well as based on multi-population approach as well as the divided population is merged with other in a pre-defined manner.

Further, the effectiveness of the proposed algorithms tested on TS-1 (basic unconstrained benchmark function) and TS-2 (IEEE CEC 2017 functions) along with 3 RWPs (real world problems). The numerical, statistical, and graphical analysis of the proposed algorithms is compared against traditional DE and PSO with their recent variants and hybrids as well as over many state-of-the-art algorithms. The comparative results shapes that the proposed algorithms become more robust and effective. Thus, it is conclusive that the proposed algorithms can be treated as a vital alterative in the field of EAs. Moreover, in the view of feasibilities, superiorities and solution optimality among proposed algorithms AHDEPSO outperformed. In addition, the effectiveness of the proposed algorithms can be tested by some more complicated real-world applications and new EAs will be developed in future.

References

1. Simpson AR, Dandy GC, Murphy LJ (1994) Genetic algorithms compared to other techniques for pipe optimization. J Water Resour Plan Manag 20:423–443
2. Goldberg DE (1989) Genetic algorithms in search, optimization, and machine learning. Addison- Wesley Publishing Company
3. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceeding of IEEE international conference on neural networks, pp 1942–1948
4. Storn R, Price K (1997) Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359
5. Murase H, Wadano A (1998) Photosynthetic algorithm for machine learning and TSP. IFAC Proceedings 31:19–24
6. de Castro LN, von Zuben FJ (2000) The clonal selection algorithm with engineering applications. In: Proceedings of the genetic and evolutionary computation conference, Las Vegas, Nevada, USA, pp 36–39
7. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. Simulation 76(2):60–68
8. Eusuff M, Lansey KE (2003) Optimization of water distribution network design using the shuffled frog leaping algorithm. J Water Resour Plan Manag 129(3):210–225
9. Wedde HF, Farooq M, Zhang Y (2004) BeeHive: an efficient fault-tolerant routing algorithm inspired by honey bee behavior. Springer, Berlin, pp 83–94
10. Pinto P, Runkler TA, Sousa JM (2005) Wasp swarm optimization of logistic systems, Adaptive and Natural Computing Algorithms, pp 264–267
11. Du H, Wu X, Zhuang J, (2006)Small-world optimization algorithm for function optimization, Advances in Natural Computation, pp 264–273
12. Mehrabian AR, Lucas C (2006) A novel numerical optimization algorithm inspired from weed colonization. Ecol Inform 1(4):355–366
13. Karaboga D, Basturk B (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony algorithm. `J Glob Optim 39(3):459–471
14. Havens TC, Spain CJ, Salmon NG, Keller JM (2008) Roach infestation optimization. In: Proceedings of the IEEE Swarm Intelligence Symposium, pp 1–7
15. Simon D (2008)Biogeography-based optimization. IEEE Trans Evol Comput 12(6):702–713
16. 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
17. Yang X (2009) Firefly algorithms for multimodal optimization, stochastic algorithms: foundations and applications, vol 5792. Springer, Berlin Heidelberg, pp 169–178
18. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) A gravitational search algorithm. Inf Sci 179(13):2232–2248
19. Yang XS (2010) A new metaheuristic bat-inspired algorithm, In: Proceedings of the fourth international workshop on nature inspired cooperative strategies for optimization (NICSO 2010), Berlin, Heidelberg. 65–74
20. Rao RV, Savsani VJ, Vakharia DP (2011)Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems. Comput Aided Des 43(3):303–315
21. Eskandar H, Sadollah A, Bahreininejad A, Hamdi M (2012) Water cycle algorithm – a novel metaheuristic optimization method for solving constrained engineering optimization problems. Comput Struct 110-111:151–166
22. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845
23. Cuevas E, Cienfuegos M, Zaldívar D, Pérez-Cisneros M (2013) A swarm optimization algorithm inspired in the behavior of the social-spider. Expert Syst Appl 40(16):6374–6384
24. Bansal JC, Sharma H, Jadon SS, Clerc M (2014) Spider monkey optimization algorithm for numerical optimization. Memetic Computing 6(1):31–47
25. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
26. Zheng YJ (2015) Water wave optimization: a new nature-inspired metaheuristic. Comput Oper Res 55:1–11
27. Mirjalili S (2015)Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm. Knowl-Based Syst 89:228–249
28. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67
29. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete and multi-objective problems. Neural Comput Applic 27(4):1053–1073
30. Saremi S, Mirjalili S, Lewis A (2017) Grasshopper optimisation algorithm: theory and application. Adv Eng Softw 105:30–47
31. Pierezan J, Dos Santos Coelho L (2018) Coyote optimization algorithm: a new metaheuristic for global optimization problems, IEEE Congress on Evolutionary Computation, pp 1–8
32. Shabani A, Asgarian B, Gharebaghi SA, Salido MA, Giret A (2019) A new optimization algorithm based on search and rescue operations. Math Probl Eng 2019:1–23
33. Marzbali AG (2020) A novel nature-inspired meta-heuristic algorithm for optimization: bear smell search algorithm. Soft Comput: 1–33
34. Neri F, Tirronen V (2010) Recent advances in differential evolution: a survey and experimental analysis. Artif Intell Rev 33(1–2): 61–106
35. Das S, Suganthan PN (2011) Differential evolution: a survey of the state-of-the-art. IEEE Trans Evol Comput 15(1):4–31
36. Thangaraj R, Pant M, Abraham A, Bouvry P (2011) Particle swarm optimization: hybridization perspectives and experimental illustrations. Appl Math Comput 217(12):5208–5226
37. Sengupta S, Basak S, Peters RA (2019) Particle swarm optimization: a survey of historical and recent developments with hybridization perspectives. Mach Learn Knowl Extract 1(1):157–191
38. Das S, Abraham A, Konar A (2008) Particle swarm optimization and differential evolution algorithms: technical analysis, applications and hybridization perspectives. Adv Comput Intell Ind Syst 5:1–38
39. Xin B, Chen J, Zhang J, Fang H, Peng Z (2012) Hybridizing differential evolution and particle swarm optimization to design powerful optimizers: a review and taxonomy. IEEE Trans Syst Man Cybern 42(5):744–767
40. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
41. Joshi R, Sanderson AC (1997) Minimal representation multisensor fusion using differential evolution. In: Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation CIRA'97. Towards New Computational Principles for Robotics and Automation, pp 266–273
42. Cheng S, Hwang C (1998) Designing pid controllers with a minimum IAE criterion by a differential evolution algorithm. Chem Eng Commun 170(1):83–115
43. Lee MH, Han C, Chang KS (1999) Dynamic optimization of a continuous polymer reactor using a modified differential evolution algorithm. Ind Eng Chem Res 38(12):4825–4831

44. Kyprianou A, Giacomin J, Worden K, Heidrich M, Bocking J (2000) Differential evolution based identification of automotive hydraulic engine mount model parameters. Proc Inst Mech Eng Part D-J Automob Eng 214(3):249–264
45. Ruzek B, Kvasnicka M (2001) Differential evolution algorithm in the earthquake hypocenter location. Pure Appl Geophys 158(4): 667–693
46. Chen C, Chen D, Cao G (2002) An improved differential evolution algorithm in training and encoding prior knowledge into feedforward networks with application in chemistry. Chemom Intell Lab Syst 64(1):27–43
47. Ilonen J, Kamarainen J-K, Lampinen J (2003) Differential evolution training algorithm for feed-forward neural networks. Neural Process Lett 17(1):93–105
48. Kapadi MD, Gudi RD (2004) Optimal control of fed-batch fermentation involving multiple feeds using differential evolution. Process Biochem 39(11):1709–1721
49. Rane TD, Dewri R, Ghosh S, Chakraborti N, Mitra K (2005) Modeling the recrystallization process using inverse cellular automata and genetic algorithms: studies using differential evolution. J Phase Equilib Diffus 26(4):311–321
50. Babu BV, Angira R (2006) Modified differential evolution (MDE) for optimization of non-linear chemical processes. Comput Chem Eng 30(6–7):989–1002
51. Chang CF, Wong JJ, Chiou JP, Su CT (2007) Robust searching hybrid differential evolution method for optimal reactive power planning in large-scale distribution systems. Electr Power Syst Res 77(5–6):430–437
52. Noman N, Iba H (2008) Differential evolution for economic load dispatch problems. Electr Power Syst Res 78(8):1322–1331
53. Das S, Konar A (2009) Automatic image pixel clustering with an improved differential evolution. Appl Soft Comput 9(1):226–236
54. Amjady N, Sharifzadeh H (2010) Solution of non-convex economic dispatch problem considering valve loading effect by a new modified differential evolution algorithm. Int J Electr Power Energy Syst 32(8):893–903
55. Uyar AS, Turkay B, Keles A (2011) A novel differential evolution application to short-term electrical power generation scheduling. Int J Electr Power Energy Syst 33(6):1236–1242
56. Dos Santos GS, Luvizotto LGJ, Mariani VC, Coelho L d S (2012) Least squares support vector machines with tuning based on chaotic differential evolution approach applied to the identification of a thermal process. Expert Syst Appl 39(5):4805–4812
57. Tsai JT, Fang JC, Chou JH (2013) Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. Comput Oper Res 40(12):3045–3055
58. Baskan O, Ceylan H (2014) Modified differential evolution algorithm for the continuous network design problem. Procedia Soc Behav Sci 111:48–57
59. Guo SM, Yang CC (2015) Enhancing differential evolution utilizing eigenvector-based crossover operator. IEEE Trans Evol Comput 19(1):31–49
60. Ayala HVH, dos Santos FM, Mariani VC, Coelho L d S (2015) Image thresholding segmentation based on a novel beta differential evolution approach. Expert Syst Appl 42(4):2136–2142
61. Chen N, Chen WN, Zhang J (2015) Fast detection of human using differential evolution. Signal Process 110:155–163
62. Do DTT, Lee S, Lee J (2016) A modified differential evolution algorithm for tensegrity structures. Compos Struct 158:11–19
63. Sethanan K, Pitakaso R (2016) Differential evolution algorithms for scheduling raw milk transportation. Comput Electron Agric 121:245–259
64. Basu M (2016)Quasi-oppositional differential evolution for optimal reactive power dispatch. Int J Electr Power Energy Syst 78: 29–40
65. Vivekanandan T, Sriman Narayana Iyengar NC (2017) Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease. Comput Biol Med 90:125–136
66. Suresh S, Lal S (2017) Modified differential evolution algorithm for contrast and brightness enhancement of satellite images. Appl Soft Comput 61:622–641
67. Sakr WS, EL-Sehiemy RA, Azmy AM (2017) Adaptive differential evolution algorithm for efficient reactive power management. Appl Soft Comput 53:336–351
68. Qiu X, Xu JX, Xu Y, Tan KC (2018) A new differential evolution algorithm for minimax optimization in robust design. IEEE Trans Cybern 48(5):1355–1368
69. Yuzgec U, Eser M (2018) Chaotic based differential evolution algorithm for optimization of baker's yeast drying process. Egypt Inform J 19(3):151–163
70. Buba AT, Lee LS (2018) A differential evolution for simultaneous transit network design and frequency setting problem. Expert Syst Appl 106:277–289
71. Yang X, Li J, Peng X (2019) An improved differential evolution algorithm for learning high-fidelity quantum controls. Sci Bull 64(19):1402–1408
72. Awad NH, Ali MZ, Mallipeddi R, Suganthan PN (2019) An efficient Differential Evolution algorithm for stochastic OPF based active-reactive power dispatch problem considering renewable generators. Appl Soft Comput 76:455–458
73. Prabha S, Yadav R (2019) Differential evolution with biological-based mutation operator. Eng Sci Technol Int J 23(2):253–263
74. Li S, Gu Q, Gong W, Ning B (2020) An enhanced adaptive differential evolution algorithm for parameter extraction of photovoltaic models. Energy Convers Manag 205:112443
75. 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. J Pet Sci Eng 188:106916
76. Zhenya H, Chengjian W, Luxi Y, Xiqi G, Susu Y, Eberhart RC, Shi Y 1998 Extracting rules from fuzzy neural network by particle swarm optimisation. In: Proceedings of IEEE International Conference on Evolutionary Computation. IEEE World Congress on Computational Intelligence, pp, 74-77
77. Eberhart RC, Xiaohui H (1999) Human tremor analysis using particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation-CEC99, pp, 1927–1930
78. Naka S, Genji T, Yura T, Fukuyama Y, Hayashi N (2002) Distribution state estimation considering nonlinear characteristics of practical equipment using hybrid particle swarm optimization. In: Proceedings of International Conference on Power System Technology, PowerCon, pp 1083–1088
79. Abido AA (2001) Particle swarm optimization for multimachine power system stabilizer design. 2001 Power Engineering Society Summer Meeting, Conference Proceedings, pp 1346–1351

80. Al-kazemi B, Mohan CK (2002) Training feedforward neural networks using multi-phase particle swarm optimization. In: Proceedings of the 9th International Conference on Neural Information Processing, pp 2615–2619
81. Gaing ZL (2003) Discrete particle swarm optimization algorithm for unit commitment. IEEE Power Engineering Society General Meeting, pp 418–424
82. Pang W, Wang K, Zhou C, Dong L 2004 Fuzzy discrete particle swarm optimization for solving traveling salesman problem. In: Proceeding of the Fourth International Conference on Computer and Information Technology
83. Esmin AAA, Lambert-Torres G, Zambroni de Souza AC (2005) A hybrid particle swarm optimization applied to loss power minimization. IEEE Trans Power Syst 20(2):859–866
84. Meissner M, Schmuker M, Schneider G (2006) Optimized particle swarm optimization (OPSO) and its application to artificial neural network training. BMC Bioinformatics 7(1):125
85. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design problems. Eng Appl Artif Intell 20(1):89–99
86. Zhang C, Sun J, Zhu X, Yang Q (2008) An improved particle swarm optimization algorithm for flowshop scheduling problem. Inf Process Lett 108(4):204–209
87. Meneses AA d M, Machado MD, Schirru R (2009) Particle Swarm Optimization applied to the nuclear reload problem of a Pressurized Water Reactor. Prog Nucl Energy 51(2):319–326
88. Azadani EN, Hosseinian SH, Moradzadeh B (2010) Generation and reserve dispatch in a competitive market using constrained particle swarm optimization. Int J Electr Power Energy Syst 32(1):79–86
89. Kang Q, He H (2011) A novel discrete particle swarm optimization algorithm for meta-task assignment in heterogeneous computing systems. Microprocess Microsyst 35(1):10–17
90. Kar R, Mandal D, Mondal S, Ghoshal SP (2012) Craziness based particle swarm optimization algorithm for FIR band stop filter design. Swarm Evol Comput 7:58–64
91. Lim WH, Mat Isa NA (2013)Two-layer particle swarm optimization with intelligent division of labor. Eng Appl Artif Intell 26(10): 2327–2348
92. Zhang W, Ma D, Wei JJ, Liang HF (2014) A parameter selection strategy for particle swarm optimization based on particle positions. Expert Syst Appl 41(7):3576–3584
93. Basu M (2015) Modified particle swarm optimization for nonconvex economic dispatch problems. Int J Electr Power Energy Syst 69:304–312
94. Eddaly M, Jarboui B, Siarry P (2016) Combinatorial particle swarm optimization for solving blocking flowshop scheduling problem. J Comput Des Eng 3(4):295–311
95. Zhang Y, Zhao Y, Fu X, Xu J (2016) A feature extraction method of the particle swarm optimization algorithm based on adaptive inertia weight and chaos optimization for Brillouin scattering spectra. Opt Commun 376:56–66
96. Ngo TT, Sadollah A, Kim JH (2016) A cooperative particle swarm optimizer with stochastic movements for computationally expensive numerical optimization problems. J Comput Sci 13:68–82
97. Li Y, Bai X, Jiao L, Xue Y (2017)Partitioned-cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. Appl Soft Comput 56:345–356
98. Phung MD, Quach CH, Dinh TH, Ha Q (2017) Enhanced discrete particle swarm optimization path planning for UAV vision-based surface inspection. Autom Constr 81:25–33
99. Qin Q, Cheng S, Chu X, Lei X, Shi Y (2017) Solving non-convex/non-smooth economic load dispatch problems via an enhanced particle swarm optimization. Appl Soft Comput 59:229–242
100. Mishra KK, Bisht H, Singh T, Chang V (2018) A direction aware particle swarm optimization with sensitive swarm leader. Big Data Research 14:57–67
101. Li Z, Hu C, Ding C, Liu G, He B (2018) Stochastic gradient particle swarm optimization based entry trajectory rapid planning for hypersonic glide vehicles. Aerosp Sci Technol 76:176–186
102. Tian D, Shi Z (2018) MPSO: modified particle swarm optimization and its applications. Swarm Evolut Comput 41:49–68
103. Parouha RP (2019)Nonconvex/nonsmooth economic load dispatch using modified time-varying particle swarm optimization. Comput Intell 35(4):717–744
104. Hosseini SA, Hajipour A, Tavakoli H (2019) Design and optimization of a CMOS power amplifier using innovative fractional-order particle swarm optimization. Appl Soft Comput 85:1–10
105. Dash PP, Patra D (2019) Mutation based self-regulating and self-perception particle swarm optimization for efficient object tracking in a video. Measurement 144:311–327
106. Lanlan K, Ruey SC, Wenliang C, Yeh C (2020)Non-inertial opposition-based particle swarm optimization and its theoretical analysis for deep learning applications. Appl Soft Comput 88:1–10
107. 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. Artif Intell Med 104:1–14
108. Phung MD, Ha QP (2020)Motion-encoded particle swarm optimization for moving target search using UAVs. Appl Soft Comput 97:106705
109. Hendtlass T (2001) A combined swarm differential evolution algorithm for optimization problems, in: proceedings of 14th international conference on industrial and engineering applications of artificial intelligence and expert systems. Lect Notes Comput Sci 2070:11–18
110. Zhang WJ, Xie XF (2003) DEPSO: hybrid particle swarm with differential evolution operator, proceedings of the IEEE international conference on systems, man and cybernetics Washington DC. USA:3816–3821
111. Talbi H, Batouche M (2004) Hybrid particle swarm with differential evolution for multimodal image registration. In: Proceedings of the IEEE International Conference on Industrial Technology. 3, 1567–1573
112. Hao ZF, Gua G-H, Huang H (2007) A particle swarm optimization algorithm with differential evolution, Proceedings of Sixth International Conference on Machine Learning and Cybernetics. 1031–1035
113. Niu B, Li L (2008) A novel PSO-DE-based hybrid algorithm for global optimization. Lect Notes Comput Sci 5227:156–163
114. Wang Y, Cai Z (2009) A hybrid multi-swarm particle swarm optimization to solve constrained optimization problems. Front Comput Sci 3:38–52
115. Caponio A, Neri F, Tirronen V (2009) Superfit control adaption in memetic differential evolution frameworks. Soft Comput 13(8–9): 811–831

116. Liu H, Cai Z, Wang Y (2010) Hybridizing particle swarm optimization with differential evolution for constrained numerical and engineering optimization. Appl Soft Comput 10(2):629–640
117. Xin B, Chen J, Peng Z, Pan F (2010) An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization. Sci China Inf Sci 53(5):980–989
118. Pant M, Thangaraj R, Abraham A (2011) DE-PSO: A new hybrid meta-heuristic for solving global optimization problems. New Math Nat Comput 7(3):363–381
119. Epitropakis MG, Plagianakos VP, Vrahatis MN (2012) Evolving cognitive and social experience in particle swarm optimization through differential evolution: a hybrid approach. Inf Sci 216: 50–92
120. Nwankwor E, Nagar AK, Reid DC (2012) Hybrid differential evolution and particle swarm optimization for optimal well placement. Comput Geosci 17(2):249–268
121. Sahu BK, Pati S, Panda S (2014) Hybrid differential evolution particle swarm optimisation optimised fuzzy proportional–integral derivative controller for automatic generation control of interconnected power system. IET Gener Transm Distrib 8(11):1789–1800
122. Yu X, Cao J, Shan H, Zhu L, Guo J (2014) An adaptive hybrid algorithm based on particle swarm optimization and differential evolution for global optimization. Sci World J 2014:215472
123. Seyedmahmoudian M, Rahmani R, Mekhilef S, Than Oo AM, Stojcevski A, Soon TK, Ghandhari AS (2015) Simulation and hardware implementation of new maximum power point tracking technique for partially shaded PV system using hybrid DEPSO method. Trans Sustain Energy 6(3):850–862
124. Parouha RP, Das KN (2015) An efficient hybrid technique for numerical optimization and applications. Comput Ind Eng 83: 193–216
125. Tang B, Zhu Z, Luo J (2016) Hybridizing particle swarm optimization and differential evolution for the Mobile robot global path planning. Int J Adv Robot Syst 13(3):1–17
126. Parouha RP, Das KN (2016) A robust memory based hybrid differential evolution for continuous optimization problem. Knowl-Based Syst 103:118–131
127. Parouha RP, Das KN (2016) DPD: An intelligent parallel hybrid algorithm for economic load dispatch problems with various practical constraints. Expert Syst Appl 63:295–309
128. Famelis IT, Alexandridis A, Tsitouras C (2017) A highly accurate differential evolution–particle swarm optimization algorithm for the construction of initial value problem solvers. Eng Optim 50(8):1364–1379
129. 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. Math Probl Eng:1–9
130. 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 Comput & Applic:1–35
131. Too J, Abdullah AR, Saad NM (2019) Hybrid binary particle swarm optimization differential evolution-based feature selection for EMG signals classification. Axioms 8(3):1–17
132. Dash J, Dam B, Swain R (2019) Design and implementation of sharp edge FIR filters using hybrid differential evolution particle swarm optimization. AEU Int J Electron Commun 114:1–61
133. Zhao X, Zhang Z, Xie Y, Meng J (2020)Economic-environmental dispatch of microgrid based on improved quantum particle swarm optimization. Energy 195:1–39
134. Awad N, Ali M, Liang J, Qu B, Suganthan P (2016) Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization, Technical Report
135. El Dor A, Clerc M, Siarry P (2012) Hybridization of differential evolution and particle swarm optimization in a new algorithm DEPSO-2S. Swarm Evolut Comput 7269:57–65
136. Zhang J, Sanderson C (2009) JADE: adaptive differential evolution with optional external archive. IEEE Trans Evol Comput 13(5):945–958
137. Tanabe R, Fukunaga A (2013)Success-history based parameter adaptation for differential evolution. In: IEEE Congress on Evolutionary Computation. 71–78
138. Mahmoodabadi MJ, Mottaghi ZS, Bagheri A (2014) High exploration particle swarm optimization. J Inf Sci 273:101–111
139. 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. Comput Phys Commun 219:79–86
140. 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. J Comput Sci 26:488–500
141. Chegini SN, Bagheri A, Najafi F (2018) A new hybrid PSO based on sine cosine algorithm and levy flight for solving optimization problems. Appl Soft Comput 73:697–726
142. Das KN, Parouha RP (2015) An ideal tri-population approach for unconstrained optimization and applications. Appl Math Comput 256:666–701
143. Wu G, Mallipeddi R, Suganthan PN, Wang R, Chen H (2016) Differential evolution with multipopulation based ensemble of mutation strategies. Inf Sci 329:329–345
144. Mohamed AW, Suganthan PN (2018)Real-parameter unconstrained optimization based on enhanced fitness-adaptive differential evolution algorithm with novel mutation. Soft Comput 22(10):3215–3235
145. Meng A, Chen Y, Yin H, Chen S (2014) Crisscross optimization algorithm and its application. Knowl-Based System 67:218–229
146. Du S-Y, Liu Z-G(2019) Hybridizing particle swarm optimization with JADE for continuous optimization. Multimed Tools Appl:1–18
147. Zar JH (1999) Biostatistical analysis. Prentice Hall, Englewood Cliffs
148. Qin AK, Huang VL, Suganthan PN (2009) Differential evolution algorithm with strategy adaptation for global numerical optimization. IEEE Trans Evol Comput 13(2):398–417
149. Wang Y, Cai ZZ, Zhang QF (2011) Differential evolution with composite trial vector generation strategies and control parameters. IEEE Trans Evol Comput 15(1):55–66
150. Li C, Yang S, Nguyen TT (2012) A self-learning particle swarm optimizer for global optimization problems. IEEE Trans Syst Man Cybern 42(3):627–646
151. Lynn N, Suganthan P (2015) Heterogeneous comprehensive learning particle swarm optimization with enhanced exploration and exploitation. Swarm Evolut Comput 24:11–24
152. Xuewen X, Ling G, Hui ZZ (2018) A multi-swarm particle swarm optimization algorithm based on dynamical topology and purposeful. Appl Soft Comput 67:126–140

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Raghav Prasad Parouha is currently Assistant Professor at Department of Mathematics, Indira Gandhi National Tribal University (A Central Univer-?sity Established by an Act of Parliament), Amarkantak, Madhya Pradesh, India. He did Ph. D. from National Institute of Technology (NIT), Silchar, Assam India in August 2015. His area of research interests related to Computational Intelligence including Algorithm Design, Real World Application etc. He has received Young Scientist Award (YSA) by International Academy of Physical Sciences (IAPS), Allahabad (U.P.), India. He has about 35 repute national and international publications and attended about 30 and organized around 05 workshops/seminars etc.

Pooja Verma is currently research scholar at Department of Mathematics, Indira Gandhi National Tribal University, Amarkantak, Madhya Pradesh, India. She did B. Sc. in electronics from Pt. Sambhunath Shukla University Shahdol (M.P.) in 2014 and M. Sc. In mathematics from St. Aloysius College (A+) Autonomous Jabalpur (M.P.) in the year of 2016. Currently she is perusing Ph. D. under the supervision of Dr. Raghav Prasad Parouha, in the area of Evolutionary Algorithms (EAs) like Differential Evolution (DE) and Nature Inspired (NI) Techniques such as Particle Swarm Optimization Techniques (PSO) etc. as well as Hybrid Algorithms.