Verma and Parouha
Journal of Electrical Systems and Inf Technol (2021) 8:10
https://doi.org/10.1186/s43067-021-00032-z

Journal of Electrical Systems and Information Technology

RESEARCH

Open Access

An advanced hybrid meta-heuristic algorithm for solving small- and large-scale engineering design optimization problems

Pooja Verma* and Raghav Prasad Parouha

*Correspondence: pooja.v0903@gmail.com
Indira Gandhi National Tribal University, Amarkantak, M.P., India

Abstract

An advanced hybrid algorithm (*h*aDEPSO) is proposed in this paper for small- and large-scale engineering design optimization problems. Suggested advanced, differential evolution (aDE) and particle swarm optimization (aPSO) integrated with proposed *h*aDEPSO. In aDE a novel, mutation, crossover and selection strategy is introduced, to avoid premature convergence. And aPSO consists of novel gradually varying parameters, to escape stagnation. So, convergence characteristic of aDE and aPSO provides different approximation to the solution space. Thus, *h*aDEPSO achieve better solutions due to integrating merits of aDE and aPSO. Also in *h*aDEPSO individual population is merged with other in a pre-defined manner, to balance between global and local search capability. The performance of proposed *h*aDEPSO and its component aDE and aPSO are validated on 23 unconstrained benchmark functions, then solved five small (structural engineering) and one large (economic load dispatch)-scale engineering design optimization problems. Outcome analyses confirm superiority of proposed algorithms over many state-of-the-art algorithms.

Keywords: Global optimization, Small- and large-scale optimization, Meta-heuristics, Hybrid algorithm

Introduction

The success of any optimization algorithm majorly depends on its proficiency to solve the complex engineering design optimization problems. Most of the design optimization problems in engineering are turning out to be complicated due to involving mixed (discrete and continuous) variables under complex constraints. Generally, these problems are small- and large-scale nonlinear constrained problems and hence cannot be solved by traditional methods efficiently. Also, these problems can be represented as follows mathematically.

$$\begin{gathered}\text{minimization/maximization}\, f(x), x = \left(x_1, x_2, \ldots, x_j\right) \in R^D \\ g_l(x) \le 0, l = 1, 2, \ldots, L;\, h_k(x) = 0, k = 1, 2, \ldots, K;\, l_j \le x_j \le u_j, j = 1, 2, \ldots, D\end{gathered} \tag{1}$$

where f: real-valued function, g_l: inequality constraint and h_k: equality constraint (these may be linear or nonlinear), $x \in R^D$: D-dimensional decision vector, l_j and u_j: lower and upper limits for jth decision vector. L and K: total number of inequality and equality constraints.

Many conventional optimization algorithms like Newton or quasi-Newton have been developed to solve engineering design optimization problems. However, they 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 meta-heuristics algorithms (MAs) has been introduced to solve complex engineering design optimization problems. According to the mechanical differences, the MAs can be categorized into four groups as follows: swarm intelligence algorithms (SIAs)—inspired from behavior of social insects or animals, evolutionary algorithms (EAs)—inspired from biology, physics-based algorithms (PBAs)—inspired by the rules governing a natural phenomenon and human behavior-based algorithms (HBAs)—inspired from the human being. Some instances of these algorithms are listed as follows.

(i) Swarm intelligence algorithms (SIAs): PSO (Particle Swarm Optimization) [2], ABC (Artificial Bee Colony Algorithm) [3], GWO (Grey wolf optimizer HHO) [4], (Harris hawks optimization: Algorithm and applications) [5], CS (Cuckoo Search) [6], DA (Dragonfly Algorithm) [7], KH (Krill Herd) [8], etc.
(ii) Evolutionary algorithms (EAs): DE (Differential Evolution) [9] and GA (Genetic Algorithm) [10] are representatives of EAs.
(iii) Physics-based algorithms (PBAs): GSA (Gravitational Search Algorithm) [11], EO (Equilibrium optimizer) [12], HS (Harmony Search) [13], WCA (Water Cycle Algorithm) [14], etc.
(iv) Human behavior-based algorithms (HBAs): TLBO (Teaching–learning-based optimization) [15], MBA (Mine blast algorithm) [16] and so on.

Among many MAs, DE and PSO have been widely used in continuous/discrete, constrained as well as constrained/unconstrained optimization problems. 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 DE has been designed in the literature like JADE [17], SADE [18], CDE [19], modified DE [20] and DSS-MDE [21]. Also, PSO has attracted attention to solve many complex optimization problems due to its efficient search ability and simplicity. However, main drawback of PSO is that it may easily get stuck at a local optimal solution region. Therefore, accelerating convergence speed and avoiding local optimal solutions are two critical issues in PSO. To overcome such issues many modified PSO are proposed in the literature such as HEPSO [22], RPSOLF [23], CPSO [24], IPSO [25], QPSO [26], PSO-OPS [27], MTVPSO [28], IPSO [29], MPSO-TVAC [30], IPSO-TVAC [31], θ-PSO [32] and MPSO [33]. Furthermore, hybrid strategy is one of the main research directions to improve the performance of single algorithm. Therefore, to enhance the performance of DE and PSO, lots of their hybrids are presented in the literature like FAPSO [34], PSOSCALF [35], CSDE [36], PSOSCANMS [37], DEPSO [38] and DPD [39].

Although a large number of MAs are introduced in the literature, they could not able to solve variety of problems [40]. In other words, a method may have the acceptable results for some problems, but not for others. Thus, there is a need to introduce some effective algorithms to solve a wider range of optimization problems. Also, hybrid techniques are now more favored over their individual effort. Hence, it is the motivation of this study to present novel variants of DE and PSO with their hybridization.

Moreover, after an extensive literature review on different variants of DE and PSO with their hybridization, following points are analyzed and motivated from them.

(i) In DE mutation and crossover strategy with their associate control parameters utilized to produce global best solution and beneficial for improving convergence behavior. Therefore, appropriate strategies and their associated parameter values of DE are considered a vital research study.
(ii) The performance of PSO greatly depends on its parameters like acceleration coefficients (guide particles to the optimum) and inertia weight (balancing diversity). Hence, many researchers have tried to modify control parameter of PSO to achieve better accuracy and higher speed.
(iii) Hybrid algorithms have aroused interest of researchers due to its effectiveness for complex optimization problems. Since DE and PSO have complementary properties, therefore their hybrids has gained prominence recently. To best of our knowledge, finding ways to combine DE and PSO is still an open problem.

Motivated by above observations and literature survey, following major contributions have been outlined for solving small- and large-scale engineering design optimization problems. Small-scale engineering design optimization problems include welded beam design (WBD), three-bar truss design (TRD), pressure vessel design (PVD), speed reducer design (SRD) and tension/compression spring design (T/CSD), whereas in large-scale engineering design optimization problem namely economic load dispatch (ELD) with or without valve-point effects considering 3-, 6-, 15-, 40- and 140-unit test system.

(i) Developed an advanced differential evolution (aDE) where combination of novel strategies with their associated parameter values 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 by hybridizing advanced DE and PSO (*h*aDEPSO: hybridization of aDE and aPSO).

Methods

In this section, following proposed methodology has been described in detail: (i) advanced differential evolution (aDE), (ii) advanced particle swarm optimization (aPSO) and (iii) *hybrid h*aDEPSO.

Advanced differential evolution (aDE)

In suggested advanced DE (aDE), modified mutation strategy and crossover rate as well as changed selection scheme are introduced as follows.

Mutation:

$$v_{i,j}^{t} = x_{i,j}^{t} + \tau \times rand(0,1) \times \left(best_j - x_{i,j}^{t}\right) \tag{2}$$

where $x_{i,j}^{t}$: target vector, $v_{i,j}^{t}$: mutant vector, $rand(0,1)$: uniformly spread random number between 0 and 1, $best_j$: best vector and τ: convergence factor (elects searching scale of all vectors). The dynamic adjustments of convergence factor (τ) are given as follows. (i) If $\tau = 1$, then a vector will be randomly generated in the range $[x_{i,j}^{t}, best_j]$. This can improve convergence rate of DE, but it may take risk of increasing probability of encountering local optima and (ii) if $\tau = \mu(1 - t/t_{max}) + 1$, where t and t_{max}: current and total iteration, μ: positive constant (determining the maximal searching scale of all vectors). In 1st iteration, $\tau \approx \mu + 1$ (as $t = 1$ is much smaller than t_{max}, then term t/t_{max} can be ignored). In *max* iteration, $\tau = 1$ (as $(1 - t/t_{max}) = 0$). Therefore, τ decreases linearly from $\mu + 1$ to 1 during the whole optimization process. This can improve the convergence as well as avoid local optima.

Since τ is composed of a series of large values and enlarges the exploring scale of all vectors earlier, whereas, τ composed a series of large values earlier and small values later which ensure global and local search capacity respectively. Also, it guaranteed to exploring the search space for all vectors of the proposed mutation strategy. These ensure global and local search capacity as well as exploring search space of all vectors of the proposed mutation strategy.

Crossover:

$$u_{i,j}^{t}\,(\text{trial vector}) = \begin{cases} v_{i,j}^{t}; & \text{if } rand(0,1) \le C_r(\text{crossover rate}) \\ x_{i,j}^{t}; & \text{otherwise} \end{cases} \tag{3}$$

In order to keep the global searching ability and improve convergence speed C_r is set as $e^{\frac{(t - t_{max})}{t_{max}}}$. It guarantees 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.

Selection: It emphasizes on the random nature of aDE which is formulated as follows.

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^{t}; & \text{if } f\left(u_{i,j}^{t}\right) > f\left(x_{i,j}^{t}\right) \text{ and } rand(0,1) < p \\ u_{i,j}^{t}; & \text{otherwise} \end{cases} \tag{4}$$

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

Begin
initialize population x_{ij}^{t}; $\forall\ i = 1,2,\ldots,np$ randomly
for $t = 1$
compute $\tau = \mu(1 - t/t_{max}) + 1$ and $C_r = e^{\frac{(t-t_{max})}{t_{max}}}$
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 $v_{i,j}^{t}$ as
$v_{i,j}^{t} = x_{i,j}^{t} + \tau \times rand(0,1) \times (best_j - x_{i,j}^{t})$
///**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

Advanced particle swarm optimization (aPSO)

Preferably, PSO needs strong exploration ability and exploitation capability at early and later phase of the evolution, respectively. In velocity update equation of PSO, inertia weight (w) and acceleration coefficient ($c_1$and c_2) are important factors to satisfy the above requirement with following concept.

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

Considering all concerns like advantages, disadvantages and parameter influences of PSO, an advanced PSO (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

$$w = w_f + (w_i - w_f)\left(\frac{t}{t_{max}}\right)^2;\ c_1 = c_{1f}\left(\frac{c_{1i}}{c_{1f}}\right)^{\left(\frac{t}{t_{max}}\right)^2}$$
$$\text{and } c_2 = c_{2f}\left(\frac{c_{2i}}{c_{2f}}\right)^{\left(\frac{t}{t_{max}}\right)^2} \quad (5)$$

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, velocity and position of the ith particle are updated by following equations in the proposed aPSO. The pseudocode of proposed aPSO is presented below.

$$v_{i,j}^{t+1} = \left(w_f + (w_i - w_f)\left(\frac{t}{t_{max}}\right)^2\right)v_{i,j}^t + \left(c_{1f}\left(\frac{c_{1i}}{c_{1f}}\right)^{\left(\frac{t}{t_{max}}\right)^2}\right)r_1\left(p_{besti,j}^t - x_{i,j}^t\right) + \left(c_{2f}\left(\frac{c_{2i}}{c_{2f}}\right)^{\left(\frac{t}{t_{max}}\right)^2}\right)r_2\left(g_{bestj}^t - x_{i,j}^t\right) \tag{6}$$

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

```
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. (6)
      update position of particles by eq. (7)
    end
  end
end
t = t + 1
End
```

Hybrid advanced DEPSO (*h*aDEPSO)

An advanced hybrid algorithm (*h*aDEPSO) is proposed to further improve solution quality. In *h*aDEPSO, entire population is sorted according to 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 contain 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 the 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-populations and 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). Basically, *h*aDEPSO is based on relating superior capability of suggested aDE and aPSO. The flowchart of *h*aDEPSO is demonstrated in Fig. 1 and the pseudocode described below.

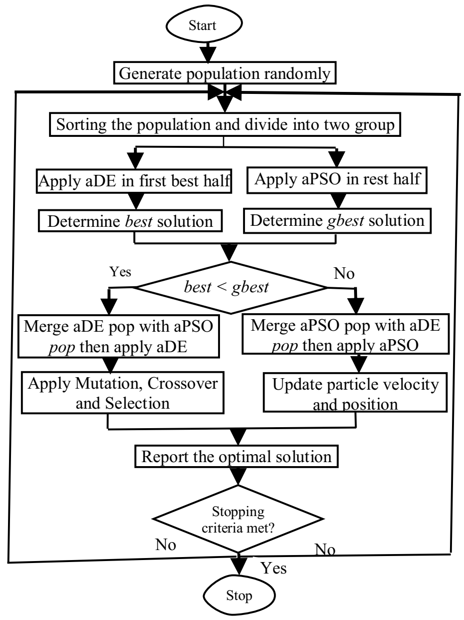


Fig. 1 Flowchart of *h*aDEPSO

```
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 is met then
      stop the process
    else
      repeat the process
    end
  end
end
t = t + 1
End
```

Implementation of *h*aDEPSO for optimization

The stepwise implementation of the proposed *h*aDEPSO for Sphere function (continuous, convex and unimodal) is demonstrated in this section. The initial parameters for the considered function are given as follows.

(i) Mathematical formulation: Minimize$f(x) = \sum_{i=1}^{D} x_i^2$

(ii) Range of decision variables: $-5 \leq x_i \leq 5$

(iii) Number of decision variables = 2

(iv) Population size (np): 10

(v) Number of iteration (t): 10

Stepwise execution of the proposed haDEPSO for Sphere function

for iteration $t = 1$.

Step I: Initialization.

Generate a random initial population and evaluate the corresponding objective function value

$$\begin{bmatrix} x_1 & x_2 \\ 0.8532 & -1.8670 \\ -4.7351 & 0.1238 \\ -1.0178 & 0.1578 \\ -4.1070 & -3.2185 \\ 2.2213 & 1.0189 \\ -3.0080 & -4.5791 \\ -2.8943 & 4.7913 \\ 4.0218 & 2.9810 \\ -3.1197 & -1.7298 \\ -19326 & 3.7452 \end{bmatrix} = \begin{bmatrix} f(x) \\ 4.2136 \\ 22.4364 \\ 1.1937 \\ 27.2261 \\ 5.9723 \\ 30.0162 \\ 31.3335 \\ 25.0612 \\ 12.7247 \\ 17.7614 \end{bmatrix}$$

required parameter:

for aDE $\tau = \mu(1 - t/t_{max}) + 1 = 0.01(1 - 1/10) + 1 = 1.009$, $best_j = [-1.0178\ 0.1578]$, $C_r = 0.4065$ and $p \in rand(0, 1]$.
for aPSO $w = 0.8995$, $c_1 = 2.4600$, $c_2 = 0.508$, $g_{bestj}^1 = [-4.7351\ 0.1238]$ and initial velocity ($v_{i,j}^t$) generated randomly in between 0 and 1 is

$$v_{i,j}^t = \begin{bmatrix} 0.2394 & 0.2218 \\ 0.1908 & 0.2419 \\ 0.2818 & 0.5319 \\ 0.3976 & 0.4134 \\ 0.1934 & 0.2357 \end{bmatrix}$$

Step II: Sorting.

Arranging the population according to the fitness function values and dividing into two groups as pop_1 and pop_2.

$$\begin{bmatrix} -1.0178 & 0.1578 \\ 0.8532 & -1.8670 \\ 2.2213 & 1.0189 \\ -3.1197 & -1.7298 \\ -1.9326 & 3.7452 \\ -4.7351 & 0.1238 \\ 4.0218 & 2.9810 \\ -4.1070 & -3.2185 \\ -3.0080 & -4.5791 \\ -2.8943 & 4.7913 \end{bmatrix} = \begin{bmatrix} 1.1937 \\ 4.2136 \\ 5.9723 \\ 12.7247 \\ 17.7614 \\ 22.4364 \\ 25.0612 \\ 27.2261 \\ 30.0162 \\ 31.3335 \end{bmatrix}$$

$$pop_1 \begin{bmatrix} -1.0178 & 0.1578 \\ 0.8532 & -1.8670 \\ 2.2213 & 1.0189 \\ -3.1197 & -1.7298 \\ -1.9326 & 3.7452 \end{bmatrix} = \begin{bmatrix} 1.0608 \\ 4.2136 \\ 5.9723 \\ 12.7247 \\ 17.7614 \end{bmatrix}$$

$$\text{and } pop_2 \begin{bmatrix} -4.7351 & 0.1238 \\ 4.0218 & 2.9810 \\ -4.1070 & -3.2185 \\ -3.0080 & -4.5791 \\ -2.8943 & 4.7913 \end{bmatrix} = \begin{bmatrix} 22.4364 \\ 25.0612 \\ 27.2261 \\ 30.0162 \\ 31.3335 \end{bmatrix}$$

Step III: Applying.

aDE in pop_1 and aPSO in pop_2.

Advanced differential evolution (aDE) for *pop*$_1$

$$\text{Mutation: } \left(v_{i,j}^t = x_{i,j}^t + \tau \times rand(0,1) \times \left(best_j - x_{i,j}^t\right)\right)$$

$$\begin{bmatrix} -1.0178 & 0.1578 \\ 1.1079 & -1.2595 \\ 3.2092 & 0.8046 \\ -2.5053 & -1.1635 \\ -1.6743 & 2.6689 \end{bmatrix}$$

$$\text{Crossover } \left(u_{i,j}^t = \begin{cases} v_{i,j}^t; & \text{if } rand(0,1) \le C_r \\ x_{i,j}^t; & \text{otherwise} \end{cases}\right)$$

$$\begin{bmatrix} -1.0178 & 0.1578 \\ 1.1079 & -1.2595 \\ 2.2213 & 0.8046 \\ -2.5053 & -1.1635 \\ -1.6743 & 2.6689 \end{bmatrix} = \begin{bmatrix} 1.0608 \\ 2.8137 \\ 5.5815 \\ 7.6302 \\ 10.8579 \end{bmatrix}$$

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

$$\begin{bmatrix}-1.0178 & 0.1578\\ 1.1079 & -1.2595\\ 2.2213 & 0.8046\\ -2.5053 & -1.1635\\ -1.6743 & 2.6689\end{bmatrix}=\begin{bmatrix}1.0608\\ 2.8137\\ 5.5815\\ 7.6302\\ 10.8579\end{bmatrix}$$

Best function value obtained by aDE is 1.0608 named as *best* on pop_1.

Advanced particle swarm optimization (aPSO) for pop_2

Updated velocity $v_{i,j}^{t}$ $\left(v_{i,j}^{t}=wv_{i,j}^{t}(\text{old})+c_1r_1\left(p_{besti,j}^{t}-x_{i,j}^{t}\right)+c_2r_2\left(g_{bestj}^{t}-x_{i,j}^{t}\right)\right)$ and updated position $x_{i,j}^{t}\left(x_{i,j}^{t}=x_{i,j}^{t}(\text{old})+v_{i,j}^{t}(\text{updated})\right)$ of aPSO are given as follows.

$$v_{i,j}^{t}=\begin{bmatrix}0.2153 & 0.1995\\ -1.6080 & -0.3631\\ 0.2274 & 1.1575\\ 0.0061 & 1.3276\\ -1.6669 & -0.7366\end{bmatrix}\text{ and } x_{i,j}^{t}=\begin{bmatrix}-4.5197 & 0.3233\\ 2.4137 & 2.6178\\ -3.8795 & -2.0610\\ -3.0014 & -3.2515\\ -4.5612 & 4.0547\end{bmatrix}=\begin{bmatrix}20.5322\\ 12.6788\\ 19.2982\\ 19.5806\\ 37.2451\end{bmatrix}$$

Best function value obtained by aPSO is 12.6788 named as *gbest* on pop_2.

Step IV: Condition

best < *gbest* then merging pop_2 with pop_1 and applying again aDE else merging pop_1 with pop_2 then applying aPSO (*best* < *gbest* condition is applicable for iteration $t=1$).

$$\text{Merged population }(x_{i,j}^{t})=\begin{bmatrix}-1.0178 & 0.1578\\ 1.1079 & -1.2595\\ 2.2213 & 0.8046\\ -2.5053 & -1.1635\\ -1.6743 & 2.6689\\ -4.5197 & 0.3233\\ 2.4137 & 2.6178\\ -3.8795 & -2.0610\\ -3.0014 & -3.2515\\ -4.5612 & 4.0547\end{bmatrix}=\begin{bmatrix}1.0608\\ 2.8137\\ 5.5815\\ 7.6302\\ 10.8579\\ 20.5322\\ 12.6788\\ 19.2982\\ 19.5806\\ 37.2451\end{bmatrix}$$

Mutation

$$v_{i,j}^{t}=\begin{bmatrix}-1.0178 & 0.1578\\ 0.0450 & -0.5508\\ 1.6195 & 0.4812\\ -1.7615 & -0.5028\\ -1.4752 & 1.4133\\ 2.7687 & 0.2405\\ 0.6980 & 1.3878\\ -2.4487 & -0.9516\\ -2.0096 & -1.5468\\ -2.7895 & 2.1062\end{bmatrix}$$

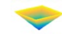
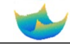
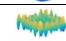
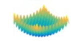
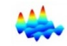
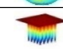
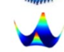
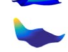
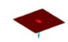
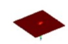
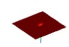
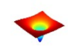
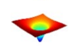
Crossover

$$u_{i,j}^{t}=\begin{bmatrix}-1.0178 & 0.1578\\ 0.0450 & -0.5508\\ 1.6195 & 0.4812\\ -1.7615 & -0.5028\\ -1.4752 & 1.4133\\ -2.5053 & 0.2405\\ 0.6980 & 2.6178\\ -2.4487 & -0.9516\\ -2.0096 & -1.5468\\ -4.5612 & 2.1062\end{bmatrix}=\begin{bmatrix}1.0608\\ 0.3054\\ 2.8543\\ 3.3556\\ 4.1736\\ 6.3343\\ 7.3400\\ 6.9016\\ 6.4310\\ 25.2406\end{bmatrix}$$

Selection

$$x_{i,j}^{t}=\begin{bmatrix}-1.0178 & 0.1578\\ 0.0450 & -0.5508\\ 1.6195 & 0.4812\\ -1.7615 & -0.5028\\ -1.4752 & 1.4133\\ 2.5053 & 0.2405\\ 0.6980 & 2.6178\\ -2.4487 & -0.9516\\ -2.0096 & -1.5468\\ -4.5612 & 2.1062\end{bmatrix}$$

Table 1 Unconstrained benchmark functions (UBFs)

fn	Formulation	Type	Dim.	Range	f_{min}	Graph
f1	$\sum_{i=1}^{D} x_i^2$		30	[-100,100]	0	
f2	$\sum_{i=1}^{D}\lvert x\rvert + \prod_{i=1}^{D}\lvert x_i\rvert$		30	[-10, 10]	0	
f3	$\sum_{i=1}^{D}\left(\sum_{j=1}^{i} x_j\right)^2$		30	[-100,100]	0	
f4	$max_i\lvert x_i\rvert, 1 \le i \le D$	unimodal	30	[-100,100]	0	
f5	$\sum_i^{D-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$		30	[-30, 30]	0	
f6	$\sum_i^D(\lfloor x_i+0.5\rfloor)^2$		30	[-100,100]	0	
f7	$(\sum_i^D i\,x_i^4)+rand[0,1)$		30	[-1.28, 1.28]	0	
f8	$\sum_i^D -x_i sin\left(\sqrt{\lvert x_i\rvert}\right)$		30	[-500,500]	-418.9829*D	
f9	$\sum_i^D[x_i^2-10cos(2\pi x i)+10]$		30	[-5.12,5.12]	0	
f10	$-20exp\left(-{}^{0.2}\sqrt{\frac{1}{D}\sum_i^D {x_i}^2}\right)-\exp\left(\frac{1}{D}\sum_i^D cos2\pi x i\right)+20+e$	multimodal	30	[-32,32]	0	
f11	$\frac{1}{4000}\sum_i^D x_i^2-\prod_i^D cos\left(\frac{x_i}{\sqrt{i}}\right)+1$		30	[-600, 600]	0	
f12	$\frac{\pi}{D}\left\{10sin^2(xy)+\sum_i^D(x_i-1)^2\left(1+sin^2(xy_{i+1})\right)+(y_D-1)^2\right\}+\sum_i^D U(x_i,10,100,4)$		30	[-50,50]	0	
f13	$0.1\{sin^2(3\pi x_i)\}+\sum_{i=1}^D(x_i-1)^2[1+sin^2(3\pi x_i+1)]+(x_D-1)^2+\sum_i^D U(x_i,5,100,4)$		30	[-50,50]	0	
f14	$\left(\frac{1}{500}+\sum_{j=1}^{25}\left(\frac{1}{j+1+\sum_{i=0}^{1}(x_i-a_{ij})^6}\right)\right)^{-1}$		2	[-65, 65]]	1	
f15	$\sum_{i=0}^{10}\left(a_i-\frac{x_0(b_i^2+b_ix_1)}{(b_i^2+b_ix_2+x_3)}\right)^2$		4	[-5,5]	0.00030	
f16	$4x_0^2-2.1x_0^4+\frac{1}{3}x_0^6+x_0x_1-4x_1^2+4x_1^4$		2	[-5,5]	−1.0316	
f17	$\left(x_1-\frac{5.1}{4\pi^2}x_0^2+\frac{5}{\pi}x_0-6\right)^2+10\left(1-\frac{1}{8\pi}\right)cosx_0+10$		2	[-5,5]	0.398	
f18	$1+(x_0+x_1+1)^2(19-14x_0+3x_0^2-14x_1-6x_0x_1+3x_1^2)\{30+(2x_0-3x_1)^2(18-32x_0+12x_0^2+48x_1-36x_0x_1+27x_1^2)\}$		2	[-2,2]	3	
f19	$-\sum_{i=1}^4 c_iexp\left(-\sum_{j=1}^3 a_{ij}(x_j-p_{ij})^2\right)$	fixed-dimension	3	[1,3]	−3.86	
f20	$-\sum_{i=1}^4 c_iexp\left(-\sum_{j=1}^6 a_{ij}(x_j-p_{ij})^2\right)$		6	[0,1]	−3.32	
f21	$-\sum_{i=1}^5((x-a_i)^T(x-a_i)+c_i)^{-1}$		4	[0,10]	−10.1532	
f22	$-\sum_{i=1}^7((x-a_i)^T(x-a_i)+c_i)^{-1}$		4	[0,10]	−10.4028	
f23	$-\sum_{i=1}^{10}((x-a_i)^T(x-a_i)+c_i)^{-1}$		4	[0,10]	−10.5363	

Step V: Termination criterion

Stop if optimal solution obtained otherwise repeat go to step II.

Similarly

for iteration $t=5$

$$\begin{bmatrix} -0.5984 & 0.1001 \\ 1.2541 & -0.1241 \\ 1.7845 & 1.0012 \\ -0.4512 & -1.5231 \\ -1.3251 & 0.1241 \\ -3.0025 & 1.0154 \\ 1.7740 & 3.0015 \\ -2.9632 & -0.0214 \\ -1.1254 & -1.7458 \\ -3.4125 & 2.0325 \end{bmatrix}$$

for iteration $t=10$

$$\begin{bmatrix} -0.1254 & 0.1201 \\ 0.2514 & -1.0215 \\ 1.1245 & 1.3251 \\ -0.6584 & -0.5125 \\ -1.7584 & 0.2141 \\ -2.1254 & 0.0100 \\ 2.1102 & 3.0102 \\ -3.1251 & -0.1542 \\ -1.3695 & -0.3251 \\ -2.0321 & 1.1254 \end{bmatrix}$$

Results and discussion

The validation of the proposed algorithms with the solution of small- and large-scale engineering design optimization problems is included in this section.

Table 2 Parameter setting of compared and proposed algorithms for UBFs

Algorithm	Year	References	Control parameter		Population size	Stopping criterion	Run
			Term	Values			
EO	2019	[12]	a_1, a_2 and GP	{1, 1.5, 2, 2.5, 3}, {0.1, 0.5, 1, 1.5, 2} and (0.1.0.25, 0.5, 0.75, 0.9}	30	500	30
HHO	2019	[5]	Escaping energy	$E < 0.5, E \geq 0.5$	30	500	30
JADE	2009	[17]	F_i and CR_i,	randc$_i$(μ_F, 0.1) and randn$_i$(μ_{CR}, 0.1)	50	1000	30
SHADE	2013	[18]	*P*best and Arc rate	0.1 and 2	30	500	30
HEPSO	2014	[22]	P_C and P_B	0.95 and 0.02	50	500	30
RPSOLF	2017	[23]	w, c_1, c_2, c_3 and β, ε	0.55, 1.49, 1.49, 1.5 and 0.99	50	500	30
FAPSO	2018	[34]	–	–	50	5000	30
PSOSCALF	2018	[35]	$w_{min}, w_{max}, c_{1min}, c_{1max}, c_{2min}, c_{2max}$ and β	0.4, 0.9, 0.5, 2.5, 0.5, 2.5 and 1.5	50	500	30
aDE	Proposed		–	–	30	500	30
aPSO			$w_i, w_f, c_{1i}, c_{1f}, c_{2i}$ and c_{2f}	0.4, 0.9, 0.5, 2.5, 2.5 and 0.5	30	500	30
*h*aDEPSO			–	–	30	500	30

Validation of proposed algorithms

The justification of suggested component aDE and aPSO with the proposed *h*aDEPSO algorithm evaluated on 23 unconstrained benchmark functions (UBFs). The descriptions of UBFs are listed in Table 1.

Simulations were conducted on Intel (R) Core (TM) i5-2350 M CPU @ 2.30 GHz, RAM: 4.00 GB, Operating System: Window 10, C-free Standard 4.0. After an extensive analysis, the parameters used in the proposed algorithms are fine-tuned and recommended as follows.w_i (initial inertia weight) = 0.4, c_{1i} (initial cognitive acceleration coefficient) = 0.5 and c_{2i} (initial social acceleration coefficient) = 2.5 and w_f (final inertia weight) = 0.9, c_{1f} (final cognitive acceleration coefficient) = 2.5 and c_{2f} (final social acceleration coefficient) = 0.5. Moreover, to handle constraints constrained optimization problems transfers to unconstrained one by adding penalty term into objective function. The bracket operator penalty [41] is picked in the present study due to its higher efficiency. And after many experiments fine-tuning value of R = 1e^{03} is acclaimed to use in proposed algorithms. The overall best values in each table are highlighted with boldface letters of the corresponding algorithms.

The produced result by proposed algorithms on 23 unconstrained benchmark functions (UBFs) is compared with traditional algorithms (HHO [5] and EO [12]), DE variants (JADE [17] and SHADE [18]), PSO variants (HEPSO [22] and RPSOLF [23]) and hybrid variants (FAPSO [34] and PSOSCALF [35]). The parameters of all above compared and proposed algorithms are listed in Table 2. For fair comparison population size, stopping criteria and independent run of proposed algorithms is taken as minimum of corresponding comparative algorithms. The comparative experimental results in terms of mean, std. (standard deviation) and ranking of the objective function values are presented in Table 3 of 30 independent runs.

It should be noted that from Table 3, the mean objective function values of the proposed aDE, aPSO and *h*aDEPSO algorithms are better and/or equal in comparison of

Table 3 Simulation results on UBFs

f_n	Criteria	Algorithms										
		Traditional algorithms		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		EO	HHO	JADE	SHADE	HEPSO	RPSOLF	PSOSCALF	FAPSO	aDE	aPSO	*ha*DEPSO
f_1	Mean	3.32e−40	2.03e+00	1.87e−31	1.42e−09	16.26772	5.065e−269	1.11014e−20	2.87e−127	**0.00e+000**	**0.00e+000**	**0.00e+000**
	std	6.78e−40	4.04e−01	6.43e−31	3.09e−09	10.01293	**0.00e+00**	1.83289e−20	1.76e−127	**0.00e+000**	**0.00e+000**	**0.00e+000**
	Rank	4	9	5	7	8	2	6	3	**1**	**1**	**1**
f_2	Mean	7.12e−23	1.70e+00	2.79e−15	0.0087	1.28424	1.000e−134	4.09460e−11	1.02e−17	**0.00e+000**	**0.00e+000**	**0.00e+000**
	std	6.36e−23	7.37e−02	9.51e−15	0.0213	0.41611	3.753e−134	5.68981e−11	1.43e−17	**0.00e+000**	**0.00e+000**	**0.00e+000**
	Rank	3	9	4	7	8	2	6	5	**1**	**1**	**1**
f_3	Mean	8.06e−09	1.17e+02	1.10e−03	15.4352	7.423e+03	7.791e−249	2.16858e−12	1.68e−11	**0.00e+000**	0.17e−129	**0.00e+000**
	std	1.60e−08	5.28e+00	5.14e−03	9.9489	7.423e+03	0.00e+00	1.03815e−11	2.49e−11	**0.00e+000**	1.67e−131	**0.00e+000**
	Rank	6	9	7	8	10	2	4	5	**1**	3	**1**
f_4	Mean	5.39e−10	2.05e+00	1.66e−03	0.9796	23.95145	1.937e−157	8.47410e−08	4.09e+03	3.28e−101	8.35e−098	**0.00e+000**
	std	1.38e−09	7.40e−02	1.98e−03	0.7995	7.71460	1.061e−156	1.23324e−07	6.53e+02	7.12e−103	7.08e−098	**0.00e+000**
	Rank	5	9	7	8	10	2	6	11	3	4	**1**
f_5	Mean	2.53e+01	2.95e+00	1.18e+01	24.4743	2.380e+03	27.42672	21.97646	6.55e−11	1.25e−021	4.85e−012	**2.21e−033**
	std	0.16e+00	8.36e−02	1.57e+01	11.2080	1.852e+03	0.24848	0.54774	1.99e−11	1.85e−023	3.21e−012	**3.72e−037**
	Rank	7	9	10	6	11	8	5	4	2	3	**1**
f_6	Mean	8.29e−06	2.49e+00	4.59e−31	5.31e−10	21.55405	2.98244	7.13998e−12	2.37e−12	**0.00e+000**	1.75e−032	**0.00e+000**
	std	5.02e−06	8.25e−02	1.65e−30	6.35e−10	9.33263	0.23250	3.65884e−11	1.84e−13	**0.00e+000**	9.16e−035	**0.00e+000**
	Rank	8	9	3	7	6	10	5	4	**1**	2	**1**
f_7	Mean	1.17e−02	8.20e+00	6.49e−03	0.0235	0.12982	0.00104	0.00012	**0.00e+00**	2.19e−003	5.11e−001	1.07e−003
	std	6.54e−04	1.69e−01	2.48e−03	0.0088	0.09727	7.644e−04	0.00010	**0.00e+00**	1.10e−004	1.70e−002	1.40e−005
	Rank	6	11	5	9	10	7	4	**1**	3	8	2
f_8	Mean	− 9016.34	4.86e+00	**− 1.24e+04**	− 11,713.1	− 2.139e+03	− 3.254e+03	**− 12,569.48**	2.48e−11	**− 1.25e+004**	− 6.37e+004	**− 1.25e+004**
	std	595.1113	1.03e+00	1.27e+02	230.49	8.282e+02	2.860e+02	2.39996e−07	6.44e−12	1.07e−017	2.10e−001	**0.00e+000**
	Rank	6	4	**1**	8	5	7	**1**	2	**1**	9	**1**

Table 3 (continued)

f_n	Criteria	Algorithms										
		Traditional algorithms		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		EO	HHO	JADE	SHADE	HEPSO	RPSOLF	PSOSCALF	FAPSO	aDE	aPSO	*h*aDEPSO
f_9	Mean	**0.00e+00**	3.77e+00	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	**0.00e+00**	8.87e−01	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	**1**	3	2	4	5	**1**	**1**	**1**	**1**	**1**	**1**
f_{10}	Mean	8.34e−14	3.75e+00	1.31e−14	0.3957	2.83842	4.085e−15	2.24609e−11	4.86e−15	1.44e−015	1.97e−014	**2.88e−016**
	std	2.53e−14	8.75e−01	2.46e−14	0.5868	0.66134	1.084e−15	2.33542e−11	1.74e−15	**0.00e+000**	**0.00e+000**	**0.00e+000**
	Rank	7	8	5	10	9	3	8	4	2	6	**1**
f_{11}	Mean	**0.00e+00**	4.17e+00	2.87e−03	0.0048	1.16858	**0.00e+00**	**0.00e+00**	1.74e−16	**0.00e+000**	3.37e−111	**0.00e+000**
	std	**0.00e+00**	5.56e−01	7.85e−03	0.0077	0.12602	**0.00e+00**	**0.00e+00**	3.60e−16	**0.00e+000**	1.11e−119	**0.00e+000**
	Rank	**1**	7	4	5	6	**1**	**1**	3	**1**	2	**1**
f_{12}	Mean	7.97e−07	1.90e+01	1.73e−02	0.0346	0.47856	0.26157	8.46465e−14	1.57e−32	1.05e−032	3.34e−002	**3.73e−033**
	std	7.69e−07	3.31e+00	7.74e−02	0.0875	0.22623	0.03386	2.79106e−13	0.00e+00	2.77e−034	1.02e−004	**3.18e−034**
	Rank	5	11	6	8	10	9	4	3	2	7	**1**
f_{13}	Mean	0.029295	1.89e+01	5.45e−24	7.32e−04	1.85056	2.05282	0.00399	**1.58e−32**	2.09e−021	9.30e−004	**1.58e−032**
	std	0.035271	1.56e+00	2.58e−23	0.0028	0.65246	0.16579	0.00928	**0.00e+00**	3.27e−023	3.71e−004	**1.89e−043**
	Rank	7	10	2	4	8	9	6	**1**	3	5	**1**
f_{14}	Mean	**0.99800**	**9.98e−01**	**9.98e−01**	**0.998004**	**0.99800**	1.54064	1.13027	**9.98e−001**	**9.98e−001**	**9.98e−001**	**9.98e−001**
	std	1.54e−16	9.23e−01	**0.00e+00**	5.83e−17	9.219e−17	1.84429	0.50338	1.27e−08	**0.00e+000**	**0.00e+000**	**0.00e+000**
	Rank	**1**	**1**	**1**	**1**	**1**	3	2	**1**	**1**	**1**	**1**
f_{15}	Mean	0.00239	3.10e−04	3.01e−03	0.002374	6.404e−04	0.00171	3.13244e−04	3.95e−04	3.83e−004	3.99e−004	**3.02e−004**
	std	0.00609	1.97e−04	6.92e−03	0.0061	2.801e−04	0.00508	2.17489e−05	6.02e−08	9.23e−012	2.96e−007	**2.38e−018**
	Rank	11	2	8	10	7	9	3	5	4	6	**1**
f_{16}	Mean	**− 1.03161**	**−1.03e+00**	**− 1.03e+00**	**− 1.03161**	**− 1.03161**	**− 1.03161**	**− 1.0316**	**−1.03e+00**	**−1.03e+000**	**−1.03e+000**	**−1.03e+000**
	std	6.04e−16	6.78e−16	6.78e−16	6.51e−16	3.554e−15	1.650e−05	4.40244e−16	**0.00e+00**	**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**	0.39787	**3.98e−001**	**3.98e−001**	**3.98e−001**	**3.98e−001**
	std	**0.00e+00**	2.54e−06	**0.00e+00**	3.24e−16	6.594e−13	5.267e−04	3.66527e−15	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**

Table 3 (continued)

f_n	Criteria	Algorithms										
		Traditional algorithms		DE variants		PSO variants		Hybrid variants		Proposed algorithms		
		EO	HHO	JADE	SHADE	HEPSO	RPSOLF	PSOSCALF	FAPSO	aDE	aPSO	*h*aDEPSO
	Rank	2	**1**	**1**	2	2	**1**	2	**1**	**1**	**1**	**1**
f_{18}	Mean	**3.00e+00**	**3.00e+00**	**3.00e+00**	**3.00e+00**	0.65246	**3.00002**	**3. 00e+00**	**3.00e+000**	**3.00e+000**	**3.00e+000**	**3.00e+000**
	std	1.56e−15	**0.00e+00**	1.82e−15	1.87e−15	5.146e−11	1.658e−05	5.96540e−13	5.31e−016	**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	**− 3.86278**	−3.87e+000	**−3.86e+000**	**−3.86e+000**	**−3.86e+000**
	std	2.59e−15	2.44e−03	2.71e−15	2.69e−15	1.008e−13	0.00283	8.31755e−15	2.11e−004	**0.00e+000**	3.36e−021	**0.00e+000**
	Rank	**1**	**1**	**1**	**1**	**1**	3	1	2	**1**	**1**	**1**
f_{20}	Mean	− 3.2687	**−3.322**	− 3.29e+00	− 3.27047	− 3.31803	− 3.10441	− 3.27168	− 3.29e+000	**− 3.32e+000**	− 3.27e+000	**− 3.32e+000**
	std	0.05701	0.137406	5.11e−02	0.0599	0.02170	0.15760	0.06371	**0.00e+000**	**0.00e+000**	**0.00e+000**	**0.00e+000**
	Rank	6	**1**	4	5	2	7	5	3	**1**	5	**1**
f_{21}	Mean	− 8.55481	**−10. 1531**	− 9.14e+00	− 9.2343	**− 10.15319**	− 4.76171	**− 10.15319**	− 7.51e+00	**− 1.01e+001**	− 9.87e+001	**− 1.01e+001**
	std	2.76377	0.885673	2.06e+00	1.3969	2.680e−05	0.73723	4.46227e−15	1.21e−01	**0.00e+000**	1.07e−011	**0.00e+000**
	Rank	5	**1**	4	3	**1**	6	**1**	3	**1**	2	**1**
f_{22}	Mean	− 9.3353	**−10.4015**	− 9.88e+00	− 10.2809	**− 10. 4015**	− 4.81927	**− 10. 40,156**	− 6.04e+01	**− 1.04e+001**	− 9.87e+000	**− 1.04e+001**
	std	2.43834	1.352375	1.61e+00	1.3995	0.01728	0.75699	1.80672e−15	2.14e−01	2.91e−012	2.08e−002	**1.10e−016**
	Rank	5	**1**	3	2	**1**	7	**1**	6	**1**	4	**1**
f_{23}	Mean	− 9.63655	**−10.5364**	− 1.03e+01	63.333	**− 10.53640**	− 5.06376	**− 10.53640**	− 5.64e+00	**− 1.05e+001**	− 9.13e+000	**− 1.05e+001**
	std	2.38811	0.927655	1.40e+00	80.872	5.845e−07	0.82968	4.84794e−15	5.70e−02	1.07e−021	2.05e−003	**0.00e+000**
	Rank	3	**1**	2	7	**1**	6	**1**	5	**1**	4	**1**
Sum of rank		102	119	87	124	125	107	75	75	35	78	**24**
Average		4.43	5.17	3.78	5.39	5.43	4.65	3.26	3.26	1.52	3.39	**1.04**
overall rank		6	8	5	9	10	7	3	3	2	4	**1**

Table 4 Statistical comparisons of proposed Vs other algorithms for UBFs

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

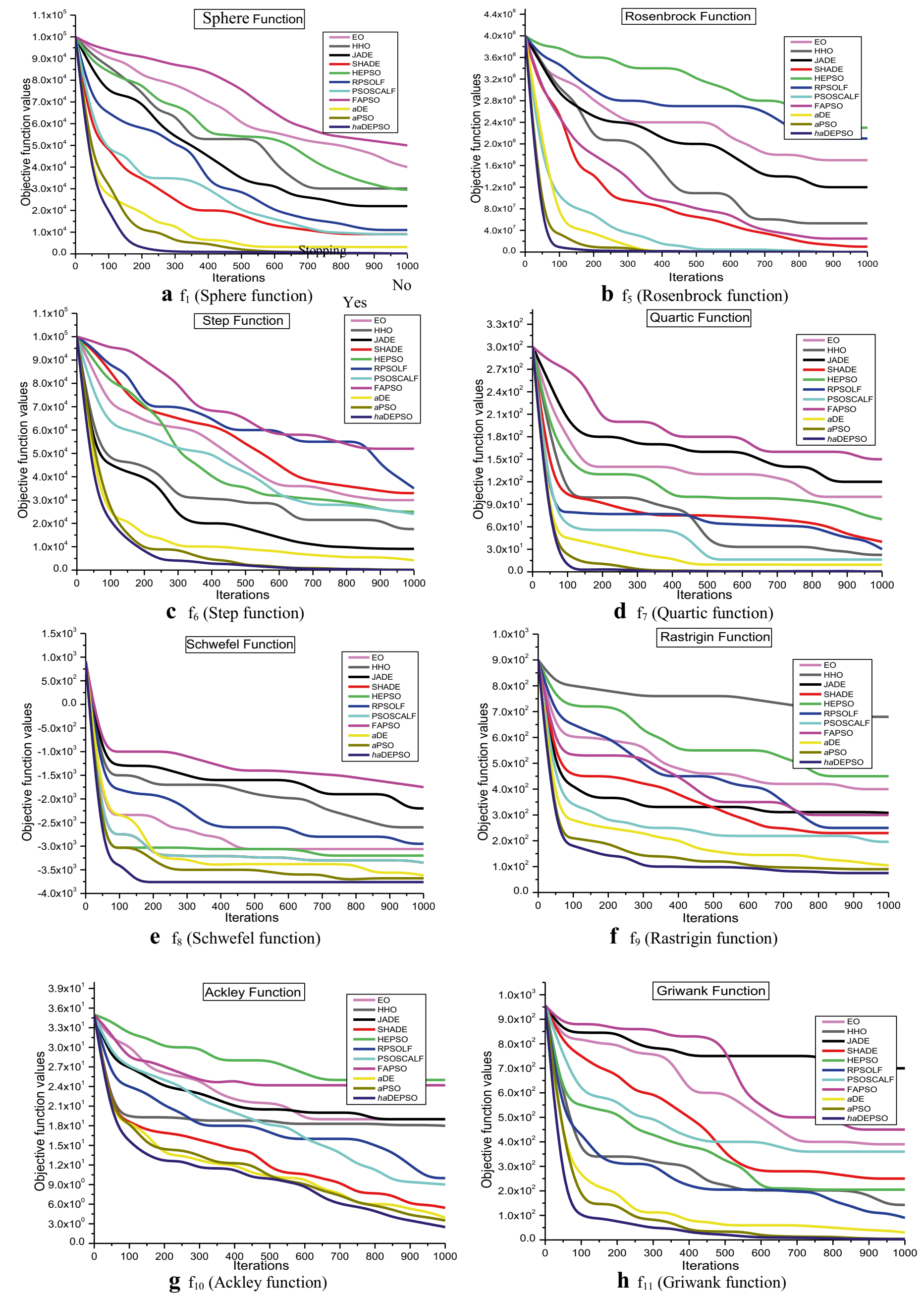


Fig. 2 Convergence graph of UBFs

above-listed traditional algorithms, DE variants, PSO variants and hybrid variants. As per the experimental results shown in Table 3, the following comparison results are summarized as follows for UBFs cases (i). Unimodal function (f_1–f_7): proposed *h*aDEPSO obtained better results in all functions (f_1–f_6) meanwhile slightly inferior on f_7, suggested aDE obtained better results for f_1, f_2, f_3 and f_6 functions whereas aPSO achieve better results for f_1 and f_2 as well as marginally similar for the rest functions. (ii). Multimodal function (f_8–f_{13}): proposed *h*aDEPSO obtained better results for all six functions (f_8–f_{13}) and similar for f_8 (on JADE and PSOSCALF), f_9 (on EO, RPSOLF, FAPSO and PSOSCALF), f_{11} (on EO,

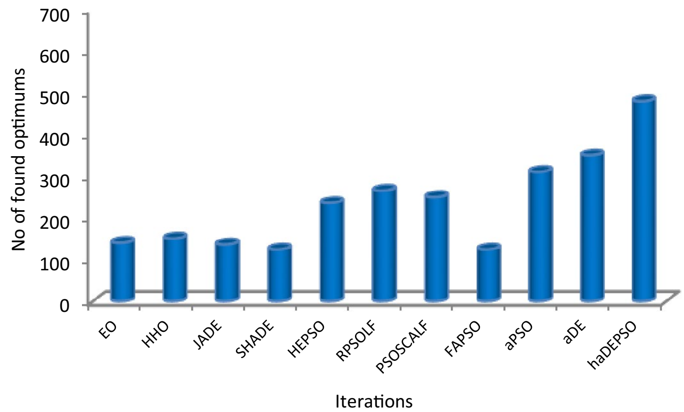


Fig. 3 Comparison of algorithms in finding the global optimal solution out of 690 runs

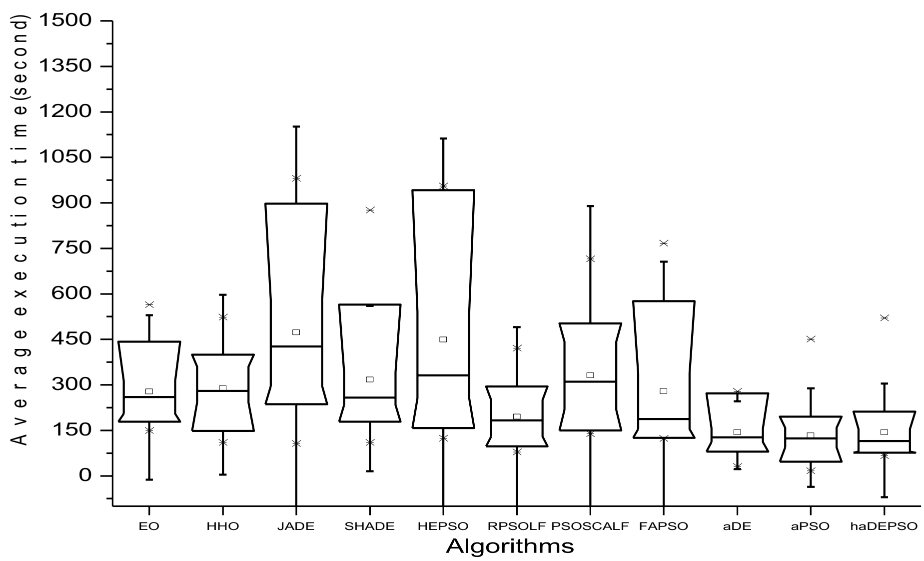


Fig. 4 Processing times of algorithms for UBFs

RPSOLF and PSOSCALF) and f_{13} (on 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 *ha*DEPSO 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 3 based on mean result values. From this table it is concluded that *ha*DEPSO, aDE and aPSO ranked 1st, 2nd and 4th sequentially. Also, average and overall rank of proposed algorithms Vs others is presented in Table 3. It is clear that (from ranking) performances of proposed algorithms are superior to others. Eventually, proposed aDE, aPSO and *h*aDEPSO produce less std. (it may 0.00E+00) for most of the cases on UBFs which describe their stability. Furthermore, superiority of proposed algorithms is statistically validated over other

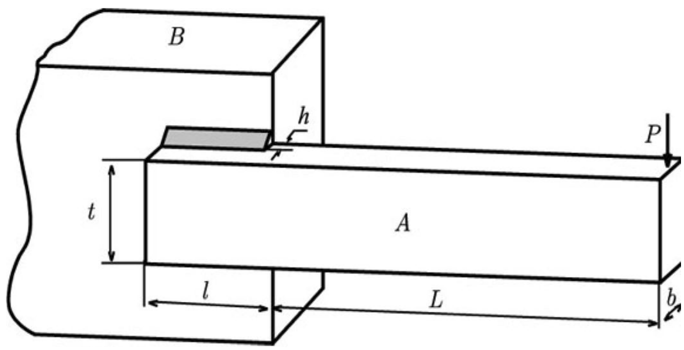


Fig. 5 Schematic diagram of welded beam design (WBD) problem

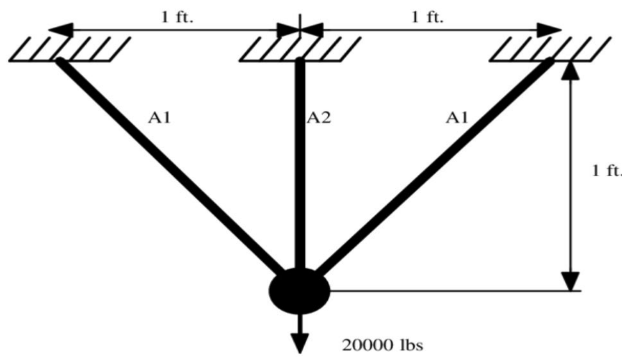


Fig. 6 Schematic diagram of three-bar truss design (TRD) problem

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 [42]. The results of t test and WSR test on UBFs are reported in Table 4. From Table 4 it can be seen that proposed algorithms have both ‘a (significantly better than other)’ and ‘a^+ (highly significance with other)’ sign (in case of t test) and perform better or equally (in case WSR test) in most of consequence. Also, the p values as reported in Table 4 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* UBFs. All plotted convergence graphs (objective function values Vs iterations) are separately presented in Fig. 2a–h. From these figures, it can be concluded that proposed aDE, aPSO and *h*aDEPSO converge much faster than other algorithms in all cases.

Also, an attempt is made to find global optimal solution total of 690 runs (30 runs for each UBFs with 30 population size) and illustrated in Fig. 3. It confers that the proposed algorithms score the highest optimum solutions.

Apart from this, computational time of proposed and compared algorithms on each UBF is computed and presented through box plots in Fig. 4. From this figure, it can be perceived that the proposed algorithms take lesser time to achieve the best value for the entire UBFs.

As a whole, above numerical, statistical and graphical result analysis shows that proposed aDE, aPSO and *h*aDEPSO are performed very competitive and/or equally with other compared algorithms. However, among three proposed algorithms *h*aDEPSO is superior.

Application

In order to further examine proposed algorithms aDE, aPSO and *h*aDEPSO are further applied to solve-following five well-known small- and one large-scale engineering design optimization problem.

Small-scale engineering design optimization problems

Considered small-scale engineering design optimization problems are briefed as follows.

(i) Welded beam design (WBD) problem

Its objective is to find the minimum cost design of a structural welded beam design, subject to constraints g_1: shear stress (τ), g_2: bending stress in the beam (σ), g_3, g_4, g_5: side constraints, g_6: end deflection of the beam (δ) and g_7: bucking load on the bar (P_c) with four design variables $h(x_1)$: thickness of the weld, $l(x_2)$: length of the welded joint, $t(x_3)$: width of the beam and $b(x_4)$: thickness of the beam. The schematic diagram of this problem is presented in Fig. 5, and it can be formulated as follows.

$$\text{minimize} \quad f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14.0 + x_2)$$

subject to:

$$g_1(x) = \tau(x) - \tau_{max} \le 0;\ g_2(x) = \sigma(x) - \sigma_{max} \le 0;$$
$$g_3(x) = x_1 - x_4 \le 0;\ g_4(x) = 0.10471x_1^2 + 0.04811x_3x_4(14 + x_2) - 5 \le 0;$$
$$g_5(x) = 0.125 - x_1 \le 0;\ g_6(x) = \delta(x) - \delta_{max} \le 0;$$
$$g_7(x) = P - P_c(x) \le 0 \text{ and } 0.1 \le x_i \le 2;\ i = 1,4\ \&\ 0.1 \le x_i \le 10;\ i = 2,3.$$

where $\tau(x) = \sqrt{(\dot{\tau})^2 + 2\dot{\tau}\ddot{\tau}\frac{x_2}{2R} + (\ddot{\tau})^2}$, $\dot{\tau} = \frac{p}{\sqrt{2}x_1x_2}$, $\ddot{\tau} = \frac{MR}{J}$, $M = P\left(L + \frac{x_2}{2}\right)$, $R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1+x_3}{2}\right)^2}$, $J = 2\left\{\sqrt{2}x_1x_2\left[\frac{x_2^2}{12} + \left(\frac{x_1+x_3}{2}\right)^2\right]\right\}$, $\sigma(x) = \frac{6PL}{x_4x_3^2}, \delta(x) = \frac{4PL^3}{Ex_3^3x_4}, P_c(x) = \frac{4.013E\sqrt{(x_3^2x_4^6/36)}}{L^2} \times \left(1 - \frac{x_3}{2L}\right)\sqrt{\frac{E}{4G}}$.

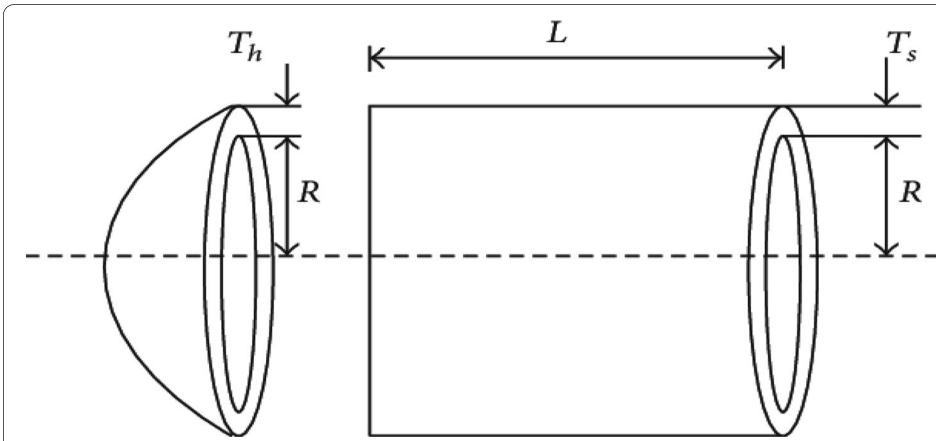


Fig. 7 Schematic diagram of pressure vessel design (PVD) problem

$P = 61lb, L = 14\text{in}, \ \mathrm{E} = 30 \times 10^6\text{psi}, G = 12 \times 10^6\text{psi}, \ \tau_{max} = 13,600\text{psi}, \sigma_{max} = 30,600\text{psi}, \delta_{max} = 0.25\text{in}.$

(ii) Three-bar truss design (TRD) problem

It is dealt with the design of a three-bar truss structure in which the volume is to be minimized subject to stress constraints. The problem has two decision variables and three constraints. The schematic diagram of this problem is presented in Fig. 6 and it can be formulated as follows.

$$\text{minimize} \quad f(x) = \left(2\sqrt{2}x_l + x_2\right) \times l$$

subject to:

$$g_1(x) = \frac{\left(\sqrt{2}x_1 + x_2\right)}{\sqrt{2}x_1^2 + 2x_1x_2}p - \sigma \le 0, \ g_2(x) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}p - \sigma \le 0$$
$$g_3(x) = \frac{1}{\sqrt{2}x_1 + x_1}p - \sigma \le 0;$$

where $0 \le x_i \le 1, i = 1, 2; l = 100\,\text{cm}, P = 2\,\text{kN/cm}^2, \ \sigma = 2\,\text{kN/cm}^2$.

(iii) Pressure vessel design (PVD) problem

Its objective is to minimize the total cost $f(x)$, including cost of the material, forming and welding with variables Ts (thickness of the shell), T_h (thickness of the head), R (inner radius) and L (length of the cylindrical section of the vessel). Both thickness variables (Ts, Th) must be integer multiple values of 0.0625 inch, which is the available thickness of rolled steel plates. R and L are continuous variables. A cylindrical vessel is capped at both ends by hemispherical heads. The schematic diagram of this problem is presented in Fig. 7 and it can be formulated as follows.

$$\text{minimize} \quad f(x) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3$$

subject to:

$$g_1(x) = -x_1 + 0.0193x_3 \le 0, \ g_2(x) = -x_2 + 0.00954x_3 \le 0,$$
$$g_3(x) = -\pi x_3^2x_4 - (4/3)\pi x_3^3 + 1296000 \le 0,$$
$$g_4(x) = x_4 - 240 \le 0$$

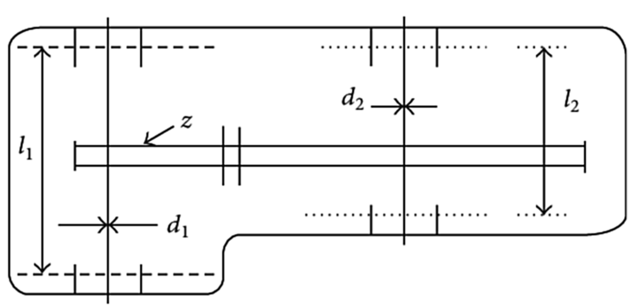


Fig. 8 Schematic diagram of speed reducer design (SRD) problem

where $0 \leq x_i \leq 100$; $i = 1, 2$ and $10 \leq x_i \leq 200$; $i = 3, 4$.

(iv) Speed reducer design (SRD) problem

Its aim is to minimize the weights of the speed reducer subject to constraints on bending stress of the gear teeth, surface stress, transverse deflections of the shafts and stresses in the shafts. The variables x_1 to x_7 represent the face width (b), module of teeth (m), number of teeth in the pinion (z), length of the first shaft between bearings (l_1), length of the second shaft between bearings (l_2) and the diameter of first (d_1) and second shafts (d_2), respectively. This is an example of a mixed integer programming problem. The third variable x_3 (number of teeth) is of integer values while all rest variables are continuous type. The schematic diagram of this problem is presented in Fig. 8 and it can be formulated as follows.

$$\begin{aligned}
&\text{minimize} \quad f(x) = 0.7854x_1x_2^2\left(3.3333x_3^2 + 14.9334x_3 - 43.0934\right) \\
&\quad - 1.508x_1\left(x_6^2 + x_7^2\right) + 7.4777\left(x_6^3 + x_7^3\right) \\
&\quad + 0.7854\left(x_4x_6^2 + x_5x_7^2\right)
\end{aligned}$$

subject to:

$$\begin{aligned}
&g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0,\ g_2(x) = \frac{397}{x_1x_2^2x_3} - 1 \leq 0, \\
&g_3(x) = \frac{1.93x_4^2}{x_1x_6^4x_3} - 1 \leq 0,\ g_4(x) = \frac{1.93x_4^2}{x_1x_7^4x_3} - 1 \leq 0, \\
&g_5(x) = \frac{\left[(745(x_4/x_2x_3))^2 + 16.9 \times 10^6\right]^{1/2}}{110x_6^3} - 1 \leq 0, \\
&g_6(x) = \frac{\left[(745(x_5/x_2x_3))^2 + 157.9 \times 10^6\right]^{1/2}}{85x_7^3} - 1 \leq 0, \\
&g_7(x) = \frac{x_2x_3}{40} - 1 \leq 0,\ g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0, \\
&g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0,\ g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0, \\
&g_{11}(x) = \frac{1.5x_7 + 1.9}{x_5} - 1 \leq 0,
\end{aligned}$$

where $2.6 \leq x_1 \leq 3.6, 0.7 \leq x_2 \leq 0.8, 17 \leq x_3 \leq 28, 7.3 \leq x_4 \leq 8.3, 7.3 \leq x_5 8.3,$ $2.9 \leq x_6 \leq 3.9, 5.0 \leq x_7 \leq 5.5.$

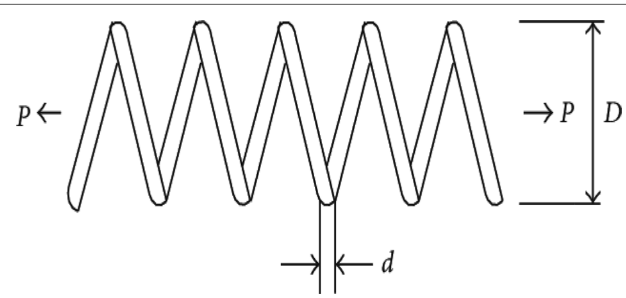


Fig. 9 Schematic diagram of tension/compression spring design (T/CSD) problem

Table 5 Optimum results for welded beam design problem

Algorithms	Optimal values for variables				Optimal cost
	$H(x_1)$	$L(x_2)$	$T(x_3)$	$b(x_4)$	
SCA	0.2444	6.2380	8.2886	0.2446	2.3854
SBM	0.2407	6.4851	8.2399	0.2497	2.4426
FSA	0.2444	6.1258	8.2939	0.2444	2.3811
DA	0.194288	3.46681	9.04543	0.205695	1.70808
EO	0.2057	3.4705	9.03664	0.2057	1.7249
CS	0.2015	3.562	9.0414	0.2057	1.73121
CDE	0.2031	3.5430	9.03350	0.2062	1.7335
IPSO	0.2444	6.2175	8.2915	0.2444	2.3810
CPSO	0.2024	3.5442	9.04821	0.2057	1.7280
ABC	0.205730	3.470489	9.036624	0.205730	1.724852
EPO	0.205411	3.472341	9.035215	0.201153	1.723589
GWO	0.205678	3.475403	9.036964	0.206229	1.726995
PSO	0.197411	3.315061	10.00000	0.201395	1.820395
MVO	0.205611	3.472103	9.040931	0.205709	1.725472
SHO	0.205563	3.474846	9.035799	0.205811	1.725661
GA	0.164171	4.032541	10.00000	0.223647	1.87397
aDE	0.184288	3.26641	8.24133	0.204585	1.70541
aPSO	0.184288	3.26641	8.24133	0.204585	1.70845
*h*aDEPSO	0.184288	3.26641	8.24133	0.204585	**1.69782**

Table 6 Comparative results for welded beam design problem

Algorithms	Best	Worst	Mean	std
SCA	2.3854	6.3996	3.2551	0.9590
SBM	2.4426	2.6315	2.5215	0.022184
FSA	2.3811	2.4889	2.4041	0.032194
DA	1.70808	1.94076	2.52106	0.250234
EO	1.724853	1.736725	1.726482	0.003257
CS	1.7312065	1.8786560	2.3455793	0.2677989
CDE	1.7335	1.824105	1.768158	0.022194
IPSO	2.3810	2.3110	2.3819	0.00523
CPSO	1.7280	1.782143	1.748831	0.012926
ABC	1.724852	1.734852	1.741913	0.031
EPO	1.723589	1.727211	1.725124	0.004325
GWO	1.726995	1.727128	1.727564	0.001157
PSO	1.820395	3.048231	2.230310	0.324525
MVO	1.725472	1.741651	1.729680	0.004866
SHO	1.725661	1.726064	1.725828	0.000287
GA	1.873971	2.320125	2.119240	0.034820
aDE	1.70541	1.72874	1.72625	0.000284
aPSO	1.70845	1.72698	1.71542	0.000354
*h*aDEPSO	**1.69782**	**1.72584**	**1.72421**	**0.000132**

Table 7 Optimum results for three-bar truss design problem

Algorithms	Optimal values for variables		Optimal cost
	$A_1(x_1)$	$A_2(x_2)$	
PSO	7.803e−01	4.330e−01	2.640e+02
DE	7.887e−01	4.080e−01	2.639e+02
CS	7.357e−01	5.945e−01	2.639e+02
CSDE	7.886e−01	4.082e−01	2.639e+02
KH	7.885e−01	4.088e−01	2.619e+02
PSO-OPS	7.886e−01	4.082e−01	2.620e+02
MBA	7.886e−01	4.086e−01	2.649e+02
SAC	0.7886210370	0.4084013340	263.8958466
DSS-MDE	0.7886751359	0.4082482868	263.8958434
GSA-GA	0.788676171219	0.408245358456	263.8958433
aDE	0.788526	0.408452	261.2654
aPSO	0.788526	0.408452	262.8536
*h*aDEPSO	0.788526	0.408452	**261.1438**

Table 8 Comparative results for three-bar truss design problem

Algorithms	Best	Worst	Mean	std
PSO	264.543754826635	2.70864524844245e+05	264.775374387455	1.85577512704186e+00
DE	263.148352624688	65,535	263.411272728312	1.05476883610644e−01
CS	263.602007628033	5.23525611223402e+12	263.671445662651	1.63380787023616e−02
CSDE	263.148352124271	65,535	263.148352318831	1.44060040776154e−08
KH	2.639e+02	2.650e+02	2.639e+02	1.658e−01
MBA	2.639e+02	2.639e+02	2.639e+02	3.930e−03
PSO-OPS	2.639e+02	2.639e+02	2.639e+02	1.354e−03
SACA	263.8958466	263.96975	263.9033	1.26e−02
DSS-MDE	263.8958434	263.8958498	263.8958436	9.72e−07
GSA-GA	263.8958433	263.8958459	263.8958437	5.34e−07
aDE	261.2654	262.7845	262.2541	1.0358e−08
aPSO	262.8536	263.5497	263.2154	1.2458e−08
*h*aDEPSO	**261.1438**	**262.9796**	**262.5782**	**2.5142e−09**

(v) Tension/compression spring design (T/CSD) problem

It minimizes the weight of the tension/compression spring, subject to constraints on the minimum deflection, shear stress, surge frequency, limits on outside diameter and on design variables. The design variables are wire diameter $d(x_1)$, mean coil diameter $D(x_2)$ and number of active coils $P(x_3)$. The schematic diagram of this problem is presented in Fig. 9, and it can be formulated as follows.

$$\text{minimize} \quad f(x) = (x_3 + 2)x_2x_1^2$$

Table 9 Optimum results for pressure vessel design problem

Algorithms	Optimal values for variables				Optimal cost
	T_s (x_1)	$T_h(x_2)$	$R(x_3)$	$L(x_4)$	
PSO	0.8125	0.4375	42.091266	176.7465	6061.0777
DE	0.8125	0.4375	42.098411	176.637690	6059.7340
ACO	0.8125	0.4375	42.103624	176.572656	6059.0888
ABC	0.812500	0.437500	42.098446	176.636596	6059.714339
GWO	0.8125	0.4345	42.089181	176.75731	6051.5639
DA	0.782825	0.384649	40.3196	200	5923.11
GA	0.752362	0.399540	40.452514	198.00268	5890.3279
HS	1.099523	0.906579	44.456397	179.65887	6550.0230
CS	0.812500	0.437500	42.0984456	176.6363595	6059.7143348
EO	0.7781	0.3846	40.319619	199.99999	5885.3279
CDE	0.8125	0.4375	42.098411	176.63769	6059.7340
CSDE	0.8125	0.4375	42.10	176.6	6060.0000
CPSO	0.8125	0.4375	42.091266	176.746500	6061.0777
IPSO	0.812500	0.437500	42.098445	176.6365950	6059.7143
CSKH	0.7781686	0.3846491	40.3196187	200.0000	6059.7010
aDE	0.8125	0.4375	40.32962	179.85887	5885.3279
aPSO	0.8125	0.4375	40.31962	179.85887	5885.3079
*h*aDEPSO	0.8125	0.4375	40.31962	179.65887	**5882.4387**

Table 10 Comparative results for pressure vessel design problem

Algorithms	Best	Worst	Mean	std
PSO	5891.3879	6531.5032	7394.5879	534.119
DE	6074.6231	6751.5312	6619.0083	358.799
ACO	6014.6231	6651.5312	6219.0083	423.524
ABC	6059.714339	6650.5102	6245.308144	205.000
GWO	5889.3689	5894.6238	5891.5247	013.910
DA	5923.11	222,536	21,342.2	470.44
GA	5890.3279	7005.7500	6264.0053	496.128
HS	6550.0230	8005.4397	6643.9870	657.523
CS	6059.714	6495.3470	6447.7360	502.693
EO	6059.7143	6668.114	7544.4925	566.24
CDE	6059.7340	6085.2303	6371.0455	43.0130
CSDE	6059.7133	1.528E+22	6261.4178	263.6758
CPSO	6061.0777	6147.1332	6363.8041	86.4545
IPSO	6059.7143	6251.5312	6289.92881	305.78
CSKH	5885.332773	5885.486467	5885.382053	0.049080
aDE	5883.3279	5884.7850	5883.7344	0.013023
aPSO	5885.3079	5885.8661	5885.4343	0.910214
*h*aDEPSO	**5882.4387**	**5882.8342**	**5882.6875**	**0.011289**

Table 11 Optimum results for speed reducer design problem

Algorithms	Optimal values for variables							Optimal cost
	b	m	p	l_1	l_2	d_1	d_2	
EPO	3.50123	0.7	17	7.3	7.8	3.33421	5.26536	2994.2472
SHO	3.50159	0.7	17	7.3	7.8	3.35127	5.28874	2998.5507
PSO	3.500019	0.7	17	8.3	7.8	3.352412	5.286715	3005.763
MVO	3.508502	0.7	17	7.3	7.8	3.358073	5.286777	3002.928
SCA	3.508755	0.7	17	7.3	7.8	3.461020	5.289213	3030.563
GSA	3.600000	0.7	17	8.3	7.8	3.369658	5.289224	3051.120
GA	3.510253	0.7	17	8.35	7.8	3.362201	5.287723	3067.561
HS	3.520124	0.7	17	8.37	7.8	3.366970	5.288719	3029.002
GWO	3.506690	0.7	17	7.380933	7.815726	3.357847	5.286768	3001.288
AFA	3.500	0.700	0.700e+00	7.3027	7.8007	3.350	5.287	3027.354
MBA [32]	3.500	0.700	0.700e+00	7.3007	7.716	3.350	5.287	3077.426
CS	3.501	0.700	0.700e+00	7.6057	7.818	3.352	5.287	3052.128
KH	3.500	0.700	0.700e+00	7.3667	7.823	3.350	5.287	3029.543
aDE	3.500	0.7	17	7.3809	7.8	3.350	5.289	2992.1242
aPSO	3.500	0.7	17	7.3809	7.8	3.350	5.289	2994.2442
*h*aDEPSO	3.500	0.7	17	7.3809	7.8	3.350	5.289	**2990.3582**

Table 12 Comparative results for speed reducer design problem

Algorithms	Best	Worst	Mean	std
EPO	2994.2472	2999.092	2997.482	1.78091
SHO	2998.5507	3003.889	2999.640	1.93193
PSO	3005.763	3211.174	3105.252	79.6381
HS	3029.002	3619.465	3295.329	57.0235
GA	3067.561	3313.199	3186.523	17.1186
GSA	3051.120	3363.873	3170.334	92.5726
SCA	3030.563	3104.779	3065.917	18.0742
MVO	3002.928	3060.958	3028.841	13.0186
GWO	3001.288	3008.752	3005.845	5.83794
AFA	2.996e+03	2.997e+03	2.996e+03	**9.000e−02**
MBA	2.994e+03	3.000e+03	2.997e+03	1.560e+00
CS	3.001e+03	3.009e+03	3.007e+03	4.968e+00
KH	2.997e+03	3.011e+03	3.006e+03	2.638e+00
aDE	2991.1242	2992.1114	2991.8756	0.01991
aPSO	2994.2442	2995.8574	2995.2347	0.08791
*h*aDEPSO	**2990.3582**	**2993.2145**	**2992.5481**	0.01693

subject to:

$$g_1(x) = 1 - \left(x_2^3 x_3/71785 x_1^4\right) \le 0,$$

$$g_2(x) = \left(4x_2^2 - x_1x_2/12566\left(x_2x_1^3 - x_1^4\right)\right) + \left(1/5108x_1^2\right) - 1 \le 0,$$

$$g_3(x) = \left(1 - \left(40.45x_1/x_2^2x_3\right)\right) \le 0,\ g_4(x) = ((x_2 + x_1)/1.5) - 1) \le 0$$

where $0.05 \le x_l \le 2.00, 0.25 \le x_2 \le 1.30, 2.00 \le x_3 \le 15.00$.

Table 13 Optimum results for tension/compression spring design problem

Algorithms	Optimal values for variables			Optimal cost
	$d(x_1)$	$D(x_2)$	$N(x_3)$	
GSA	0.050276	0.323680	13.525410	0.012702
CPSO	0.051728	0.357644	11.244543	0.012674
CDE	0.051609	0.354714	11.410831	0.012670
SCA	0.052160	368,159	10.648442	0.012669
EO	0.0516199100	0.355054381	11.387967	0.012666
PSOSCANMS	0.05072	0.334801	10.79431	0.012475
PSO	0.051728	0.357644	11.244543	0.0126747
HS	0.051154	0.349871	12.076432	0.0126706
DE	0.051609	0.354714	11.410831	0.0126702
AFA	5.167E−02	3.562E−01	1.132E+01	0.012624
GWO	0.05169	0.356737	11.28885	0.012666
WCA	0.0517208702	0.3579276279	11.1912042488	0.012630231
Modified DE	0.051688	0.356692	11.290483	0.012665
QPSO	0.051515	0.352529	11.538862	0.012665
SHO	0.051144	0.343751	12.0955	0.012674000
HS	0.05025	0.316351	15.23960	0.012776352
EPO	0.051087	0.342908	12.0898	0.012656987
aDE	0.05012	0.328431	11.49631	0.012552
aPSO	0.05012	0.328431	11.49631	0.012568
*h*aDEPSO	0.05012	0.328431	11.49631	**0.012475**

Table 14 Comparative results for tension/compression spring design problem

Algorithms	Best	Worst	Mean	std
GSA	0.012873881	0.014211731	0.013438871	0.000287
SCA	0.012669	0.016717	0.012922	5.92e−04
EO	0.012666	0.013997	0.013017	3.91e−04
CPSO	0.012674	0.012924	0.012730	5.19e−05
DE	0.012693	0.020034	0.012744	3.680e−05
PSO	0.016508	0.015234	0.049161	4.027e−04
AFA	0.012670	0.012710	0.012680	1.281e−05
GWO	0.012666	0.012654	0.012741	1.281e−05
WCA	0.012630231	0.017722009	0.013388089	1.0864e−03
QPSO	0.012665	0.017759	0.013524	1.268e−03
Modified DE	0.012665	0.012654	0.012666	2.0e−06
SHO	0.012674000	0.012715185	0.012684106	0.000027
HS	0.012776352	0.015214230	0.013069872	0.000375
EPO	0.012656987	0.012667902	0.012678903	0.001021
aDE	0.012552	0.012755	0.012654	2.02e−06
aPSO	0.012568	0.012689	0.012645	2.12e−06
*h*aDEPSO	**0.012475**	**0.012652**	**0.012564**	**1.12e−06**

The results of the proposed hybrid *h*aDEPSO and its suggested component algorithms aDE and aPSO algorithm on five small-scale engineering design optimization problems are compared with PSO [2], ABC [3], GWO [4], CS [6], DA [7], KH [8], DE [9], GA [10], GSA [11], EO [12], HS [13], WCA [14], MBA [16], CDE [19], modified DE [20],

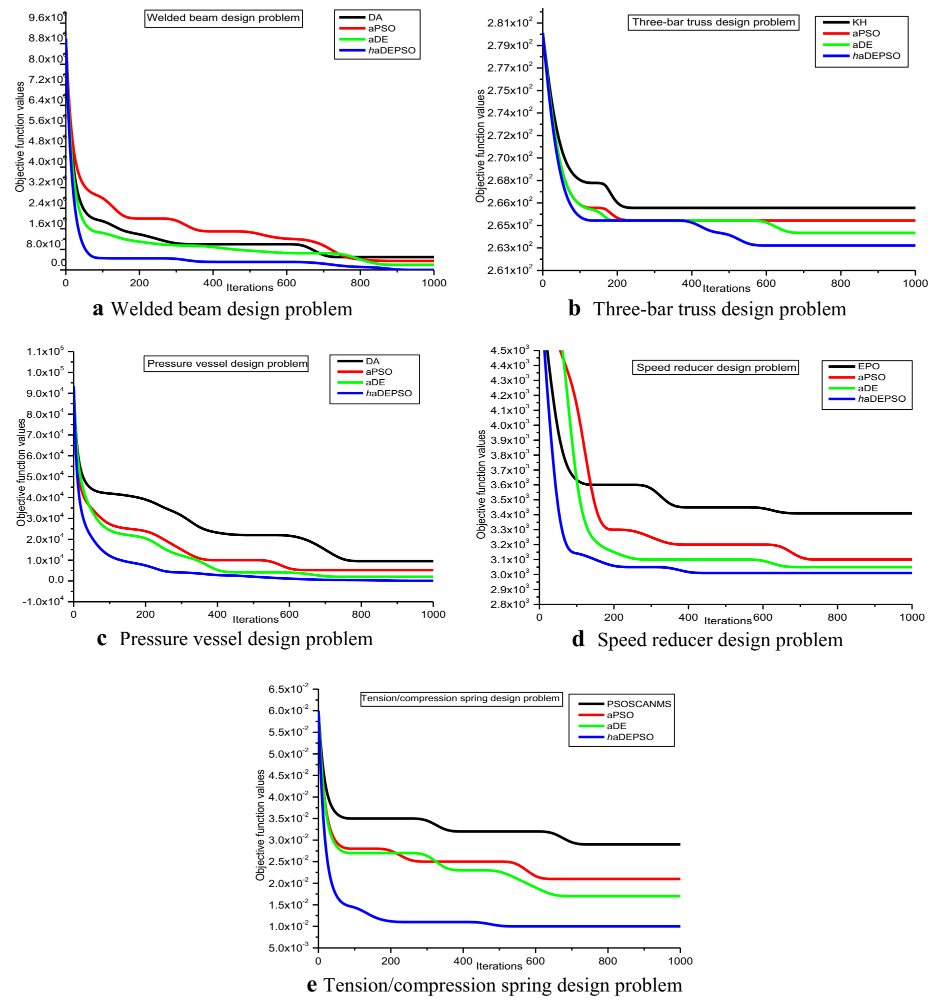


Fig. 10 Convergence graph for small-scale engineering design optimization problems

DSS-MDE [21], CPSO [24], IPSO [25], QPSO [26], PSO-OPS [27], CSDE [36], PSOS-CANMS [37], ACO [43], CSKH [44], SCA [45], SBM [46], FSA [47], EPO [48], MVO [49], SHO [50], AFA [51], SAC [52] and GSA-GA [53]. For fair comparison population size (30), stopping criteria (1500 iterations) and independent run (25) of proposed algorithms are taken same as comparative algorithms. The results of the comparative algorithms are taken from the original references, rest parameter of proposed algorithms as same as above. The optimal and comparative results of proposed algorithms with others on respective small-scale engineering design optimization problems are presented in Tables 5 and 6 (for WBD), Tables 7 and 8 (for TRD), Tables 9 and 10 (for PVD), Tables 11 and 12 (for SRD) and Tables 13 and 14 (for T/CSD).

As delineated in these tables, the produced optimal cost by proposed algorithms are summarized as follows for all five problems: (i) proposed aDE for WBD, TRD, PVD, SRD and T/CSD are 1.70541, 261.2654, 5885.3279, 2992.1242 and 0.012552, respectively, (ii) proposed aPSO for WBD, TRD, PVD, SRD and T/CSD are 1.70845, 262.8536, 5885.3079, 2994.2442 and 0.012568, respectively, and (iii) proposed haDEPSO for WBD, TRD, PVD, SRD and T/CSD are 1.69782, 261.1438, 5882.4387, 2990.3582 and 0.012475, respectively. Further, it can be concluded that the proposed algorithm outperformed and achieves the result with better best, worst, mean than other comparative algorithms. Moreover, securing less std. produced by proposed aDE, aPSO and *h*aDEPSO in all five problems describe their stability. Therefore, the proposed algorithm shows superior and competitive performance to other algorithms in all considered small-scale engineering problems.

The convergence graphs of all proposed and best non-proposed algorithm (to avoid complicacy) is plotted for all small-scale engineering design optimization problems and presented in Fig. 10a–e. From these figures, it can be clearly visualized that proposed algorithms converge faster than others. Hence, proposed algorithms are computationally efficient.

In general, from the all above result analysis it can be declared that proposed aDE, aPSO and *h*aDEPSO are performing better and/or equally with others. However, among three proposed algorithms *h*aDEPSO has larger competence.

Large-scale engineering design optimization problem: Economic load dispatch (ELD) problem with or without valve-point effects

Objective function of ELD problem with succeeding constraints can be represented as follows.

$$\text{minimize} \quad F = \sum_{i=1}^{n} F_i(P_i) = \sum_{i=1}^{n} a_i P_i^2 + b_i P_i + c_i + \alpha \left| e_i \sin\left(f_i \left(P_i^{min} - P_i \right) \right) \right|$$

where F: total fuel cost, n: number of generating unit, $F_i(P_i)$: operating fuel cost (real power output) and $a_i, b_i \& c_i$: cost coefficient of generating unit i. And P_i^{min}: minimum generation limit of unit i.

Constraints.

- Generator constraint: $P_i^{\min} \leq P_i \leq P_i^{\max}$, where $P_i^{\min}$ and $P_i^{\max}$: minimum and maximum power generation by unit i.
- Power balance constraint: $\sum_{i=1}^{n} P_i = D + P_L$, with $P_L = \sum_{i=1}^{n} \sum_{j=1}^{n} P_i B_{ij} P_j + \sum_{i=1}^{n} P_i B_{oi} + B_{oo}$, where D: total load demand, P_L: total transmission line loss and B_{ij},B_{oi},B_{oo}: transmission loss coefficient.
- Prohibited operating zone constraint: $P_i^{\min} \leq P_i \leq P_{i,1}^{l}$: $P_{i,k-1}^{u} \leq P_i \leq P_{i,k}^{l}$: $P_{i,n_i}^{u} \leq P_i \leq P_i^{\max}$; $k = 2, 3, \ldots n_i$ where n_i: number of prohibited operating zone and $P_{i,k}^{l}$ and $P_{i,k}^{u}$: lower and upper limit of kth prohibited zone of generating unit i.
- Ramp rate limit constraint: $\max\left(P_i^{\min}, P_i^{t-1} - \mathrm{DR}_i\right) \leq P_i^t \leq \min\left(P_i^{\max}, P_i^{t-1} + \mathrm{UR}_i\right)$, where P_i^t and P_i^{t-1} current and previous output power and UR_i and DR_i: up and down ramp limit of generating unit i.

In the next section, ELD problem is solved with and without valve-point loading effects using below considered 3-, 6-, 15-, 40- and 140-unit test system (TSys) and the results are compared state-of-the-art algorithms.

Table 15 Simulation results for TSys-1 (3-unit system)

Power output (MW)	Compared algorithms						Proposed algorithms		
	PSO	DE	GA	MTVPSO	THS	MGSO	aDE	aPSO	*h*aDEPSO
1	300.27	300.27	300.00	300.27	300	300.27	300.27	300.27	300.26
2	400.00	400.00	400.00	400.00	400	400.00	400.00	400.00	400.00
3	149.73	149.73	150.00	149.73	149.7331	149.74	149.73	149.73	149.74
Total power(MW)	850.00	850.00	850	850.00	850	850	850.00	850.00	850.00
Min cost ($/h)	**8234.07173**	**8234.07173**	**8234.60**	**8234.07173**	**8234.07**	**8234.07**	**8234.0719**	**8234.0721**	**8234.0717**
Mean cost ($/h)	8235.979526	8234.823309	8236.75	**8234.07173**	8234.55	8237.85	8236.7525	8235.8475	8235.3598
Max cost ($/h)	8241.587522	8241.587522	8239.99	**8234.07173**	8236.87	8240.54	8240.3584	8241.2583	8240.1522
Std. of cost ($/h)	3.2186	2.29328	3.58	1.8501e−12	1.8020	5.26	1.5874e−13	2.1251e−13	**1.2351e−14**
CPU mean time (s)	4.37	2.06	2.27	1.29	1.48	2.85	1.27	1.33	**1.22**

Table 16 Simulation results for TSys-2 (6-unit system)

Power output (MW)	Compared algorithms										Proposed algorithms		
	PSO	GA	MTVPSO	IPSO	EHM	BCO	NCS	MPSO-TVAC	IPSO-TVAC	θ-PSO	aDE	aPSO	*h*aDEPSO
1	446.986	474.8066	451.5204	449.802	449.1546	444.9513	446.71	448.170	447.5840	447.3555	451.2214	451.5384	451.5204
2	170.196	178.6363	172.1750	171.042	173.0613	173.8016	173.01	173.291	173.2010	173.2577	172.452	172.825	172.175
3	252.902	262.2089	258.4186	250.865	266.0092	263.3943	265.00	263.145	263.3310	263.3848	258.5174	259.3497	258.4186
4	150.000	134.2826	140.6441	150.000	127.1203	138.6992	139.00	138.714	138.8520	139.0440	140.6441	140.5341	140.6441
5	178.780	151.9039	162.0797	159.347	174.2603	167.9755	165.23	165.960	165.3280	165.3317	162.0797	161.0893	162.0797
6	77.085	74.1812	90.3415	94.633	85.8777	87.1664	86.78	86.691	87.1500	87.0593	90.3415	90.1115	90.3415
Total power (MW)	1275.95	1276.03	1275.1795	1275.69	1275.4834	1275.9882	1275.7	1275.97	1275.4460	1275.433	1275.256	1275.448	1275.179
Power loss (MW)	12.95	13.0217	12.1795	12.69	12.4834	12.9864	12.733	12.97	12.4460	12.4429	12.256	12.448	12.179
Power balance (MW)	0.000	0.0083	0.000	0.000	0.0	0.0018	0.0048	0.000	0.000	−0.0099	0.000	0.000	0.000
Min cost ($/h)	15,450.00	15,459.00	15,441.1084	15,453.50	15,441.5974	15,450.031	15,447.00	15,449.91	15,443.063	15,442.9411	15,441.3561	15,441.8451	**15,440.1288**
Mean cost ($/h)	15,454.00	15,461.35	15,441.1087	15,462.59	15,442.8547	15,452.257	15,448.58	15,450.17	15,443.582	15,442.9419	15,442.0215	15,443.2215	**15,441.6548**
Max cost ($/h)	15,492.00	15,485.87	15,441.1104	15,468.48	15,446.5874	15,455.458	15,449.85	15,451.57	155,445.114	15,442.9500	15,446.7845	15,447.1847	**15,444.8541**
Std. of cost ($/h)	0.0025	0.0078	0.0031	0.84	0.00046	0.0069	0.00145	0.37	0.00255	0.0015	0.0027	0.0031	**0.0022**
CPU mean time (s)	14.89	41.58	**0.32**	1.25	**0.32**	3.10	7.58	1.68	0.89	5.4429	0.38	0.41	0.34

Table 17 Simulation results for TSys-3 (15-unit system)

Power output (MW)	Compared algorithms										Proposed algorithms		
	PSO	GA	MTVPSO	EHM	BCO	NCS	MPSO-TVAC	θ-PSO	DEPSO	DPD	aDE	aPSO	*h*aDEPSO
1	455.00	415.3108	454.9812	455	452.9151	455.00	455.00	455	455	454.9999	454.8834	454.8823	454.9813
2	380.00	359.7206	455	380	358.8547	380.00	380.00	380	420	454.9999	455.0000	455.0000	455.0000
3	130.00	104.4250	130	130	127.6452	130.00	130.00	130	130	130.0000	130.0000	130.0000	130.0000
4	130.00	74.9853	130	130	128.4156	130.00	130.00	130	130	130.0000	130.0000	130.0000	130.0000
5	154.42	380.2844	235.5844	170	276.0158	170.00	170.00	170	270	234.2005	239.1899	240.9334	239.4334
6	460.00	426.7902	460	460	429.9371	460.00	459.99	460	460	460.0000	460.0000	460.0000	460.0000
7	430.00	341.3164	465	430	437.8152	430.00	430.00	430	430	464.9999	462.0000	460.0000	460.0000
8	60.00	124.7867	60	90.14947	62.8458	60.00	72.60	75.0139	60	60.00000	61.0000	60.0000	60.0000
9	74.27	133.1445	25	37.75777	59.5343	71.05	58.32	55.8293	25	25.00000	24.0000	26.0000	25.0000
10	160.00	89.2567	28.9867	160	96.7215	159.85	159.73	160	62	30.99387	28.6810	28.9997	28.8757
11	80.00	60.0572	76.8357	80	75.2118	80.00	80.00	80	80	76.70138	75.9970	76.1987	75.1865
12	79.60	49.9998	80	80	78.4522	80.00	80.00	80	80	79.99999	80.0000	80.0000	80.0000
13	25.00	38.7713	25	25	35.5118	25	25.01	25.0012	25	25.00000	26.0000	25.0000	25.0000
14	27.59	41.9425	15	15	19.1549	15	15.00	15.0000	15	15.00000	15.0000	15.0000	15.0000
15	15.00	22.6445	15	15	21.0143	15	15.00	15.0181	15	15.00000	15.0000	15.0000	15.0000
Total power (MW)	2660.88	2630.0	2656.3882	2657.90724	2660.0483	2660.9	2660.66	2660.8625	2657.96	2656.89544	2656.7513	2657.0141	2653.4769
Power loss (MW)	30.88	38.2782	26.3882	27.90724	29.4073	30.908	30.66	30.8699	27.976	26.89544	26.7513	27.0141	23.4769
Power balance (MW)	**0.00000**	0.1218	**0.00000**	**0.0**	0.641	**0.0**	**0.00000**	−0.0074	0.010	**0.00000**	**0.00000**	**0.00000**	**0.00000**
Min cost ($/h)	32,731.96	33,113	32,542.7320	32,672.9595	32,714.265	32,708	32,704.47	32,706.6048	32,588.81	32,548.5857	32,542.7820	32,542.4512	**32,542.1452**
Mean cost ($/h)	33,039.00	33,254	32,550.9885	32,699.2548	32,725.251	32,854	32,705.00	32,709.3196	32,591.49	32,556.6793	32,550.9055	32,550.5421	**32,550.7457**
Max cost ($/h)	33,331.00	33,285	32,562.4861	32,708.8547	32,755.369	32,975	32,728.99	32,739.4865	32,588.99	32,564.4051	32,562.4814	32,559.2411	**32,558.2584**
Std. of cost ($/h)	2.8475	1.598	2.04619	3.5841	2.4584	6.2541	3.51	7.3140	4.02	2.095632	1.85214	1.97548	**1.04350**
CPU mean time (s)	26.59	49.31	0.190	0.368	12.42	12.79	12.78	11.7380	1.96	1.98594	0.175	0.172	**0.181**

Table 18 Simulation results for TSys-4 (40-unit system)

Power output (MW)	Compared algorithms							Proposed algorithms		
	MTVPSO	THS	DEPSO	DPD	MPSO	IABC	DHS	aDE	aPSO	*h*aDEPSO
1	110.7998	114	110.802	111.7629	111.3021	110.8067	110.7998	110.8069	110.7998	110.7993
2	110.7998	113.3808	110.801	111.6926	110.8937	110.8163	110.7998	110.7937	110.8001	111.1416
3	97.3999	97.4102	97.400	97.40940	97.4024	97.4000	97.3999	97.3998	97.401	97.4122
4	179.7331	179.7357	179.733	179.7721	179.7417	179.7330	179.7331	179.7332	179.7327	179.7443
5	87.7999	96.9973	87.800	88.30690	96.2717	87.8133	87.7999	87.9566	87.8019	88.1519
6	140.0000	140.0000	140.000	139.9833	140.0000	139.9999	140.0000	140.0000	139.9977	139.9957
7	259.5997	259.6047	259.600	259.7218	259.5998	259.5996	259.5997	259.5996	259.5986	259.6064
8	284.5997	284.6041	284.600	284.7273	284.6047	284.6008	284.5997	284.5996	284.6058	284.6046
9	284.5997	284.6018	284.600	284.6157	284.6048	284.5997	284.5997	284.5996	284.6004	284.6149
10	130.0000	130	130.000	130.0583	130.0020	130.0000	130.0000	130.0000	130.0000	130.0000
11	94.0000	168.8034	94.000	168.7990	168.7993	168.7998	94.0000	168.7998	94.0007	168.8029
12	94.0000	214.7619	94.000	168.7894	94.0019	94.0000	94.0000	94.0000	94.0000	94.0000
13	214.7598	394.2794	214.760	214.7593	214.7600	125.0000	214.7598	214.7597	214.7579	214.7593
14	394.2794	304.5215	394.279	304.5391	394.2799	400.0000	394.2794	394.2795	394.2796	394.2717
15	394.2794	304.5209	394.279	394.2707	394.2789	394.2791	394.2794	304.5197	394.2784	304.5205
16	394.2794	489.2841	394.279	394.2713	304.5202	394.2793	394.2794	394.2793	394.2799	394.2835
17	489.2794	489.2891	489.279	489.2894	489.2798	489.2796	489.2794	489.2796	489.2787	489.2911
18	489.2794	511.2813	489.279	489.3177	489.2823	489.2794	489.2794	489.2792	489.279	489.2878
19	511.2794	511.2790	511.279	511.2724	511.2796	511.2792	511.2794	511.2794	511.2785	511.2976
20	511.2794	523.2838	511.279	511.2800	511.2823	511.2793	511.2794	511.2794	511.2793	511.2792
21	523.2794	523.2819	523.279	523.3291	523.2799	523.2798	523.2794	523.2794	523.2787	523.2957
22	523.2794	523.2779	523.279	523.2992	523.2794	523.2793	523.2794	523.2794	523.2793	523.2848
23	523.2794	523.2801	523.279	523.3545	523.2815	523.2793	523.2794	523.2794	523.2798	523.2857
24	523.2794	523.2824	523.279	523.2793	523.2800	523.2793	523.2794	523.2794	523.2793	523.2979
25	523.2794	523.2799	523.279	523.3890	523.2799	523.2793	523.2794	523.2794	523.2783	523.2799

Table 18 (continued)

Power output (MW)	Compared algorithms							Proposed algorithms		
	MTVPSO	THS	DEPSO	DPD	MPSO	IABC	DHS	aDE	aPSO	*h*aDEPSO
26	523.2794	523.2799	523.279	523.2776	523.2812	523.2793	523.2794	523.2794	523.2787	523.291
27	10.0000	10	10.000	10.00000	10.0010	10.0000	10.0000	10.0000	10.0002	10.0064
28	10.0000	10	10.000	10.00000	10.0002	10.0000	10.0000	10.0000	10.0010	10.0018
29	10.0000	10	10.000	10.00000	10.0021	10.0000	10.0000	10.0000	10.0011	10.0000
30	87.7999	96.99	87.800	88.72800	88.0447	91.4006	87.7999	97.0000	87.8000	96.2132
31	190.0000	190	190.000	189.9811	189.9997	189.9999	190.0000	190.0000	189.9996	189.9997
32	190.0000	190	190.000	189.9982	189.9996	189.9999	190.0000	190.0000	189.9979	189.9997
33	190.0000	190	190.000	189.9795	189.9999	190.0000	190.0000	190.0000	189.9981	189.9981
34	164.7998	164.8838	164.800	164.8962	164.8005	164.7997	164.7998	164.7997	199.8696	164.9126
35	200.0000	200	194.395	165.0916	199.9950	199.9978	200.0000	200.0000	194.5334	199.9941
36	194.3978	200	200.000	165.2998	199.9953	199.9999	194.3978	200.0000	164.8003	200.0000
37	110.0000	110	110.000	109.9827	109.9987	110.0000	110.0000	110.0000	109.9992	109.9988
38	110.0000	110	110.000	109.9824	109.9997	109.9998	110.0000	110.0000	109.9984	109.9994
39	110.0000	110	110.000	109.9634	109.9990	110.0000	110.0000	110.0000	109.9974	109.9974
40	511.2794	511.2795	511.279	511.5279	511.2957	511.2793	511.2794	511.2793	511.2797	511.2800
Total power (MW)	10,500	10,500	10,500	10,500	10,500	10,500	10,500	10,500	10,500	10,500
Min cost ($/h)	**121,403.5355**	121,467.44	121,412.56	121,410.5355	121,379.43	121,491.2751	**121,403.5355**	121,405.7384	121,404.5378	121,403.5454
Mean cost ($/h)	121,410.5967	121,524.26	121,419.31	121,412.5729	121,384.43	121,539.4175	121,410.5967	121,411.5829	121,410.5977	**121,410.4658**
Max cost ($/h)	121,417.2274	121,598.65	121,468.25	121,441.9027	121,391.07	121,582.3865	121,417.2274	121,415.2314	121,416.3319	**121,413.9827**
Std. of cost ($/h)	0.001714	36.7026	0.00584	0.437608	2.3685	0.00548	4.80	0.0037608	0.001754	**0.001473**
CPU mean time (s)	11.32101	13.5846	7.895	21.76485	5.43	1.950	1.32	11.12101	12.78485	**11.02356**

Table 19 Simulation results for TSys-5 (140-unit system)

Power output (MW)	Compared algorithms			Proposed algorithms				Compared algorithms			Proposed algorithms		
	MTVPSO	MPSO	DHS	aDE	aPSO	*h*aDEPSO		MTVPSO	MPSO	DHS	aDE	aPSO	*h*aDEPSO
1	116.2518	116.2514	119.0000	116.2528	116.2418	116.2528	71	143.3803	143.3803	500.0000	143.4785	143.8778	143.3804
2	188.8395	188.8395	119.0000	188.8345	188.8295	188.8375	72	388.3236	388.3236	500.0000	388.3785	388.1248	388.7854
3	189.9734	189.9734	164.0000	189.9724	189.9834	189.9734	73	202.9482	202.9482	241.0000	202.7785	202.7854	202.1245
4	189.8984	189.8984	164.0000	189.8974	189.8984	189.8974	74	176.0783	176.0783	241.0000	176.0785	176.0745	176.7854
5	168.6455	168.6455	190.0000	168.6445	168.6655	168.6455	75	175.8523	175.8523	241.0000	175.8785	175.8545	175.4561
6	187.3990	187.3990	190.0000	187.394	187.389	187.391	76	177.8040	177.8040	241.0000	177.8874	177.8045	177.8125
7	489.9976	489.9976	190.0000	489.9946	489.9976	489.9977	77	179.5410	179.5410	774.0000	179.5254	179.5411	179.4856
8	489.8645	489.8645	190.0000	489.8645	489.8745	489.8645	78	330.6797	330.6797	774.0000	330.6258	330.6745	330.6785
9	495.9917	495.9917	168.5398	495.9997	495.9817	495.9918	79	530.9939	530.9939	769.0000	530.9358	530.9488	530.9987
10	495.9000	495.9000	168.5398	495.904	495.91	495.7854	80	530.9852	530.9852	769.0000	530.9154	530.9878	530.9258
11	495.9999	495.9999	190.0000	495.9949	495.9299	495.7896	81	260.0529	260.0529	3.0000	260.0535	260.0787	260.7854
12	495.9418	495.9418	190.0000	495.9408	495.9318	495.9408	82	56.04720	56.0472	82 3.0000	56.04715	56.04748	56.07854
13	505.9503	505.9503	490.0000	505.9513	505.9803	505.9513	83	115.2868	115.2868	83 3.0000	115.2154	115.2487	115.2987
14	508.9842	508.9842	490.0000	508.9832	508.9242	508.9822	84	115.0050	115.0050	84 3.0000	115.0785	115.1452	115.1545
15	505.9854	505.9854	490.0000	505.9844	505.9454	505.9814	85	115.3836	115.3836	250.0000	115.3834	115.3987	115.3886
16	504.9535	504.9535	490.0000	504.9525	504.9435	504.9635	86	207.0000	207.0000	250.0000	207.0788	207.0078	207.0000
17	505.9841	505.9841	496.0000	505.9881	505.9941	505.9741	87	207.0965	207.0965	250.0000	207.0985	207.0965	207.0785
18	505.9365	505.9365	496.0000	505.9365	505.9265	505.9465	88	175.1027	175.1027	250.0000	175.1078	175.1125	175.1485
19	504.9788	504.9788	496.0000	504.9748	504.9688	504.9888	89	175.1656	175.1656	250.0000	175.1678	175.1145	175.1678
20	504.9618	504.9618	496.0000	504.9628	504.9718	504.9418	90	182.1361	182.1361	250.0000	182.1354	182.1361	182.1378
21	504.9769	504.9769	496.0000	504.9779	504.9869	504.9369	91	175.1669	175.1669	250.0000	175.1678	175.1785	175.1754
22	504.9081	504.9081	496.0000	504.9091	504.9281	504.9181	92	579.9855	579.9855	250.0000	579.9854	579.9452	579.9785
23	504.9514	504.9514	496.0000	504.9524	504.9614	504.9814	93	645.0000	645.0000	250.0000	645.0999	645.0089	645.0754
24	504.9742	504.9742	496.0000	504.9742	504.9442	504.9642	94	983.9400	983.9400	250.0000	983.9485	983.9200	983.9487
25	536.9944	536.9944	506.0000	536.9934	536.9844	536.9844	95	977.9925	977.9925	250.000	977.9978	977.9785	977.9925

Table 19 (continued)

Power output (MW)	Compared algorithms			Proposed algorithms				Compared algorithms			Proposed algorithms		
	MTVPSO	MPSO	DHS	aDE	aPSO	*h*aDEPSO		MTVPSO	MPSO	DHS	aDE	aPSO	*h*aDEPSO
26	536.9804	536.9804	506.0000	536.9814	536.9704	536.9704	96	681.9838	681.9838	250.0000	681.9825	681.9025	681.9895
27	548.9866	548.9866	509.0000	548.9867	548.9666	548.9466	97	719.9953	719.9953	250.0000	719.9985	719.9954	719.9975
28	548.9698	548.9698	509.0000	548.9699	548.9498	548.9398	98	718.0000	718.0000	250.0000	718.0897	718.0854	718.0045
29	500.9609	500.9609	506.0000	500.9604	500.9309	500.9509	99	719.9553	719.9553	250.0000	719.9558	719.9785	719.9553
30	500.8584	500.8584	506.0000	500.8585	500.8984	500.8484	100	963.8138	963.8138	250.0000	963.8145	963.825	963.8125
31	505.9982	505.9982	505.0000	505.9987	505.9482	505.8982	101	957.9252	957.9252	165.0000	957.9285	957.9152	957.9245
32	505.9890	505.9890	505.0000	505.9899	505.999	505.889	102	1006.9391	1006.9391	165.0000	1006.9381	1006.9321	1006.9311
33	505.9872	505.9872	506.0000	505.9472	505.9472	505.8872	103	1005.9547	1005.9547	165.0000	1010.5031	1005.9547	1005.9547
34	505.8398	505.8398	506.0000	505.8394	505.8398	505.9398	104	1012.9951	1012.9951	165.0000	1012.9981	1012.9971	1012.9781
35	499.9759	499.9759	506.0000	499.9554	499.9759	499.8759	105	1019.9532	1019.9532	165.0000	1019.9512	1019.9512	1019.9572
36	499.8309	499.8309	506.0000	499.7854	499.8801	499.8325	106	953.9784	953.9784	165.0000	953.9785	953.9784	953.9778
37	240.9712	240.9712	505.0000	240.8454	240.9718	240.9745	107	951.9893	951.9893	165.0000	951.9847	951.9893	951.9898
38	240.9668	240.9668	505.0000	280.9596	240.9668	240.9154	108	1005.9502	1005.9502	165.0000	1005.957	1005.9502	1005.9522
39	773.9998	773.9998	505.0000	773.9154	773.9198	773.9896	109	1012.9827	1012.9827	180.0000	1012.9847	1012.9827	1012.9828
40	768.9143	768.9143	505.0000	768.9154	767.9123	768.9145	110	1020.9864	1020.9864	180.0000	1020.9874	1020.9864	1020.9887
41	3.484200	3.4842	505.0000	3.484295	3.984201	3.48422	111	1014.9943	1014.9943	180.0000	1014.9973	1014.9943	1014.9987
42	3.027400	3.0274	505.0000	3.027444	3.087478	3.027401	112	94.03550	94.0355	180.0000	94.78569	94.0355	94.035485
43	186.7539	186.7539	505.0000	186. 9636	186.5847	186.7525	113	94.03730	94.0373	103.0000	132.0374	94.0373	94.037378
44	216.9286	216.9286	505.0000	216.9255	216.4896	216.9287	114	94.01730	94.0173	103.0000	94.25732	94.0173	94.017325
45	249.3928	249.3928	505.0000	298.3924	249.2541	249.3928	115	244.3547	244.3547	198.0000	244.3547	244.3547	244.35474
46	249.9292	249.9292	505.0000	249.9299	249.4856	249.9293	116	244.0590	244.0590	198.0000	244.159	244.059	244.05902
47	246.5461	246.5461	505.0000	246.5445	246.1596	246.5458	117	244.1444	244.1444	312.0000	289.1844	244.1444	241.7985
48	248.6987	248.6987	505.0000	248.7854	248.6562	248.6785	118	95.32550	95.3255	312.0000	95.3215	95.3255	95.32458
49	248.7121	248.7121	537.0000	248.7854	248.7486	248.7485	119	95.01860	95.0186	308.5679	95.0166	95.0186	95.01848
50	248.8088	248.8088	537.0000	248.2541	248.7854	248.8052	120	116.1636	116.1636	308.5881	116.1536	116.1636	116.1645
51	166.6126	166.6126	537.0000	166.1548	166.8526	166.6115	121	175.0352	175.0352	163.0000	175.0152	175.0352	175.0351

Table 19 (continued)

Power output (MW)	Compared algorithms			Proposed algorithms				Compared algorithms			Proposed algorithms		
	MTVPSO	MPSO	DHS	aDE	aPSO	*h*aDEPSO		MTVPSO	MPSO	DHS	aDE	aPSO	*h*aDEPSO
52	165.0625	165.0625	537.0000	165.8651	165.7425	165.0156	122	2.00640	2.0064	163.0000	2.00646	2.0064	2.00645
53	169.5018	169.5018	549.0000	169.8745	169.8918	169.5785	123	4.04020	4.0402	95.0000	4.84021	4.0402	4.04022
54	166.3101	166.3101	549.0000	166.9658	166.3101	166.3125	124	15.0993	15.0993	95.0000	15.8992	15.0993	15.0993
55	180.0956	180.0956	549.0000	180.0956	180.7856	180.0154	125	9.00550	9.0055	511.0000	9.9755	9.0055	9.00555
56	180.2748	180.2748	549.0000	180.2254	180.7848	180.2715	126	12.0655	12.0655	511.0000	12.0555	12.0655	12.0654
57	103.1771	103.1771	501.0000	103.1741	103.1521	103.1715	127	10.0080	10.0080	511.0000	10.018	10.008	10.0084
58	198.1757	198.1757	501.0000	198.1351	198.7847	198.1745	128	112.0382	112.0382	511.0000	111.4382	112.0382	112.038
59	309.9475	309.9475	499.0000	309.9874	309.9985	309.9479	129	4.08120	4.0812	490.0000	4.7852	4.0812	4.08123
60	293.8267	293.8267	499.0000	293.8895	293.8157	293.8214	130	5.02750	5.0275	490.0000	5.9255	5.0275	5.0275
61	163.5888	163.5888	506.0000	163.5147	163.5898	163.5145	131	5.05170	5.0517	256.7772	5.0557	5.0517	5.05145
62	95.30030	95.3003	506.0000	95.30454	95.3014	95.30152	132	50.1071	50.1071	256.7802	50.9171	50.1071	50.1458
63	168.1584	168.1584	506.0000	168.1785	168.1244	168.1598	133	5.04680	5.0468	490.0000	5.0468	5.0468	5.78542
64	166.0715	166.0715	506.0000	166.0701	166.0725	166.0025	134	42.0244	42.0244	490.0000	42.9244	42.0244	42.0785
65	486.8818	486.8818	506.0000	486.8458	486.8808	486.8085	135	42.0341	42.0341	490.0000	42.9331	42.0341	42.2548
66	205.5539	205.5539	506.0000	205.7889	205.5529	205.5025	136	41.0343	41.0343	490.0000	41.9343	41.0343	41.7854
67	483.1274	483.1274	506.0000	483.1984	483.1324	483.1215	137	17.0443	17.0443	130.0000	17.9453	16.1243	17.7869
68	486.0292	486.0292	506.0000	486.1442	486.1522	486.0785	138	7.14970	7.1497	130.0000	7.1498	7.1497	7.14948
69	130.2962	130.2962	500.0000	130.2542	130.2252	130.2902	139	7.02920	7.0292	339.4395	7.0291	7.0292	7.02921
70	339.3610	339.3610	500.0000	339.3055	339.314	339.3678	140	26.4594	26.4594	339.4395	26.4894	26.4594	26.4684
							Total power (MW)	49,342.000	49,342	49,342	49,342.000	49,342.000	49,342.000
							Min cost ($/h)	1,560,436.71	1,560,436.0	1,657,944.8622	1,560,436.76	1,560,435.88	**1,560,434.54**
							Mean cost ($/h)	1,560,446.22	1,560,445.0	1,657,944.8627	1,560,445.43	1,560,444.56	**1,560,440.32**
							Max cost ($/h)	1,560,460.55	1,560,462.0	1,657,944.8652	1,560,461.66	1,560,461.52	**1,560,460.89**
							Std. of cost ($/h)	0.00003	0.000458	0.0005	0.0000054	0.0000076	**0.0000003**
							CPU mean time (s)	17.03	18.43	4.6	5.66	5.98	**3.98**

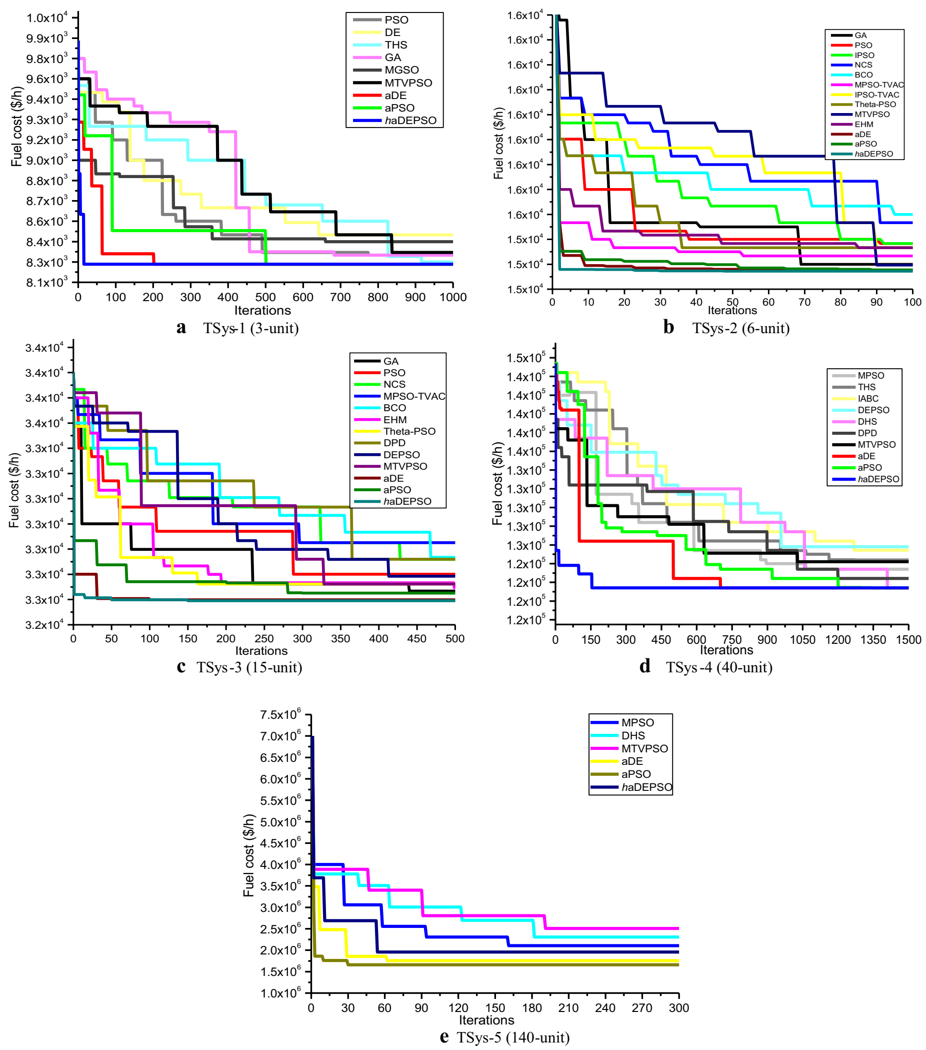


Fig. 11 Cost convergence characteristic for different test systems of ELD problem

Test systems	Description
TSys-1 (3-unit test system) [54]	It involves valve-point effects with 850 MW total demand
TSys-2 (6-unit test system) [55]	It consists of transmission losses, ramp-rate limit and prohibited operating zone constraints with 1263 MW total demand
TSys-3 (15-unit test system) [55]	It comprises ramp-rate limits and prohibited operating zone constraints with 2630 MW total demand
TSys-4 (40-unit test system) [54]	It implicates valve-point effects with 10,500 MW total demand
TSys-5 (140-unit test system) [56]	It consists of ramp-rate limits, valve-point loading effects and prohibited operating zone constraints with 49,342 MW total demand

The results produced by proposed algorithms on above considered different test systems of ELD problem are compared with other state-of-the-art algorithms. These compared algorithms are listed as follows: PSO [2], DE [9], GA [10], MTVPSO [28], IPSO [29], MPSO-TVAC [30], IPSO-TVAC [31], θ-PSO [32], MPSO [33], DEPSO [38], DPD [39], THS [57], MGSO [58], EHM [59], BCO [60], NCS [61], IABC [62] and

DHS [63]. In order to check the efficiency of the proposed algorithms aDE, aPSO and *h*aDEPSO, least values of population size (30), maximum number of iterations (1000) and independent runs (30) have been considered among compared algorithms.

The comparative simulation results of proposed and compared algorithms for TSys-1, TSys-2, TSys-3, TSys-4 and TSys-5 are reported in Tables 15, 16, 17, 18 and 19, respectively, over 30 runs.

As reported in these tables, the global optimal cost produced by- (i) proposed aDE for TSys-1, TSys-2, TSys-3, TSys-4 and TSys-5 are 8234.0719 ($/h), 15,441.3561 ($/h), 32,542.7820 ($/h), 121,405.7384 ($/h) and 1,560,436.76 ($/h), respectively, (ii) proposed aPSO for TSys-1, TSys-2, TSys-3, TSys-4 and TSys-5 are 8234.0721 ($/h), 15,441.8451 ($/h), 32,542.4512 ($/h), 121,404.5378 ($/h) and 1,560,435.88 ($/h), respectively, and (iii) proposed *h*aDEPSO for TSys-1, TSys-2, TSys-3, TSys-4 and TSys-5 are 8234.0717 ($/h), 15,440.1288 ($/h), 32,542.1452 ($/h), 121,403.5454 ($/h) and 1,560,434.54 ($/h), respectively. These reported cost results for all test systems show that the proposed algorithms succeed in finding the best solution in comparison with other algorithms. Furthermore, the mean and maximum fuel cost together with standard deviation and CPU mean time for each test systems are also recorded in the same tables. Also, it can be seen from these tables that proposed algorithms surpassed all other comparative algorithms by providing the best result with regard to minimum, mean and maximum cost. It is noteworthy that the proposed algorithms can still yield better solutions with low standard deviations and acceptable CPU time. It signifies that the proposed algorithm has stronger convergence with higher stability and reliability/robustness compared to other existing algorithms.

The convergence curves of proposed with compared algorithms are plotted in Fig. 11a–e for TSys-1, TSys-2, TSys-3, TSys-4 and TSys-5 of ELD problem in terms of fuel cost and iterations. This figure shows that proposed algorithms aDE, aPSO and *h*aDEPSO have more robust convergence where the results improved as the iterations increased.

Conclusion

In this study, an advanced hybrid algorithm *h*aDEPSO proposed for solving small- and large-scale engineering design optimization problems, where an advanced DE (aDE) and PSO (aPSO) are integrating in suggested hybrid. The brief summary of these proposed algorithms is given as follows.

(i) An advanced hybrid algorithm (*h*aDEPSO) has been established by combining aDE and aPSO. It is based on multi-swarm approach where the population of one is merged with other in a pre-defined manner which yields guaranteed convergence and diversifying the solutions.

(ii) To enhance performance and easily adjust the control parameters of DE, an advanced DE (aDE) is developed. The novel mutation strategy, crossover probability and altered selection schemes of aDE will guarantee high and low population diversity at start and end of the algorithm, respectively.

(iii) To avoid particles stagnant, an advanced PSO (aPSO) is proposed which consists of novel gradually varying (decreasing and/or increasing) parameters. These control parameters can well-balance the exploration and exploitation capabilities and promotes the particles to search high-quality solution of aPSO.

The effectiveness of the proposed hybrid *h*aDEPSO and its suggested component algorithms aDE and aPSO algorithm are tested on 23 unconstrained benchmark functions, then applied on five well-known small engineering design optimization problems, namely welded beam design (WBD), three-bar truss design (TRD), pressure vessel design (PVD), speed reducer design (SRD) and tension/compression spring design (T/CSD) problem and one large-scale engineering design optimization problem, viz., economic load dispatch (ELD) having five different test systems (3-, 6-, 15-, 40-, 140-unit). The numerical, statistical and graphical analyses of the proposed algorithms are compared against the state-of-the-art algorithms. The comparative results shape that the proposed algorithms become more robust and effective to solve complex engineering design optimization problems. Thus, it is conclusive that the proposed algorithms can be treated as a vital alterative in the field of MAs. Moreover, in the view of feasibilities, superiorities and solution optimality, among all and suggested algorithms *h*aDEPSO outperformed.

In addition, the proposed algorithms have higher time complexity compared to some DE, PSO and hybrid variants. The main reason of time-consuming of proposed algorithms is the matrix operation execution. This operation is repeated per individuals in each iteration and increases the running time of the algorithm to some extent. Moreover, the proposed algorithms may not suitable for all engineering design optimization problems as others. As a part of our future work, some novel parameters will be designed for the proposed aDE, aPSO and *h*aDEPSO in the hope of finding more accurate solutions and reduce the time complexity. Additionally, effectiveness of the proposed algorithms can be tested by some more complicated real-world applications and new MAs will be developed in future. Finally, this paper is expected more attention to the analysis of how to strengthen the robustness of the proposed algorithms for complex optimization problems.

Abbreviations

MAs: Meta-heuristics algorithms; SIAs: Swarm intelligence algorithms; EAs: Evolutionary algorithms; PBAs: Physics-based algorithms; HBAs : Human behavior-based algorithms; PSO: Particle swarm optimization; ABC: Artificial bee colony; GWO: Grey wolf optimizer; HHO: Harris hawks optimization; CS: Cuckoo search; DA: Dragonfly algorithm; f : Real-valued function; D : Dimension; l_j and u_j : Lower and upper limits for *j*th decision vector limits; L and K : Total number of inequality and equality constraint; g_l : Inequality constraint; h_k : Equality constraint; x_i : Position vector of *i*th particle; v_i : Velocity vector of *i*th particle; $pbest_i$: Individual best position of *i*th particle; KH: Krill Herd; DE: Differential evolution; GA: Genetic algorithm; GSA: Gravitational search algorithm; EO: Equilibrium optimizer; HS: Harmony search; WCA: Water cycle algorithm; TLBO: Teaching–learning-based optimization; MBA: Mine blast algorithm; UBFs: Unconstrained benchmark functions; WBD: Welded beam design; $gbest_j$: Global best position of particle; t : Iteration index; c_1 and c_2 : Cognitive and social acceleration coefficient; r_1 and r_2 : Uniform random numbers in [0, 1]; w : Inertia weight; np : Population size; x_l and x_u : Lower and upper boundaries; x_{ij}^t : Target vector; v_{ij}^t : Mutant vector; SRD: Speed reducer design; TRD: Three-bar truss design; PVD: Pressure vessel design; T/CSD: Tension/compression spring design; ELD: Economic load dispatch; aDE: Advanced differential evolution; aPSO: Advanced particle swarm optimization; *h*aDEPSO: *Hybrid* Advanced DEPSO; std.: Standard deviation; *df*: Degree of freedom; WSR: Wilcoxon signed rank; u_{ij}^t : Trial vector; F : Scaling vector; C_r : Crossover rate; $i :\in [1, np]$; $j :\in [1, D]$; rand : Random numbers; t_{max} : Maximum iterations; τ : Convergence factor.

Acknowledgements

Not applicable.

Authors' contributions

We hereby declare that both authors contributed to the design and implementation of the research, whereas PV (Pooja Verma) conducted literature review, interpretation of the data and provide the resources for the paper and RPP (Raghav Prasad Parouha) contributed to the analysis of the results and to the writing of the manuscript as well as implementation of the simulation model in the C language environment. The manuscript has been read and approved by all named authors, and there are no other persons who satisfied the criteria for authorship but are not listed. We further confirm that the order of authors listed in the manuscript has been approved by all of us. All authors read and approved the final manuscript.

Funding
This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Availability of data and materials
We confirm that all data generated or analyzed during this study are included in the submitted manuscript. All authors confirm that all relevant data are included in the article in the "Application" section and "Conclusion" section from Table 1 to Table 19. Also, availability of data and materials are cited in references of the manuscript. Moreover, the data that support the findings of this study are available from the corresponding author upon reasonable request. No additional data archiving is necessary.

Declarations

Competing interests
The authors declared that they had no conflicts of interest with respect to their authorship or the publication of this article.

Received: 3 September 2020 Accepted: 10 March 2021
Published online: 29 March 2021

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. Kennedy J, Eberhart RC (1995) Particle swarm optimization. In: Proceeding of IEEE international conference on neural networks, pp 1942–1948
3. 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
4. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61
5. Heidari AA, Mirjalili S, Faris H, Aljarah I, Mafarja M, Chen H (2019) Harris hawks optimization: algorithm and applications. Futur Gener Comput Syst 97:849–872
6. 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
7. Mirjalili S (2016) Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete and multi-objective problems. Neural Comput Appl 27(4):1053–1073
8. Gandomi AH, Alavi AH (2012) Krill herd: a new bio-inspired optimization algorithm. Commun Nonlinear Sci Numer Simul 17(12):4831–4845
9. Storn R, Price K (1997) Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim 11:341–359
10. Davis L (1991) Handbook of genetic algorithms
11. Rashedi E, Nezamabadi-pour H, Saryazdi S (2009) A gravitational search algorithm. Inf Sci 179(13):2232–2248
12. Faramarzi A, Heidarinejad M, Stephens B, Mirjalili S (2019) Equilibrium optimizer: a novel optimization algorithm. Knowl-Based Syst 191:1–34
13. Geem ZW, Kim JH, Loganathan GV (2001) A new heuristic optimization algorithm: harmony search. SIMULATION 76(2):60–68
14. 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
15. Rao RV, Savsani VJ, Vakharia DP (2012) Teaching–learning—based optimization: an optimization method for continuous non-linear large scale problems. Inf Sci 183:1–15
16. Sadollah A, Bahreininejad A, Eskandar H, Hamdi M (2013) Mine blast algorithm: a new population based algorithm for solving constrained engineering optimization problems. Appl Soft Comput 13(5):2592–2612
17. Zhang J, Sanderson C (2009) JADE: Adaptive Differential Evolution with optional external archive. IEEE Trans Evol Comput 13(5):945–958
18. Tanabe R, Fukunaga A (2013) Success-history based parameter adaptation for Differential Evolution. In: IEEE Congress on Evolutionary Computation, pp 71–78
19. Huang F, Wang L, He Q (2007) An effective co-evolutionary differential evolution for constrained optimization. Appl Math Comput 186:340–356
20. Montes EM, Coello C, Reyes J, Muñoz-Dávila L (2007) Multiple trial vectors in differential evolution for engineering design. Eng Optim 39:567–589
21. Zhang M, Luo W, Wang X (2008) Differential evolution with dynamic stochastic selection for constrained optimization. Inf Sci 178(15):3043–3074
22. Mahmoodabadi MJ, Mottaghi ZS, Bagheri A (2014) High exploration particle swarm optimization. J Inf Sci 273:101–111
23. 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

24. He Q, Wang L (2007) An effective co-evolutionary particle swarm optimization for constrained engineering design prob . Eng Appl Artif Intell 20:89–99
25. He S, Prempain E, Wu QH (2004) An improved particle swarm optimizer for mechanical design optimization problems. Eng Optim 36:585–605
26. Coelho L (2010) Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems. Expert Syst Appl 37:1676–1683
27. Isiet M, Gadala M (2020) Sensitivity analysis of control parameters in particle swarm optimization. J Comput Sci 41:1–33
28. Parouha RP (2019) Nonconvex/nonsmooth economic load dispatch using modified time-varying particle swarm optimization. Comput Intell 35:717–744. https://doi.org/10.1111/coin.12210
29. Safari A, Shayegui H (2011) Iteration particle swarm optimization procedure for economic load dispatch with generator constraints. Expert System Appl 38(5):6043–6048
30. Abdullah MN, Bakar AHA, Rahim NA, Mokhlis H, Illias HA, Jamian JJ (2014) Modified particle swarm optimization with time varying acceleration coefficients for economic load dispatch with generator constraints. J Electr Eng Technol 9(1):15–26
31. Mohammadi BL, Rabiee A, Soroudi A, Ehsan M (2012) Iteration PSO with time varying acceleration coefficients for solving non-convex economic dispatch problems. Int J Electr Power Energy Syst 42(1):508–516
32. Hosseinnezhad V, Babaei E (2013) Economic load dispatch using θ-PSO. Int J Electr Power Energy Syst 49:160–169
33. Basu M (2015) Modified particle swarm optimization for nonconvex economic dispatch Problems. Electr Power Energy Syst 69:304–312
34. 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
35. 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
36. Zhang Z, Ding S, Jia W (2019) A hybrid optimization algorithm based on cuckoo search and differential evolution for solving constrained engineering problems. Eng Appl Artif Intell 85:254–268
37. Fakhouri HN, Hudaib A, Sleit A (2020) Hybrid particle swarm optimization with Sine Cosine Algorithm and Nelder-Mead Simplex for solving engineering design problems. Arab J Sci Eng 4:3091–3109
38. Sayah S, Hamouda A (2013) A hybrid differential evolution algorithm based on particle swarm optimization for nonconvex economic dispatch problems. Appl Soft Comput 13(4):1608–1619
39. Parouha RP, Das KN (2016) A novel hybrid optimizer for solving economic load dispatch problem. Int J Electr Power Energy Syst 78:108–126
40. Wolpert DH, Macready WG (1997) No free lunch theorems for optimization. IEEE Trans Evol Comput 1(1):67–82
41. Deb K (1995) Optimization for engineering design: algorithms and examples. Prentice-Hall of India, New Delhi
42. Das KN, Parouha RP (2015) An ideal tri-population approach for unconstrained optimization and applications. Appl Math Comput 256:666–701
43. Kaveh A, Talatahari S (2010) An improved ant colony optimization for constrained engineering design problems. Eng Comput 27:155–182
44. Basset M, Wang G, Sangaiah AK, Rushdy E (2019) Krill Herd algorithm based on cuckoo search for solving engineering optimization problems. Multimed Tools Appl 78:3861–3884
45. Mirjalili S (2016) SCA: A sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133
46. Akhtar S, Tai K, Ray T (2002) A socio-behavioural simulation model for engineering design optimization. Eng Optim 34:341–354
47. Hedar AR, Fukushima M (2006) Derivative-free filter simulated annealing method for constrained continuous global optimization. J Glob Optim 35:521–549
48. Dhiman G, Kumar V (2018) Emperor Penguin optimizer: a bio-inspired algorithm for engineering problems. Knowl-Based Syst 159:20–50
49. Mirjalili S, Mirjalili SM, Hatamlou A (2015) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513
50. Dhiman G, Kumar V (2017) Spotted hyena optimizer: a novel bio-inspired based metaheuristic technique for engineering applications. Adv Eng Softw 114:48–70
51. Baykasoglu A, Ozsoydan FB (2015) Adaptive firefly algorithm with chaos for mechanical design optimization problems. Appl Soft Comput 36:152–164
52. Ray T, Liew KM (2003) Society and civilization: an optimization algorithm based on the simulation of social behavior. IEEE Trans Evol Comput 7(4):386–396
53. Garg H (2019) A hybrid GSA-GA algorithm for constrained optimization problems. Inf Sci 478:499–523
54. Sinha N, Chakrabarti R, Chattopadhyay PK (2003) Evolutionary programming techniques for economic load dispatch. IEEE Trans Evol Comput 7(1):83–94
55. Gaing ZL (2003) Particle swarm optimization to solving the economic dispatch considering the generator constraints. IEEE Trans Power Syst 18(3):1187–1195
56. dos Santos Coelho L, Bora TC, Mariani VC (2014) Differential evolution based on truncated Lévy-type flights and population diversity measure to solve economic load dispatch problems. Int J Electr Power Energy Syst 57:178–188
57. Mohammed AA, Mohammed AA, Ahamad TK, Asaju LB (2016) Tournament-based harmony search algorithm for non-convexeconomic load dispatch problem. Appl Soft Comput 47:449–459
58. Zare K, Haque MT, Davoodi E (2012) Solving non-convex economic dispatch problem with valve point effects using modified group search optimizer method. Electr Power Syst Res 84(1):83–89
59. Kasmaei MP, Nejad MR (2011) An effortless hybrid method to solve economic load dispatch problem in power systems. Energy Convers Manag 52:2854–2860
60. Chokpanyasuwan C, Anantasate S, Pothiya S, Pattaraprakom W, Bhasaputra P (2009) Honey bee colony optimization to solve economic dispatch problem with generator constraints. IEEE ECTI-Conference

61. Kuo CC (2008) A novel coding scheme for practical economic dispatch by modified particle swarm approach. IEEE Trans Power Syst 23:1825–1835
62. Aydın D, Liao T, Montes M, Stützle T (2011) Improving performance via population growth and local search the case of the artificial bee colony algorithm. In: International conference on artificial evolution, pp 85–96
63. Wang L, Li L (2013) An effective differential harmony search algorithm for the solving non-convex economic load dispatch problems. Electr Power Energy Syst 44:832–843

Publisher's Note

Pooja Verma is currently research scholar at Department of Mathematics, Indira Gandhi National Tribal University, (A Central University Established by an Act of Parliament) Lalpur, Amarkantak, Dist.-Anuppur Madhya Pradesh - 484887, 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 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), genetic algorithm (GA) and nature inspired (NI) techniques such as particle swarm optimization techniques (PSO), artificial bee colony (ABC), etc., as well as hybrid algorithms.

Raghav Prasad Parouha, is currently Assistant Professor at Department of Mathematics, Indira Gandhi National Tribal University, (A Central University Established by an Act of Parliament) Lalpur, Amarkantak, Dist.-Anuppur Madhya Pradesh - 484887, India. He did Ph.D. from National Institute of Technology (NIT), Silchar, Assam, India, in August 2015 and M.Sc. from Govt. Model Science College (Auto.) 'A+', Jabalpur, (M.P.). His area of research interests is 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 07 workshops/seminars, etc.