frontiers | Frontiers in Energy Research

TYPE Original Research
PUBLISHED 19 January 2023
DOI 10.3389/fenrg.2022.1061408

OPEN ACCESS

EDITED BY
K. Sudhakar,
Universiti Malaysia Pahang, Malaysia

REVIEWED BY
Bishwajit Dey,
Gandhi Institute of Engineering and Technology University (GIET), India
Amit Kumar,
Thapar Institute of Engineering and Technology, India

*CORRESPONDENCE
Huiling Chen,
chenhuiling.jlu@gmail.com
Guoxi Liang,
guoxiliang2017@gmail.com

SPECIALTY SECTION
This article was submitted to Solar Energy, a section of the journal Frontiers in Energy Research

RECEIVED 04 October 2022
ACCEPTED 05 December 2022
PUBLISHED 19 January 2023

CITATION
Dong R, Ma L, Chen H, Heidari AA and Liang G (2023), Hybrid kernel search and particle swarm optimization with Cauchy perturbation for economic emission load dispatch with valve point effect.
Front. Energy Res. 10:1061408.
doi: 10.3389/fenrg.2022.1061408

Hybrid kernel search and particle swarm optimization with Cauchy perturbation for economic emission load dispatch with valve point effect

Ruyi Dong[1], Long Ma[1], Huiling Chen[2]*, Ali Asghar Heidari[3] and Guoxi Liang[4]*

[1]College of Information and Control Engineering, Jilin Institute of Chemical Technology, Jilin, China, [2]College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, China, [3]School of Surveying and Geospatial Engineering, College of Engineering, University of Tehran, Tehran, Iran, [4]Department of Information Technology, Wenzhou Polytechnic, Wenzhou, China

Due to growing concerns over environmental protection, economic and environmentally responsible power dispatching has become a hot topic in the field of power system control. Multi-objective optimization minimizes fuel costs and pollution emissions without violating operational constraints. To solve this problem, the MOP is decomposed into individual objects *via* the weighted sum method, and Newton's method is used to tackle equality constraints iteratively. To this end, a hybrid algorithm (HKSOPSO-CP) based on kernel search optimization (KSO) and particle swarm optimization (PSO) with Cauchy perturbation is proposed in this paper. An experiment with 23 CEC benchmark functions shows that HKSOPSO-CP offers better performance compared with various popular algorithms proposed in recent years. When employed to solve the IEEE standard economic emission dispatch (EED) problems with 6, 10, 40, and 110 units, the proposed HKSOPSO-CP algorithm produces results indicating a better trade-off between the objectives relating to fuel costs and emissions compared to other algorithms that have recently been reported on in the literature.

KEYWORDS

economic emission dispatch, particle swarm optimization, kernel search optimization, Cauchy perturbation, swarm intelligence

1 Introduction

The goal of economic dispatch (ED) is to schedule power generation to minimize the cost of producing electricity within the confines of operational constraints, such as the demand load and transmission loss. Decision makers in energy fields need effective hybrid models in order to reduce the cost of operations as much as possible (Ge et al., 2022a; Ge et al., 2022b; Chen, 2022). Even minor improvements in the operation of power plants can bring substantial economic benefits. Furthermore, thermal power plants inevitably emit a large amount of pollution during the operation of power systems, which has become a critical issue with increasing concerns over global warming. Therefore, the problem of simultaneous combined minimization of fuel costs and pollutant emissions, known as the economic emission dispatch (EED) problem, has become a research focus.

With people's increasing awareness of pollutant emissions, the economic emission dispatch problem (EED), which seeks to minimize fuel costs and pollution, has been a growing concern.

The EED problem represents a multi-objective optimization problem (MOP) (El-Keib and Ding, 1994). The aim in solving it is to find the best possible power distribution for each generator without violating the relevant constraints. In order to solve this type of problem, various techniques have been developed.

Optimization methods can be regarded as falling into various classes, including deterministic, stochastic, and evolutionary (Wang et al., 2022a). Traditional methods attempt to guarantee exact solutions (Zhang et al., 2022a), while in contrast, recent evolutionary methods cannot guarantee the identification of trustworthy optimal solutions, and it is regarded as sufficient to return solutions only satisfying local optima (Cao et al., 2022a). Recently, many studies have been conducted to solve the EED problem. There is a growing interest in many modern random-based metaheuristic algorithms (MAs), because they do not need any derivative information or gradient information in the search process (Jadhav et al., 2013); these methods include particle swarm optimization (PSO) (Cao et al., 2020; Lin et al., 2022), the colony predation algorithm (CPA) (Tu et al., 2021), hunger games search (HGS) (Yang et al., 2021), Harris hawks optimization (HHO) (Heidari et al., 2019), the slime mould algorithm (SMA) (Li et al., 2020a), the weighted mean of vectors (INFO) (Ahmadianfar et al., 2022), and the Runge Kutta optimizer (RUN) (Ahmadianfar et al., 2021), which have yielded great success in a variety of fields, such as feature selection (Hu et al., 2022; Liu et al., 2022), economic emission dispatch (Dong et al., 2021), bankruptcy prediction (Xu et al., 2019; Zhang et al., 2021), train scheduling (Song et al., 2023), image segmentation (Hussien et al., 2022; Yu et al., 2022), multi-objective problems (Hua et al., 2021; Deng et al., 2022a), gate resource allocation (Wu et al., 2020a; Deng et al., 2020), complex optimization problems (Deng et al., 2022b), resource allocation (Deng et al., 2022c), expensive optimization problems (Li et al., 2020b; Wu et al., 2021a), airport taxiway planning (Deng et al., 2022d), robust optimization (He et al., 2020), scheduling problems (Gao et al., 2020; Han et al., 2021; Wang et al., 2022b), and medical diagnosis (Wang et al., 2017) (Chen et al., 2016). Hota et al. (2010) used fuzzy computing to solve EED problems using a modified bacterial foraging algorithm (MBFA). Roy and Bhui used quasi-oppositional teaching learning based optimization (QOTLBO) (Roy and Bhui, 2013) to solve the problem of EED with valve point loading. de Athayde Costa e Silva et al. (2013) used an improved scatter search (ISS) to solve EED problems; experimental results showed that the ISS method achieves a better optimal result than other optimization algorithms.

Yang (Abdelaziz et al., 2016a) proposed the flower pollination algorithm (FPA). This approach can meet the needs of single-objective and multi-objective optimization problems, so it has strong universality in tackling various problems in the domain of power systems. Therefore, Abdelaziz et al. (2016b) used the FPA to solve the EED problem. Simulation results show that the FPA method achieves better results than other methods for various optimization problems, especially for large power systems. Kumar et al. (2012) proposed the Pareto bee colony optimization algorithm and applied it to the standard IEEE 30-bus system to tackle the EED problem. The experimental results were better than those achieved by other multi-objective optimization algorithms. Singh and Dhillon used an opposition-based greedy heuristic search (OGHS) (Singh and Dhillon, 2016) to tackle the power system scheduling problem. In this type of problem, an initial population is randomly generated by means of a uniform distribution, and the inverse solution is generated by means of oppositional learning. The optimal solution is retained for further optimization, which is beneficial to accelerate the convergence rate. Goudarzi et al. (2019) tackled the EED problem using a combination of a genetic algorithm (GA) and particle search optimization (PSO). Hosny et al. (2021) adopted an improved version of the slime mould algorithm (ISMA) to solve the EED problem, enhancing its search capabilities and balancing global and local search. The results showed that, compared with other optimization algorithms, the ISMA method achieves better optimization. Dong et al. proposed a kernel search optimization algorithm (KSO) (Dong and Wang, 2020; Dong et al., 2021) to solve the EED problem. By projecting the function from low-dimensional space to high-dimensional space, the non-linear problem can be transformed into a linear problem. In comparison with other algorithms, better optimal results are obtained.

As mentioned above, the identification of a better multi-objective optimization algorithm is always a research hotspot in the field of power system scheduling optimization. Such improvements take a great deal of effort, and even a small change represents a major improvement in environmental and economic terms. This article proposes an enhanced new metaheuristic for optimization, which uses kernel techniques from the KSO algorithm combined with a PSO algorithm, to tackle the EED problem. The main contributions of this paper can be summarized as follows:

- The local search ability of the KSO algorithm is improved by introducing sociality and self-recognition from the PSO algorithm. The update to the kernel vector in KSO is calculated based on the current velocity and position of the particle instead of *via* randomization.
- The Cauchy distribution approaches 0 more slowly than the normal distribution, so Cauchy mutation has the better disturbance ability. Cauchy mutation is utilized to update the target position to expand the distribution area of feasible solutions and improve the algorithm's global optimization performance.
- A hybrid KSO and PSO algorithm with Cauchy perturbation is utilized to solve the EED problem. Moreover, the weighted sum method transforms the multi-objective EED into a single-objective problem, and the Newton–Raphson method handles the power balance constraint.
- The HKSOPSO-CP algorithm is verified by solving EED problem cases involving four electrical systems. It is thereby demonstrated that the proposed algorithms can provide better solutions to EED problems than conventional KSO and other algorithms entered into this comparison.
- Under testing with 23 CEC benchmark functions, the HKSOPSO-CP algorithm outperforms KSO and several other well-known algorithms in accuracy.

The remainder of this article is organized as follows: Section 2 lays out the mathematical model of the EED problem; Section 3 presents the original KSO and PSO algorithms and the Cauchy mutation mechanism, as well as the new HKSOPSO-CP algorithm; Section 4 presents the experimental results and analysis; and Section 5 provides the conclusion.

2 The economic emission dispatch problem

In the EED problem, the objective is to minimize fuel costs while simultaneously minimizing pollutant emissions by identifying the generator schedule that represents the best compromise.

2.1 The objective function

The objective function of the fuel costs can be expressed as follows (Sierpina, 2013):

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

where C is the fuel cost in the problem to be solved; a_i, b_i, and c_i represent the cost coefficients for the i th unit; e_i and f_i represent the valve point effect coefficients; Pi represents the power output; and N indicates the number of generating units.

The total emission of pollutants can be defined as follows (Mavrovouniotis and Yang, 2011):

$$E = \sum_{i=1}^{N} \left[\alpha_i + \beta_i P_i + \gamma_i P_i^2 + \eta_i \exp \left(\delta_i P_i \right) \right] \tag{2}$$

where E is the pollution emission in the problem to be solved, and α_i, β_i, γ_i, η_i, and δ_i represent the emission coefficients of the i th generator.

There are two different objectives in this multi-objective optimization problem. The weighted sum method decomposes the MOP into individual objectives to solve this problem. The method combines conflicting objectives by introducing weight factors. The combined objective function takes the following form (Yaşar and Özyön, 2012):

$$F = wC + \gamma (1 - w) E \tag{3}$$

where w is the weight factor and γ is the scaling factor.

2.2 System constraints

1) Power balance constraints: the total power of all the generators must cover the sum of the load demand and the losses of the real power system (Zhan et al., 2014),

$$\sum_{i=1}^{N} P_i = P_D + P_L,\ P_L = \sum_{i=1}^{N} \sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{N} Pi + B_{00}, \tag{4}$$

where N is the loss coefficient, B_{ij} represents the number of generators, P_L represents the transmission losses, and P_D represents the system load.

2) Power capacity limitation: Each generator's output must fall within its relative output range (maximum-minimum),

$$P_i{}^{min} \le P_i \le P_i{}^{max} \tag{5}$$

where $P_i{}^{max}$ and $P_i{}^{min}$ are the operational limits for the i th generating unit.

The inequality constraint of Eq. 5 can be easily satisfied by setting the upper limit and lower limit in the initialization stage. But the equality constraint of Eq. 4 is usually difficult to satisfy. Therefore, Newton's method is used to tackle the constraints iteratively:

$$P_i = P_i{}^{\min} + rand[0,1] \times \left(P_i{}^{\max} - P_i{}^{\min} \right) i = 1, 2, \ldots\ldots N - 1 \tag{6}$$

where P_i is the output of the i th generating unit.

It is possible to solve Eq. 4 iteratively for the original output of the Nth generator:

$$P_N^{old} = p_D - \sum_{i=1}^{N-1} P_i \tag{7}$$

The transmission loss of the power system P_L^{old} can be expressed in the form of B-matrix coefficients (Zou et al., 2017):

$$P_L^{old} = \sum_{i=1}^{N} \sum_{j=1}^{N} P_i B_{ij} P_j + \sum_{i=1}^{n} B_{0i} P_i + B_{00} \tag{8}$$

where P_L^{old} is the original output of the Nth generator.

Subsequently, the new output of the Nth generator can be calculated as follows:

$$P_N^{new} = P_D - \sum_{i=1}^{N-1} P_i - P_L^{old} \tag{9}$$

The new output of the Nth generator is then substituted into Eq. 8 until the difference between them is less than some minimal value.

The Nth generator P_N, following the above steps, is able to fulfill the equality constraint of Eq. 4, but it is uncertain whether its output falls within the feasible domain or not. Therefore, a common penalty function is utilized here as an effective way to address this.

For Eq. 10, an efficient method employing a penalty function is applied to determine whether the output of the Nth generator that satisfies the constraint of Eq. 4 is within the feasible region.

$$\tilde{F} = F + \lambda \left[\max \left(P_N{}^{\min} - P_N, 0 \right) + \max \left(P_N - P_N{}^{\max}, 0 \right) \right] \tag{10}$$

where λ is a penalty factor.

3 The principle of the HKSOPSO-CP algorithm

The conventional KSO algorithm and the improved HKSOPSO-CP technique developed using the PSO algorithm and Cauchy perturbation are described in this section.

3.1 Overview of the KSO algorithm

KSO, proposed by Dong (Dong and Wang, 2020) in 2020, is a novel swarm intelligence algorithm whose search strategy is based on the kernel method. KSO was inspired by existing physical or biological MAs, but differs markedly from them: all the MAs mentioned above involve non-linear iterations to approach the optimal values of the target functions, but KSO involves linear iteration *via* the kernel method. The inspiration is as follows.

The non-linear objective function $y = f(x)$, $x = (x_1, x_2, \ldots x_n)$ can be converted into a linear one in higher-dimensional space by a mapping function $u = \phi(x)$. As the dimensionality of the space increases, the possibility of a non-linear function being transformed into a linear function becomes stronger. This is represented as:

$$y = f(x) = w^T \cdot u + b$$
$$w = (w_1, w_2, \ldots w_m)$$
$$u = (u_1, u_2, \ldots u_m)$$

where m is the dimensionality of the higher-dimensional space, and $w = (w_1, w_2, \ldots w_m)$ and $u = (u_1, u_2, \ldots u_m)$ are both m-dimensional vectors.

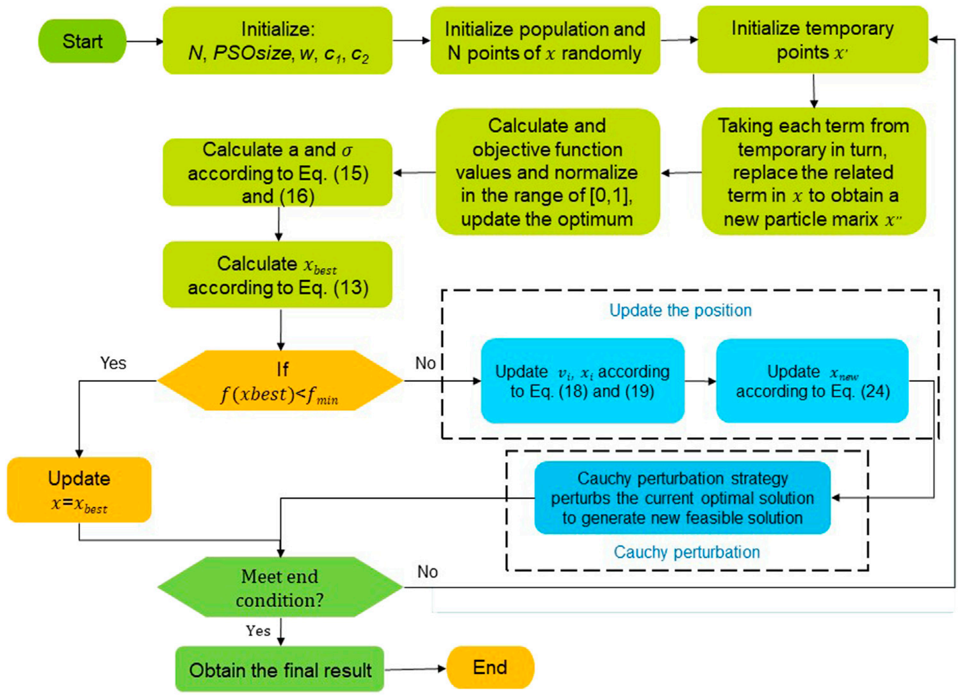

FIGURE 1
Flowchart representing the steps of HKSOPSO-CP.

An m-dimensional vector of $a = (a_1, a_2, \ldots a_n)$ is mapped from the n-dimensional vector w, that is $w = \phi(a)$. The slope of a hyperplane in m dimensions is the direction along which the optimal value lies, denoted by $\phi(a)$; then:

$$y = f(x) = w^T \cdot u + b = \varphi(a) \cdot \varphi(x) + b = K(a, x) + b \tag{11}$$

where $K(a, x)$ is the kernel function.

As a function in a higher-dimensional space, the optimal value is the solution of the lower-dimensional objective function.

The RBF function maps the objective function to an infinitely high-dimensional space (Basu and Basu, 2011) (KSO), which may transform a non-linear function into a linear function. Therefore, the RBF function is used:

$$\begin{aligned} k(x, y) &= \exp\left(\frac{\| x - a\|^2}{\sigma}\right),\ y = f(x) \\ &= K(a, x) + b = \exp\left(\frac{\| x - a\|^2}{\sigma}\right) \end{aligned} \tag{12}$$

The optimization results of the fitted kernel function are considered to approximately represent the minimum of the objective function; these results are converged on by several iterations to bring them close to the best possible result. At the same time, Eq. 12 provides an approximate solution to the optimization problem by a single iteration of the objective function.

The minimum value of the formula is given directly.

$$x_{best} \begin{cases} x_{min}\ \sigma < 0\ and\ a \geq \frac{1}{2}(x_{min} + x_{max}) \\ x_{max}\ \sigma < 0\ and\ a < \frac{1}{2}(x_{min} + x_{max}) \\ a\ \sigma > 0\ and\ x_{min} \leq a \leq x_{max} \\ x_{max}\ \sigma > 0\ and\ a > x_{max} \end{cases} \tag{13}$$

where $x \in [x_{min}, x_{max}]$, and the minimum is x_{best}.

Based on Eq. 13, it can be seen that the minimum value x_{best} is at the boundary, or vector a is equivalent to it, corresponding to the low-dimensional pre-image mapped to the high-dimensional hyperplane slope. The vector a is referred to as the kernel vector, which points the searches made in the iterative optimization process in a particular direction. Specifically, it is solved for *via* Eq. 14.

$$\sigma \ln(y - b) = \|x - a\|^2 = (x_1 - a_1)^2 + (x_2 - a_2)^2 + \ldots (x_n - a_n)^2 \tag{14}$$

To solve Eq. 14, posit a random vector $temp1 = (x_1', x_2', \ldots x_n')$; the i th item of the primitive vector x is replaced by the corresponding item of temp1, forming a new vector $(x_1, x_2, \ldots x_i', \ldots x_n)$. The new vectors form a matrix whose function is y_i'.

$$x' = \begin{pmatrix} x_1', x_2, \ldots x_i, \ldots x_n \\ \ldots \\ x_1, x_2, \ldots x_i', \ldots x_n \\ \ldots \\ x_1, x_2 \ldots x_i, \ldots x_n' \end{pmatrix}$$

Then,

TABLE 1 Unimodal test functions.

Test function	Function body	Search range
Sphere	$F_1(X) = \sum_{i=1}^{n} x_i^2$	$[-100, 100]^n$
Schwefel 2.22	$F_2(X) = \sum_{i=1}^{n} \lvert x_i \rvert + \prod_{i=1}^{n} \lvert x_i \rvert$	$[-10, 10]^n$
Schwefel 1.2	$F_3(X) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$	$[-100, 100]^n$
Schwefel 2.21	$F_4(X) = \max_i \{\lvert x_i \rvert, 1 \le i \le n\}$	$[-100, 100]^n$
Rosenbrock	$F_5(X) = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^n$
Step	$F_6(X) = \sum_{i=1}^{n} [(x_i + 0.5)]^2$	$[-100, 100]^n$
Quartic	$F_7(X) = \sum_{i=1}^{n} i x_i^4 + random[0, 1)$	$[-1.28, 1.28]^n$

TABLE 2 Multimodal test functions.

Test function	Function body	Search range
Schwefel 2.26	$F_8(X) = \sum_{i=1}^{n} - x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500, 500]^n$
Rastrigin	$F_9(X) = \sum_{i=1}^{n} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$
Ackley	$F_{10}(X) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}\right)$ $-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^n$
Griewank	$F_{11}(X) = \frac{1}{4000}\sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}}) + 1$	$[-600, 600]^n$
Penalized	$F_{12}(X) = \frac{\pi}{n}\left\{\sum_{i=1}^{n} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})]\right.$ $\left.+10\sin(\pi y_1) + (y_n - 1)^2\right\}$ $+\sum_{i=1}^{n} u(x_i, 10, 100, 4)$	$[-50, 50]^n$
	$y_i = 1 + \frac{x_i + 1}{4}$	
	$u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 - a < x_i < a \\ k(-x_i - a)^m & x_i < a \end{cases}$	
Penalized 2	$F_{13}(X) = \sum_{i=1}^{n} u(x_i, 5, 100, 4) + 0.1\{\sin^2(3\pi x_1)$ $+(x_n - 1)^2[1 + \sin^2(2\pi x_n)]$ $+\sum_{i=1}^{n} (x_i - 1)^2 [1 + \sin^2(3\pi x_i + 1)]\}$	$[-50, 50]^n$

$$a_i = \frac{1}{2}\left[x_i + x_j' - \sigma \ln\left(\frac{y-b}{y_i'-b}\right) \Big/ \left(x_i - x_i'\right)\right] \tag{15}$$

Similarly, positing a further random vector $temp2 = (x_1'', x_2'', \ldots x_n'')$, conduct the same operations following $temp1$. Then,

$$\sigma = \frac{x_j' - x_j''}{\ln\left(\frac{y-b}{y_j'-b}\right) \Big/ \left(x_j - x_j'\right) - \ln\left(\frac{y-b}{y''-b}\right) \Big/ \left(x_j - x_j''\right)} \tag{16}$$

After obtaining a and σ, x is updated for the next iteration as follows:

$$xnew = \begin{cases} xbest \text{ If } f(xbest) < f(xgbest) \\ xgbest + rand^* \exp\left(\frac{-t}{T_{\max}}\right)^* (xbest - xgbest) \quad \text{Else} \end{cases} \tag{17}$$

where t and T_{max} are the current and maximum number of iterations, respectively.

3.2 Overview of the PSO algorithm

The particle swarm optimization (PSO) algorithm was designed and developed by Eberhart and Kennedy (1995) based on the simulation of social behavior (Kennedy and Eberhart, 1995). Specifically, PSO was inspired by a simulation of the flocking behaviors of birds and uses velocity and position equations to search for optimal behavior. Social and cognitive factors influence the velocity of each particle. In other words, one notable aspect of PSO is the trade-off between the current and previous best options. The position of the particle's optimum fitness value is regarded as the local best ($pbest$), while the position with the highest fitness value is regarded as the global best ($gbest$). In the search space, any particle is treated as a point. All particles attempt to update their positions using the following model:

TABLE 3 Multimodal test functions with fixed dimensionality.

Test function	Function body	Search range
Foxholes	$F_{14}(X) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j+\sum_{i=1}^{2}(x_i-a_{ij})^6}\right)^{-1}$	$[-65.5, 65.5]^2$
	$a_{ij} = \begin{pmatrix} -32, -16, 0, 16, 32, -32, ..., 0, 16, 32 \\ -32, -32, -32, -32, -16, ..., 32, 32 \end{pmatrix}$	
Kowalik	$F_{15} = \sum_{i=1}^{11} [a_i - \frac{x_1(b_i^2+b_i x_2)}{b_i^2+b_i x_3+x_4}]^2$	$[-5, 5]^4$
	$a_i = \{0.1957, 0.1947, 0.1735, 0.16, 0.0844, 0.0627, 0.0456, 0.0342, 0.0323, 0.0235, 0.0246\}$	
	$b_i^{-1} = \{0.25, 0.5, 1, 2, 4, 6, 8, 10, 12, 14, 16\}$	
Six-hump Camel Branin	$F_{16}(X) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1 x_2 - 4x_2^2 + 4x_2^4$	$[-5, 5]^2$
	$F_{17}(X) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	$[-5, 15]^2$
Goldstein–Price	$F_{18}(X) = [1 + (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \times (x_1 + x_2 + 1)^2] \times [30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	$[-5, 5]^2$
Hartmann 3	$F_{19}(X) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2)$	$[0, 1]^3$
	$c_i = \{1, 1.2, 3, 3.2\}$	
	$a_{ij} = \begin{pmatrix} 3, 10, 30 \\ 0.1, 10, 35 \\ 3, 10, 30 \\ 0.1, 10, 30 \end{pmatrix} p_{ij} = \begin{pmatrix} 0.3689, 0.117, 0.2673 \\ 0.4699, 0.4387, 0.747 \\ 0.1091, 0.8732, 0.5547 \\ 0.03815, 0.5743, 0.8828 \end{pmatrix}$	
Hartmann 6	$F_{20}(X) = -\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2)$	$[0, 1]^6$
	$p_{ij} = \begin{pmatrix} 0.1312, 0.1696, 0.5569, 0.0124, 0.8283, 0.5886 \\ 0.2329, 0.4135, 0.8307, 0.3736, 0.1004, 0.9991 \\ 0.2348, 0.1415, 0.3522, 0.2883, 0.3047, 0.6650 \\ 0.4047, 0.8828, 0.8732, 0.5743, 0.1091, 0.0381 \end{pmatrix} a_{ij} = \begin{pmatrix} 10, 3, 17, 3.5, 1.7, 8 \\ 0.05, 10, 17, 0.1, 8, 14 \\ 3, 3.5, 1.7, 10, 17, 8 \\ 17, 8, 0.05, 10, 0.1, 14 \end{pmatrix}$	
Shekel 5	$F_{21}(X) = -\sum_{i=1}^{5} \left[(X - a_i)(X - a_i)^T + c_i\right]^{-1} a_{ij}^T = \begin{pmatrix} 4, 1, 8, 6, 3, 2, 5, 8, 6, 7 \\ 4, 1, 8, 6, 7, 9, 5, 1, 2, 3.6 \\ 4, 1, 8, 6, 3, 2, 3, 8, 6, 7 \\ 4, 1, 8, 6, 7, 9, 3, 1, 2, 3.6 \end{pmatrix}$	$[0, 10]^4$
	$c_i = \{1, 0.2, 0.2, 0.4, 0.4, 0.6, 0.3, 0.7, 0.5, 0.5\}$	
Shekel 7	$F_{22}(X) = -\sum_{i=1}^{7} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	$[0, 10]^4$
	a_{ij} & c_i as F_{21}	
Shekel 10	$F_{23}(X) = -\sum_{i=1}^{10} [(X - a_i)(X - a_i)^T + c_i]^{-1}$	$[0, 10]^4$
	a_{ij} & c_i as F_{21}	

$$v_i^k(t+1) = w(t)v_i^k(t) + c_1 r_1\left(pbest_i^k(t) - x_i^k(t)\right) + c_2 r_2\left(gbest^k(t) - x_i^k(t)\right) \tag{18}$$

$$x_i^k(t+1) = x_i^k(t) + v_i^k(t+1) \tag{19}$$

where v_i^k is the velocity of the i th particle at the k th iteration represented by an n-dimensional vector; r_1 and r_2 are random numbers between 0 and 1; w is the factor of inertia weight; c_1 and c_2 are the acceleration constants; $pbest_i$ is the individual best fitness value achieved by the i th particle at the k th iteration, based on its own experience; $gbest^k$ is the best particle position based on all individual best positions; x_i^k is the position of the i th particle at the k th iteration; and t is the current number of iterations.

In the initialization process, the particle swarm optimization algorithm initializes particles randomly in the search space and approaches the optimal position continuously through iteration. At each iteration, the velocity of the particle is calculated *via* Eq. 18, and the position of the particle is then updated *via* Eq. 19. The process of updating the particle's position continues until the stop condition is met.

3.3 Cauchy perturbation

A Cauchy perturbation is derived from a Cauchy distribution, which has the following probability density (Cima et al., 2006):

$$f(x; x0, \gamma) = \frac{1}{\pi^* \gamma\left[1 + \left(\frac{x - x0}{\gamma}\right)^2\right]} \tag{20}$$

10.3389/fenrg.2022.1061408

TABLE 4 The results of the comparison between methods with 23 CEC benchmark functions.

F	Measure	COCA	BWOA	RSA	SOA	SSA	TSA	KSO	HKSOPSO-CP
F1	Worst	1.0190E+00	3.4358E−05	.0000E+00	2.9894E−21	3.3675E−02	6.4319E−38	2.6139E−09	.0000E+00
	Average	9.6188E−01	3.4358E−05	.0000E+00	1.2293E−21	1.1246E−02	6.0539E−38	2.5734E−09	.0000E+00
	Best	8.7604E−01	3.4358E−05	.0000E+00	1.2428E−23	4.3367E−22	5.6760E−38	2.5412E−09	.0000E+00
F2	Worst	1.4761E−01	1.2711E−03	.0000E+00	1.6758E−13	6.7206E+00	5.1187E−22	5.5587E−03	.0000E+00
	Average	1.3751E−01	9.1668E−04	.0000E+00	7.9835E−14	2.2970E+00	2.9663E−22	5.4972E−03	.0000E+00
	Best	1.2080E−01	2.0793E−04	.0000E+00	1.8978E−14	8.8133E−10	8.1388E−23	5.4564E−03	.0000E+00
F3	Worst	4.7661E+02	7.5800E+02	.0000E+00	2.3467E−09	8.4526E−01	1.3895E−06	4.5216E+03	.0000E+00
	Average	4.6775E+02	7.5800E+02	.0000E+00	8.6919E−10	2.8342E−01	1.3895E−06	3.2768E+03	.0000E+00
	Best	4.6292E+02	7.5800E+02	.0000E+00	2.8856E−11	6.4615E−06	1.3895E−06	2.0556E+03	.0000E+00
F4	Worst	1.5629E+01	2.6031E+01	.0000E+00	5.1912E−05	6.4387E−02	6.6044E+00	4.6851E−01	.0000E+00
	Average	1.5049E+01	2.2908E+01	.0000E+00	1.9482E−05	2.5853E−02	5.2396E+00	4.1694E−01	.0000E+00
	Best	1.4336E+01	1.6661E+01	.0000E+00	7.6197E−07	5.9136E−04	3.8748E+00	3.6652E−01	.0000E+00
F5	Worst	4.8037E+02	5.3968E+02	4.8990E+01	4.8811E+01	4.8837E+01	4.8768E+01	4.6289E+01	4.6618E+01
	Average	4.1375E+02	5.3968E+02	4.8988E+01	4.8724E+01	2.3116E+01	4.8673E+01	4.5316E+01	4.5274E+01
	Best	3.5079E+02	5.3968E+02	4.8982E+01	4.8559E+01	3.4468E+00	4.8578E+01	4.3608E+01	4.4602E+01
F6	Worst	3.3000E+01	2.8869E−06	1.2250E+01	6.9530E+00	1.3451E+00	5.6785E+00	.0000E+00	.0000E+00
	Average	2.7000E+01	9.6425E−07	1.2151E+01	6.6732E+00	8.7289E−01	5.4443E+00	.0000E+00	.0000E+00
	Best	1.7000E+01	3.8777E−15	1.2088E+01	6.2597E+00	1.3280E−01	5.2101E+00	.0000E+00	.0000E+00
F7	Worst	7.2903E−02	4.3757E−04	4.8945E-04	8.3872E-04	5.1586E−02	6.1563E−03	5.2977E−03	9.7939E−06
	Average	5.8907E−02	3.6478E−04	2.0086E-04	3.9795E-04	2.6449E−02	6.1563E−03	4.6117E−03	5.5547E−06
	Best	4.9296E−02	2.5492E−04	4.3791E-05	1.7271E-04	7.4521E−03	6.1563E−03	4.2214E−03	4.3753E−07
F8	Worst	−2.0942E+04	−1.1306E+04	−8.3071E+03	−6.4988E+03	−3.6387E+03	−1.0394E+04	−2.0949E+04	−1.1168E+04
	Average	−2.0943E+04	−1.1682E+04	−8.6683E+03	−7.1780E+03	−4.3875E+03	−1.0394E+04	−2.0949E+04	−1.5465E+04
	Best	−2.0944E+04	−1.1982E+04	−8.8489E+03	−7.7147E+03	−5.4402E+03	−1.0394E+04	−2.0949E+04	−2.0948E+04
F9	Worst	1.0429E+01	2.2338E−04	.0000E+00	1.0001E+00	3.3536E+02	2.2083E+02	1.2885E−07	.0000E+00
	Average	9.4712E+00	7.4499E−05	.0000E+00	3.3336E−01	1.1184E+02	2.2083E+02	1.0416E−07	.0000E+00
	Best	8.8922E+00	4.3207E−08	.0000E+00	.0000E+00	7.3769E−08	2.2083E+02	8.1546E−08	.0000E+00
F10	Worst	3.5147E+00	2.7690E−03	8.8818E−16	1.9965E+01	1.1703E−01	2.2204E−14	7.8681E−05	8.8818E−16
	Average	3.3859E+00	9.8733E−04	8.8818E−16	1.9964E+01	3.9011E−02	2.2204E−14	7.4568E−05	8.8818E−16
	Best	3.1851E+00	2.6579E−06	8.8818E−16	1.9963E+01	9.3420E−07	2.2204E−14	6.7371E−05	8.8818E−16
F11	Worst	6.6468E−01	1.7682E−02	.0000E+00	.0000E+00	6.0801E−03	.0000E+00	3.6575E−06	.0000E+00
	Average	6.3190E−01	6.5841E−03	.0000E+00	.0000E+00	2.0267E−03	.0000E+00	1.9180E−06	.0000E+00
	Best	5.8027E−01	1.2798E−04	.0000E+00	.0000E+00	3.9468E−11	.0000E+00	4.4038E−07	.0000E+00
F12	Worst	3.6182E−01	4.3062E−18	2.1670E+00	7.2713E−01	5.5007E−02	1.0597E+01	3.8466E−09	4.4640E−05
	Average	1.2921E−01	2.8755E−18	1.3511E+00	5.2736E−01	2.1451E−02	1.0597E+01	2.5625E−09	3.8702E−05
	Best	9.3685E−03	1.4237E−20	9.2783E−01	3.4283E−01	3.1369E−03	1.0597E+01	1.3585E−09	2.7777E−05
F13	Worst	1.0963E+01	1.5961E−05	7.5999E−19	3.9651E+00	3.0838E−01	5.2188E+00	1.2402E−06	2.4314E−03
	Average	9.0997E+00	7.9422E−06	2.5333E−19	3.8584E+00	2.3810E−01	5.2188E+00	5.7049E−07	1.2380E−03
	Best	6.7712E+00	4.7000E−09	3.3219E−32	3.6863E+00	1.4956E−01	5.2188E+00	1.0300E−07	5.7995E−04

(Continued on following page)

TABLE 4 (*Continued*) **The results of the comparison between methods with 23 CEC benchmark functions.**

F	Measure	COCA	BWOA	RSA	SOA	SSA	TSA	KSO	HKSOPSO-CP
F14	Worst	9.9800E−01	2.9822E+00	5.9721E+00	2.9821E+00	1.2671E+01	1.2671E+01	9.9800E−01	9.9800E−01
	Average	9.9800E−01	2.6522E+00	4.2711E+00	1.6594E+00	1.2671E+01	1.2671E+01	9.9800E−01	9.9800E−01
	Best	9.9800E−01	1.9920E+00	2.9821E+00	9.9800E−01	1.2671E+01	1.2671E+01	9.9800E−01	9.9800E−01
F15	Worst	3.0749E−04	3.7326E−03	2.3100E−02	1.2273E−03	1.1656E−03	4.5817E−04	4.3803E−04	3.0749E−04
	Average	3.0749E−04	2.0324E−03	8.5860E−03	1.2256E−03	8.7099E−04	4.5817E−04	3.8605E−04	3.0749E−04
	Best	3.0749E−04	1.1058E−03	9.7933E−04	1.2239E−03	6.8563E−04	4.5817E−04	3.0749E−04	3.0749E−04
F16	Worst	−1.0316E+00	−1.0306E+00	−1.0173E+00	−1.0316E+00	−1.0300E+00	−1.0316E+00	−1.0316E+00	−1.0316E+00
	Average	−1.0316E+00	−1.0313E+00	−1.0257E+00	−1.0316E+00	−1.0308E+00	−1.0316E+00	−1.0316E+00	−1.0316E+00
	Best	−1.0316E+00	−1.0316E+00	−1.0316E+00	−1.0316E+00	−1.0315E+00	−1.0316E+00	−1.0316E+00	−1.0316E+00
F17	Worst	3.9789E−01	3.9796E-01	4.8177E−01	3.9799E−01	3.9917E−01	3.9789E−01	3.9789E−01	3.9789E−01
	Average	3.9789E−01	3.9791E-01	4.5381E−01	3.9795E−01	3.9892E−01	3.9789E−01	3.9789E−01	3.9789E−01
	Best	3.9789E−01	3.9789E-01	3.9789E−01	3.9789E−01	3.9844E−01	3.9789E−01	3.9789E−01	3.9789E−01
F18	Worst	3.0000E+00	5.2787E+00	3.0087E+00	3.0001E+00	3.0520E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Average	3.0000E+00	3.7606E+00	3.0051E+00	3.0000E+00	3.0413E+00	3.0000E+00	3.0000E+00	3.0000E+00
	Best	3.0000E+00	3.0000E+00	3.0002E+00	3.0000E+00	3.0302E+00	3.0000E+00	3.0000E+00	3.0000E+00
F19	Worst	−3.8628E+00	−3.8446E+00	−3.6018E+00	−3.8548E+00	−3.8559E+00	−3.8628E+00	−3.8628E+00	−3.8628E+00
	Average	−3.8628E+00	−3.8525E+00	−3.7550E+00	−3.8549E+00	−3.8581E+00	−3.8628E+00	−3.8628E+00	−3.8628E+00
	Best	−3.8628E+00	−3.8621E+00	−3.8621E+00	−3.8549E+00	−3.8596E+00	−3.8628E+00	−3.8628E+00	−3.8628E+00
F20	Worst	−3.3220E+00	−2.9579E+00	−1.4363E+00	−3.0155E+00	−2.9366E+00	−3.2016E+00	−3.3220E+00	−3.3220E+00
	Average	−3.3220E+00	−3.1888E+00	−2.0529E+00	−3.0155E+00	−3.0816E+00	−3.2016E+00	−3.3220E+00	−3.3220E+00
	Best	−3.3220E+00	−3.3144E+00	−2.4916E+00	−3.0156E+00	−3.1818E+00	−3.2016E+00	−3.3220E+00	−3.3220E+00
F21	Worst	−1.0153E+01	−9.6985E+00	−5.0552E+00	-4.9730E-01	−4.8863E+00	−2.6476E+00	−1.0153E+01	−1.0153E+01
	Average	−1.0153E+01	−9.9224E+00	−5.0552E+00	−6.9167E+00	−6.4721E+00	−6.3110E+00	−1.0153E+01	−1.0153E+01
	Best	−1.0153E+01	−1.0153E+01	−5.0552E+00	−1.0133E+01	−7.9217E+00	−9.9745E+00	−1.0153E+01	−1.0153E+01
F22	Worst	−1.0403E+01	−1.0403E+01	−5.0877E+00	−5.2393E-01	−6.5016E+00	−1.0106E+01	−1.0403E+01	−1.0403E+01
	Average	−1.0403E+01	−1.0403E+01	−5.0877E+00	−5.3207E+00	−8.1081E+00	−1.0111E+01	−1.0403E+01	−1.0403E+01
	Best	−1.0403E+01	−1.0403E+01	−5.0877E+00	−1.0351E+01	−9.8519E+00	−1.0116E+01	−1.0403E+01	−1.0403E+01
F23	Worst	−1.0536E+01	−8.8018E+00	−5.1285E+00	−1.0470E+01	−7.2908E+00	−3.8107E+00	−1.0536E+01	−1.0536E+01
	Average	−1.0536E+01	−9.7699E+00	−5.1285E+00	−1.0503E+01	−8.9942E+00	−7.1265E+00	−1.0536E+01	−1.0536E+01
	Best	−1.0536E+01	−1.0535E+01	−5.1285E+00	−1.0521E+01	−1.0021E+01	−1.0442E+01	−1.0536E+01	−1.0536E+01

where x_0 is the position parameter defining the peak of the distribution, and γ is the scale parameter at half of the maximum value.

When $x_0 = 0$, $\gamma = 1$, the distribution is considered a standard Cauchy distribution.

$$cauchy\,(0, 1) = \frac{1}{\pi^{*}\,(1 + x^{2})} \tag{21}$$

The Cauchy distribution takes a very wide range of values, and even very large values have some probability of occurring. The Cauchy distribution approaches 0 more slowly than the normal distribution, so the Cauchy mutation has better disturbance ability than the normal variation. The Cauchy mutation is introduced into the target position updating technique, and the Cauchy operator uses its perturbation ability to enhance the algorithm's global optimization performance.

3.4 Hybrid HKSOPSO-CP algorithm with Cauchy perturbation

The proposed HKSOPSO-CP algorithm is put forward in order to enhance the ability of the primitive KSO approach to identify local optima through hybridization with the PSO algorithm and to enhance the range of feasible solutions identified by the algorithm using Cauchy perturbation.

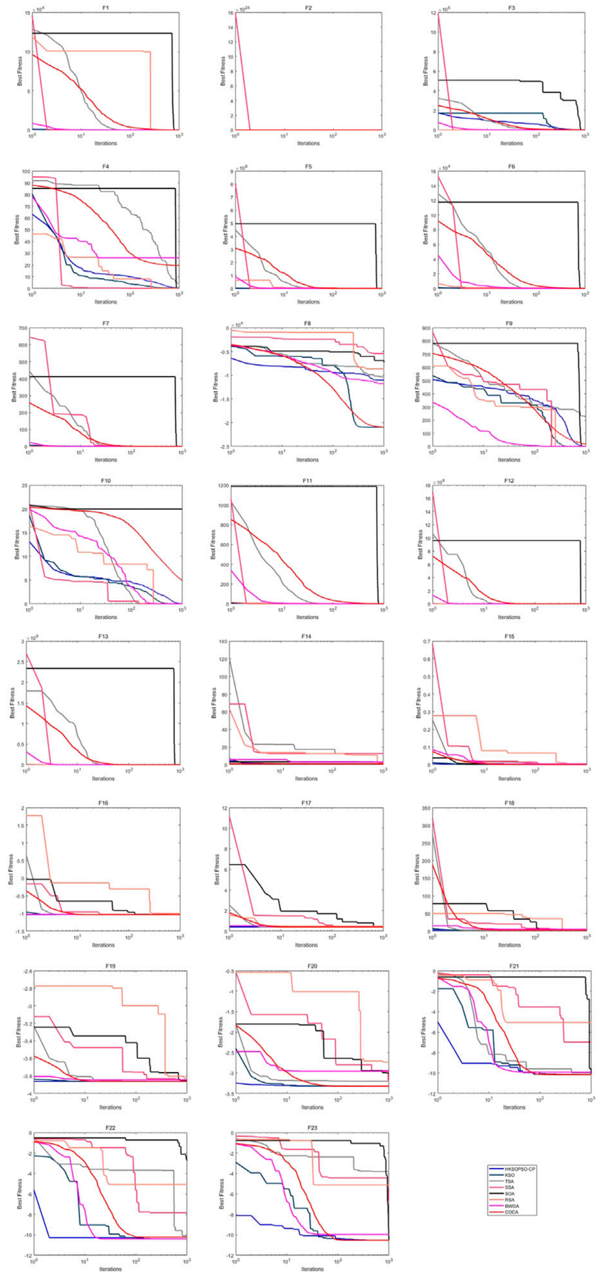

FIGURE 2
The convergence behavior of the compared methods with 23 CEC functions.

The KSO algorithm projects the objective function from low-dimensional to high-dimensional space. With the increase in dimensionality, the non-linear objective function gradually tends toward becoming linear. Compared with other algorithms, the KSO algorithm has good global search ability and a good optimization effect when facing high-dimension objective functions, but it has no obvious advantage when dealing with a low-dimension objective function. The PSO algorithm has strong optimization ability when the dimensionality of the objective function is low and has good local search ability. The advantages of the PSO algorithm and KSO algorithm are combined to improve the local search ability of KSO so as to balance exploration and exploitation. At the same time, the Cauchy distribution is utilized to expand the distribution range of feasible solutions and further improve the global search ability of the algorithm when dealing with various scenarios.

The KSO update formula Eq. 17 is combined with the PSO update formula Eq. 18, resulting in the following proposed update formula:

$$x_{new1} = gbest + rand*(a - gbest) * \exp\left(\frac{t}{T_{max}}\right) + v_i^k \tag{22}$$

Subsequently, the Cauchy mutation operator is employed to perturb and mutate the new solution obtained by Eq. 22, and the update formula is calculated by Eq. 23, which improves the algorithm's capacity to jump out of local optimum situations.

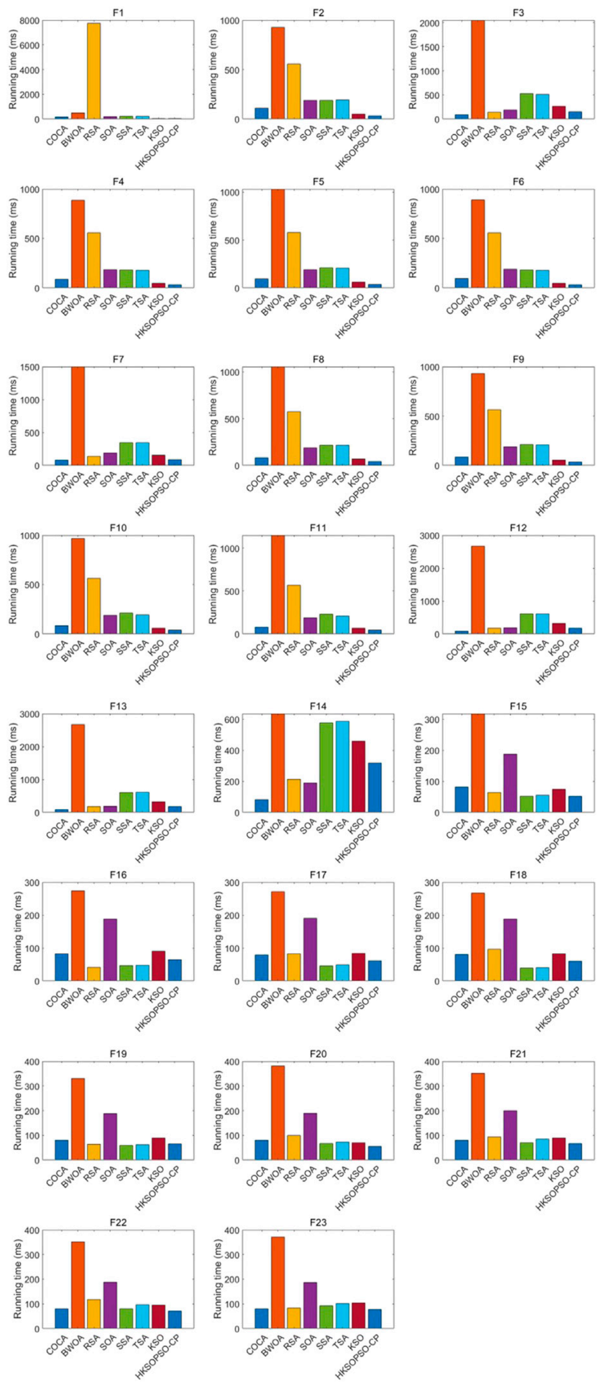

FIGURE 3
CPU running time consumed by the various algorithms entered into comparison.

$$x_{new2} = x_{new1} + cauchy(0,1) * x_{new1} \tag{23}$$

Finally, the full update formula of the HHKSOPSO-CP algorithm with Cauchy perturbation can be expressed as follows:

$$xnew = \begin{cases} xnew1 & If\ f(xnew1) < f(xnew2) \\ xnew2 & Else \end{cases} \tag{24}$$

The HKSOPSO-CP algorithm seeks out the optimal value by exploring the activity of many particles within the feasible region. In the initialization phase, the algorithm generates particles randomly in the feasible region and approaches the optimal position continuously through iteration. In each iteration, the velocity of each particle is calculated using Eq. 18; the position of the particle is then updated using Eq. 24, and finally the distribution range of the solution is extended by Cauchy mutation. The particle updating behavior does not stop until the end condition is met.

In summary, this paper describes the process of using the HKSOPSO-CP algorithm to solve the EED problem, which involves the following steps:

TABLE 5 Best solution values for a six-unit system with different weights.

w	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
P_1	.4108	.3626	.3213	.2853	.2537	.2256	.2004	.1777	.1570	.1381	.1207
P_2	.4635	.4369	.4130	.3916	.3723	.3547	.3387	.3241	.3106	.2982	.2866
P_3	.5442	.5496	.5549	.5600	.5647	.5688	.5725	.5756	.5782	.5801	.5815
P_4	.3902	.4776	.5548	.6242	.6872	.7454	.7997	.8510	.9000	.9472	.9934
P_5	.5443	.5470	.5488	.5495	.5492	.5479	.5454	.5417	.5369	.5310	.5237
P_6	.5153	.4911	.4694	.4500	.4324	.4164	.4017	.3882	.3757	.3641	.3532
V(MW)	2.15E−16	1.77E−16	4.16E−17	2.19E−16	9.71E−17	7.63E−17	5.90E−17	1.11E−16	2.19E−16	2.08E−16	1.46E−16
PL(MW)	.0344	.0308	.0283	.0266	.0255	.0248	.0244	.0243	.0244	.0247	.0251
Fuel cost ($/h)	645.9986	634.7793	626.4476	620.2255	615.5797	612.1370	609.6320	607.8735	606.7232	606.0816	605.8791
Emissions (ton/h)	.1942	.1947	.1962	.1983	.2008	.2036	.2066	.2099	.2133	.2169	.2207

Step 1: Initialize HKSOPSO-CP parameters, including N, PSO_{size}, c_1, c_2, w.

Step 2: Generate a random population of N particles and initialize particle x.

Step 3: Calculate the current power of the generator for every particle in the current population; calculate the objective function values of x, x', x'', and the generation matrix x.

Step 4: Calculate the fitness value for all particles.

Step 5: Calculate a and σ using Eq. 15 and Eq. 16.

Step 6: Calculate x_{best} according to Eq. 13 and determine whether it is the optimal solution.

Step 7: Calculate the velocity and position of all particles using Eq. 18 and Eq. 19.

Step 8: Generate new feasible solutions using Eq. 24.

Step 9: Go to Step 3 until the end condition is met.

A flowchart representing the steps of the HKSOPSO-CP algorithm is provided in Figure 1.

4 Experimental results

The effectiveness of the HKSOPSO-CP algorithm is verified by the CEC standard test system and EED problem with four test cases with and without valve points (Zou et al., 2017).

We conducted all tests under fair comparison conditions (Zheng et al., 2021a; Zheng et al., 2022a; Zheng et al., 2022b). A potentially helpful factor in this respect is that all variables in the computing system exert a uniform effect on the performance of the compared algorithms, as per other AI-based work (Zheng et al., 2021b; Zheng et al., 2021c). The performance of HKSOPSO-CP is compared with the original KSO, PSO, and other recent algorithms. All algorithms were implemented in Matlab 2020b and ran on an Intel Core CPU i7-10700@4.59GHz (8 CPU), 32 GB RAM system.

4.1 Comparison on CEC benchmark functions

In this section, the proposed HKSOPSO-CP algorithm is compared with the original KSO, PSO, and other algorithms proposed in recent years, such as SOA (Dhiman and Kumar, 2018), TSA (Kaur et al., 2020), SSA (Xue and Shen, 2020), RSA (Abualigah et al., 2021), BWOA (Hayyolalam and Pourhaji Kazem, 2019), and COCA (Pierezan and Coelho, 2018). Tables 1–3 give the 23 CEC benchmark functions used in the experimental study, including unimodal test functions, multimodal test functions, and multimodal test functions with fixed dimensionality.

Table 4 illustrates the experimental results, including the worst, average, and best results over 30 runs. It can be seen that HKSOPSO-CP achieved the best results in most of the tests. In terms of the best result, HKSOPSO-CP obtained better results with 19 of the test functions, followed by KSO and COCA. In terms of the average result, HKSOPSO-CP again achieved the best results with 19 of the test functions, followed by COCA and KSO. Its advantages are especially apparent in dealing with multimodal test functions. The results show that HKSOPSO-CP may be capable of outperforming the algorithms included in the comparison carried out here. Therefore, HKSOPSO-CP is an effective method for handling various complex problems that require a strong search ability.

Figure 2 shows the iteration curves of HKSOPSO-CP and other algorithms participating in the comparison. In most cases, HKSOPSO-CP exhibits a strong, fast convergence ability, with a greater capacity to jump out of local optima, and its convergence effect is stabler than that of other methods. Therefore, HKSOPSO-CP offers a strong global search capability.

Another measure of algorithm performance is CPU running time. Figure 3 shows the CPU running time consumed by the various algorithms entered into the comparison. It can be seen

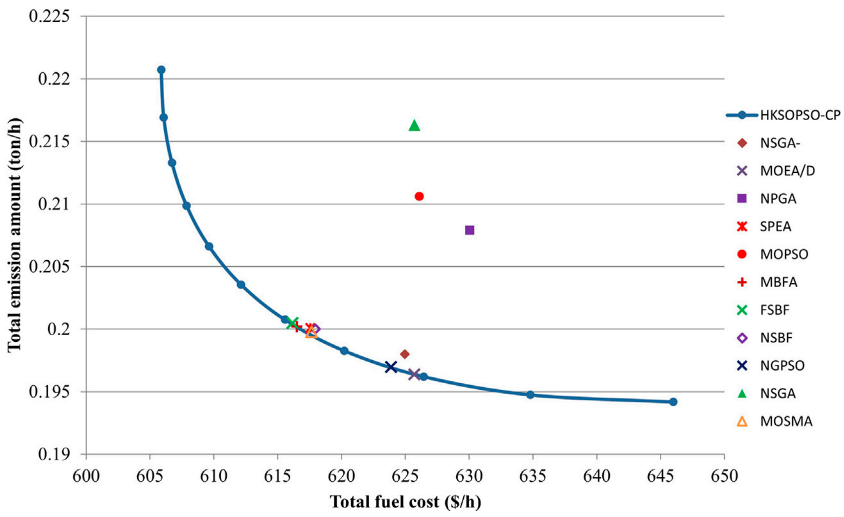

FIGURE 4
Case 1: Solutions generated by the HKSOPSO-CP algorithm, and the best compromise solutions provided by other algorithms.

TABLE 6 Case 1 comparison between different algorithms optimizing for fuel cost or emission rate.

Algorithm	Best fuel cost (w = 1.0)				Best pollution emissions (w = .0)			
	C($/h)	E(ton/h)	P_l (pu)	V(pu)	C($/h)	E(ton/h)	P_l (pu)	V(pu)
MBFA	606.1700	.2174	.0255	1.89E−05	643.84	.194201	.0345	2.51E−05
MSA	605.9984	.2207	.0256	3.96E−05	646.20	.194179	.0353	2.71E−05
PSOGSA	605.9984	.2207	.0256	6.10E−05	646.21	.194179	.0353	2.92E−05
MOPSO	607.8400	.2192	.0255	7.38E−03	642.90	.194230	.0346	3.82E−05
PSO(wsm)	607.8400	.2198	.0257	7.45E−03	645.23	.194230	.0352	4.13E−03
MOPSO-2	607.7900	.2193	.0257	7.56E−03	644.74	.194185	.0350	4.11E−03
GA	607.7814	.2199	.0256	7.58E−03	645.22	.194180	.0352	4.12E−03
NSGA	607.9800	.2191	.0265	8.07E−03	638.98	.194678	.0327	2.96E−03
HHO	606.5292	.2234	.0247	1.70E−16	631.05	.195728	.0281	1.59E−16
NPGA	608.0593	.2207	.0251	8.59E−03	644.23	.194270	.0355	4.06E−03
SPEA	607.8600	.2176	.0258	7.43E−03	644.77	.194279	.0347	4.66E−03
DE	608.0658	.2193	.0255	8.72E−03	645.09	.194181	.0352	4.80E−03
GSA	605.9984	.2207	.0256	1.37E−04	646.21	.194179	.0353	6.98E−05
OGSA	605.9982	.2207	.0256	5.69E−05	646.21	.194179	.0353	2.92E−05
SMODE	619.0700	.2034	.0216	2.49E−03	643.01	.194201	.0344	4.50E−03
BBMOPSO	605.9817	.2202	.0256	1.24E−04	646.48	.194179	.0354	2.92E−05
MOEA/D	619.5300	.2017	.0227	2.39E−03	644.98	.194187	.0348	5.02E−03
MOLBSA	606.0081	.2205	.0258	3.74E−07	645.02	.194200	.0347	2.32E−04
SOA	605.8791	.2208	.0344	1.18E−16	646.1517	.194179	.0345	1.11E−16
TSA	605.8799	.2209	.0251	1.21E−16	646.1772	.194180	.0344	2.08E−17
SMA	606.0020	.2206	.0256	5.06E−09	642.54	.194340	.0346	3.08E−08
ISMA	605.9980	.2207	.0256	8.59E−11	646.66	.194180	.0354	2.74E−10
KSO	605.8960	.2211	.0258	5.18E−04	646.60	.194178	.0353	2.09E−03
HKSOPSO-CP	605.8791	.2207	.0251	1.46E−16	645.99	.194178	.0344	2.15E−16

TABLE 7 Best solution values for a 10-unit system with different weights.

w	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1
P_1	55.00	55.00	55.00	55.00	55.00	55.00	55.00	55.00	55.00	55.00	55.00
P_2	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00
P_3	81.13	81.11	81.08	81.37	82.53	83.89	85.93	87.84	89.59	96.17	106.75
P_4	81.36	81.16	80.91	80.93	81.79	82.77	84.34	85.60	86.55	91.36	100.89
P_5	160.00	160.00	160.00	160.00	160.00	152.89	133.86	116.17	99.79	89.77	81.36
P_6	240.00	240.00	240.00	231.68	201.73	174.95	152.34	130.00	108.69	94.69	83.00
P_7	294.49	292.61	290.46	289.79	293.37	297.05	300.00	300.00	300.00	300.00	300.00
P_8	297.27	297.00	296.66	298.19	304.64	311.82	317.89	325.17	330.28	340.00	340.00
P_9	396.77	397.81	399.02	402.45	411.40	421.72	436.20	450.76	466.77	470.00	470.00
P_{10}	395.58	396.95	398.56	402.48	412.07	423.23	438.83	454.97	470.00	470.00	470.00
V(MW)	2.71E−12	2.79E−12	2.79E−12	2.84E−12	3.21E−12	3.58E−12	4.09E−12	4.80E−12	4.57−E3	5.01E−03	9.94E−03
PL(MW)	81.60	81.64	81.68	81.89	82.53	83.32	84.38	85.51	86.68	86.99	87.03
Fuel cost ($/h)	1,16,412	1,16,401	1,16,388	1,16,050	1,14,995	1,14,011	1,12,998	1,12,243	1,11,727	1,11,547	1,11,497
Emissions (ton/h)	3,932	3,932	3,932	3,944	3,991	4,060	4,162	4,279	4,407	4,487	4,572

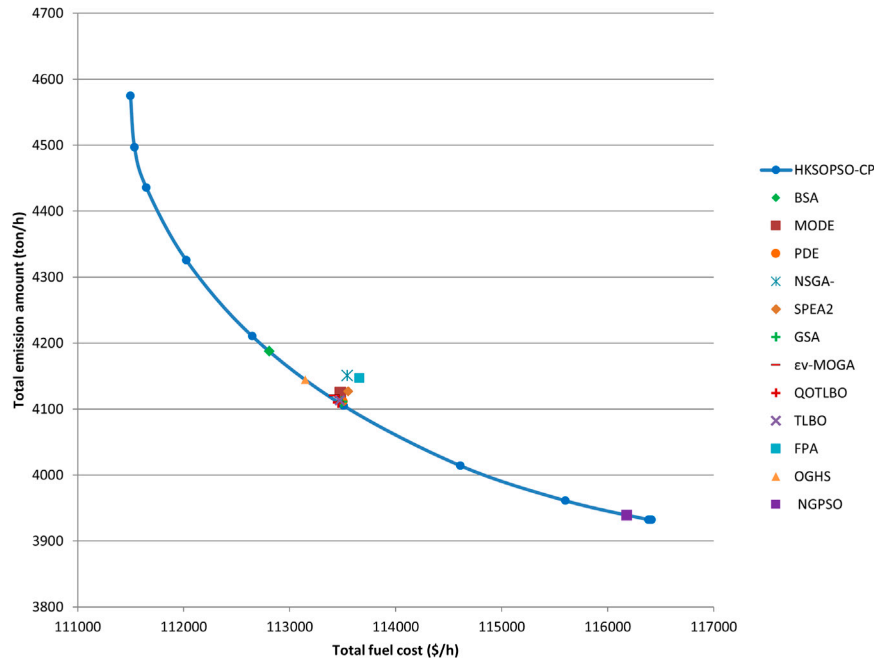

FIGURE 5
Case 2: solutions generated by the HKSOPSO-CP algorithm, and the best compromise solutions provided by other algorithms.

that across all 23 of the test functions, the amount of time consumed by HKSOPSO-CP is relatively small, and this method is associated with the shortest running time in the case of 14 tests, performing notably better than BWOA in particular. Consequently, HKSOPSO-CP can be judged to provide better performance than KSO and other algorithms, as well as offering an improved degree of stability.

4.2 Comparison of performance on the EED problem

The four test cases for the EED problem are: 1) a 6-unit system ($P_D = 2.834pu$); 2) a 10-unit system (P_D = 2000 MW); 3) a 40-unit system (P_D = 10500 MW); and 4) a 110-unit system (P_D = 15000 MW).

TABLE 8 Case 2 comparison between different algorithms optimizing for fuel cost.

Algorithm	OGHS	NGPSO	SOA	TSA	SMA	ISMA	KSO	HKSOPSO-CP
P_1	55.00	55.00	55.00	55.00	55.00	55.00	55.00	55.00
P_2	80.00	80.00	80.00	80.00	80.00	80.00	80.00	80.00
P_3	106.99	106.94	107.58	108.19	106.84	106.87	106.84	106.75
P_4	100.54	100.58	100.92	101.60	100.88	100.54	100.92	100.89
P_5	81.45	81.50	81.58	80.20	81.50	81.61	81.32	81.36
P_6	83.07	83.02	81.96	82.12	82.82	83.02	82.95	83.00
P_7	300.00	300.00	300.00	300.00	300.00	300.00	300.00	300.00
P_8	340.00	340.00	340.00	340.00	340.00	340.00	340.00	340.00
P_9	470.00	470.00	470.00	470.00	470.00	470.00	470.00	470.00
P_{10}	470.00	470.00	470.00	469.94	470.00	470.00	470.00	470.00
V(MV)	.00	.00	.00	.00	.00	.00	.01	.00
PL(MW)	87.04	87.04	87.04	87.04	87.04	87.04	87.04	87.03
Fuel cost ($/h)	111497.61	111497.63	111497.65	111498.68	111497.65	111497.63	111497.27	111496.97
Emissions (ton/h)	4572.27	4572.20	4576.97	4582.93	4573.06	4571.68	4573.24	4572.67

TABLE 9 Case 2 comparison between different algorithms optimizing for emission rate.

Algorithm	OGHS	NGPSO	SOA	TSA	SMA	ISMA	KSO	HKSOPSO-CP
P_1	55.00	55.00	55.00	55.00	55.00	55.00	55.00	55.00
P_2	80.00	80.00	80.00	80.00	80.00	79.99	80.00	80.00
P_3	81.11	81.13	81.14	80.39	80.80	81.25	81.13	81.13
P_4	81.41	81.36	81.43	80.72	81.22	81.42	81.36	81.36
P_5	160.00	160 .0000	160.00	160.00	160.00	160.00	160 .0000	160.00
P_6	240.00	240.00	240.00	240.00	240.00	240.00	240.00	240.00
P_7	294.51	294.49	294.26	294.50	295.34	294.49	294.49	294.49
P_8	297.26	297.27	297.46	299.34	297.78	296.94	297.27	297.27
P_9	396.74	396.77	396.93	395.89	395.07	397.04	396.77	396.77
P_{10}	395.57	395.58	395.37	395.76	396.40	395.47	395.58	395.58
V(MV)	.00	.00	.00	.00	.00	.00	.00	2.71E−12
PL(MV)	81.59	81.60	81.59	81.61	81.60	81.59	81.60	81.60
Fuel cost ($/h)	116412.65	116412.44	116412.13	116413.24	116416.26	116412.21	116412.44	116412.44
Emissions (ton/h)	3932.24	3932.24	3932.24	3932.33	3932.28	3932.24	3932.24	3932.24

4.2.1 Case 1 comparison

In case 1, an optimization algorithm such as HKSOPSO-CP is applied to a 6-unit system with a load demand of 2.834 pu, along with other optimization algorithms handling six generators, considering the valve point effect.

Table 5 displays the results of the best solution for case 1 with a range of values for weight factor w between 0 and 1 and a step size of .1. The constraints are the generating constraint V and the transmission losses constraint P_l. As shown in this table, when the only objective function is to minimize fuel costs, HKSOPSO-CP achieves an optimal value of 605.8791($/h). When only emissions are minimized, HKSOPSO-CP achieves an optimal value of .194178 (ton/h).

In case 1, the HKSOPSO-CP algorithm is applied to obtain the common optimal values of fuel cost C and pollution emissions E. Figure 4 depicts the Pareto optimal values and also plots the optimal solutions provided by other algorithms. Taking the graphs and tables

TABLE 10 Best solution values for a 40-unit system with different weights.

w	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
P_1	114.00	114.00	114.00	114.00	114.00	114.00	110.80	110.88	112.08	111.58	110.91
P_2	114.00	114.00	114.00	114.00	114.00	114.00	110.80	111.15	112.46	110.80	111.40
P_3	120.00	120.00	120.00	120.00	120.00	120.00	120.00	97.40	97.73	97.43	97.44
P_4	169.37	176.51	179.73	179.73	179.73	179.73	179.73	179.73	179.73	179.73	179.73
P_5	97.00	97.00	97.00	97.00	97.00	97.00	87.80	95.81	93.98	88.15	88.19
P_6	124.26	128.13	134.06	140.00	140.00	140.00	140.00	140.00	140.00	140.00	140.00
P_7	299.71	300.00	300.00	300.00	300.00	300.00	259.60	260.03	293.80	259.63	259.62
P_8	297.91	299.08	300.00	300.00	300.00	284.60	284.60	284.60	285.29	284.75	284.60
P_9	297.26	298.24	300.00	300.00	300.00	284.60	284.60	286.28	285.12	286.33	284.60
P_{10}	130.00	130.00	130.00	130.00	130.00	130.00	130.00	130.00	130.00	130.00	130.00
P_{11}	298.41	305.51	314.10	318.40	318.40	318.40	243.60	242.46	243.47	168.78	168.80
P_{12}	298.03	304.96	313.51	318.40	318.40	318.40	243.60	243.55	168.80	243.32	168.80
P_{13}	433.56	433.61	438.48	394.28	394.28	394.28	394.28	304.52	214.76	214.76	214.76
P_{14}	421.73	413.47	394.28	394.28	394.28	394.28	394.28	394.26	304.52	304.52	394.28
P_{15}	422.78	415.44	394.28	394.28	394.28	394.28	394.28	393.85	304.52	394.07	394.28
P_{16}	422.78	415.44	394.28	394.28	394.28	394.28	394.28	394.25	304.52	304.52	304.52
P_{17}	439.41	448.40	467.01	485.35	489.28	489.28	489.28	489.26	489.28	489.28	489.28
P_{18}	439.40	448.42	467.04	485.38	489.28	489.28	489.28	489.27	489.28	489.28	489.28
P_{19}	439.41	437.95	434.81	427.83	423.98	421.52	510.09	421.52	511.28	511.28	511.28
P_{20}	439.41	437.95	434.81	427.83	423.98	421.52	510.09	421.52	511.28	511.28	511.28
P_{21}	439.45	437.88	435.89	433.86	435.20	433.52	433.52	520.89	523.28	523.28	523.28
P_{22}	439.45	437.88	435.89	433.86	435.20	433.52	433.52	519.04	523.28	523.28	523.28
P_{23}	439.77	438.34	436.53	434.67	436.41	433.52	433.52	519.54	523.28	523.28	523.28
P_{24}	439.77	438.34	436.53	434.67	436.41	433.52	433.52	520.49	523.28	523.28	523.28
P_{25}	440.11	438.26	435.83	433.52	433.75	433.52	433.52	433.52	523.27	523.28	523.28
P_{26}	440.11	438.26	435.83	433.52	433.75	433.52	433.52	522.31	523.27	523.28	523.28
P_{27}	28.99	21.07	15.74	12.61	11.05	10.00	10.00	10.00	10.00	10.00	10.00
P_{28}	28.99	21.07	15.74	12.61	11.05	10.00	10.00	10.00	10.00	10.00	10.00
P_{29}	28.99	21.07	15.74	12.61	11.05	10.00	10.00	10.00	10.00	10.00	10.00
P_{30}	97.00	97.00	97.00	97.00	97.00	97.00	87.80	89.13	88.81	88.52	87.85
P_{31}	172.33	174.89	179.86	189.41	190.00	190.00	190.00	162.90	190.00	190.00	190.00
P_{32}	172.33	174.89	179.86	189.41	190.00	159.76	190.00	188.58	188.99	190.00	190.00
P_{33}	172.33	174.89	179.86	189.41	190.00	190.00	190.00	190.00	190.00	190.00	190.00
P_{34}	200.00	200.00	200.00	200.00	200.00	200.00	200.00	184.60	164.80	168.59	165.27
P_{35}	200.00	200.00	200.00	200.00	200.00	200.00	200.00	197.00	197.25	167.21	165.91
P_{36}	200.00	200.00	200.00	200.00	200.00	200.00	200.00	195.34	199.53	196.70	166.97
P_{37}	100.84	103.36	107.83	110.00	110.00	110.00	110.00	110.00	109.11	109.70	110.00
P_{38}	100.84	103.36	107.83	110.00	110.00	110.00	110.00	108.07	109.40	89.42	110.00
P_{39}	100.84	103.36	107.83	110.00	110.00	110.00	110.00	96.74	109.28	109.42	110.00

(Continued on following page)

TABLE 10 (*Continued*) Best solution values for a 40-unit system with different weights.

w	.0	.1	.2	.3	.4	.5	.6	.7	.8	.9	1.0
P_{40}	439.41	437.95	434.81	427.83	423.98	502.68	510.09	421.52	511.28	511.28	511.28
Fuel cost	129955	129005	127421	125888	125671	125324	124255	123317	121576	121538	121374
Emissions	176682	177765	183489	193425	195440	201876	228740	273383	350724	351084	356390

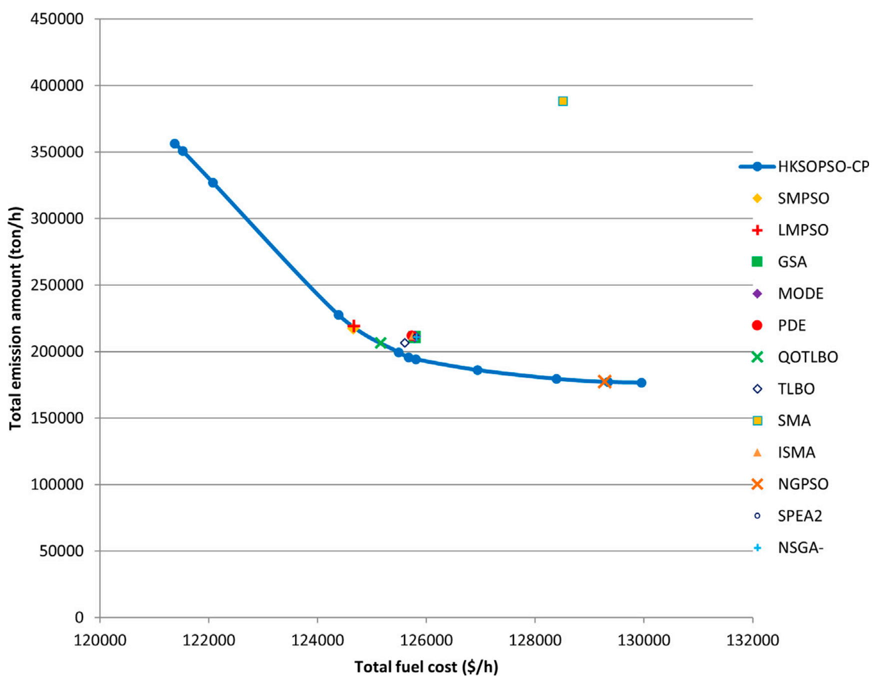

FIGURE 6
Case 3: Solutions generated by the HKSOPSO-CP algorithm, and the best compromise solutions provided by other algorithms.

together, it is clear that HKSOPSO-CP achieves a better cost and overall a better compromise solution compared to NPGA (Horn et al., 1994), MOPSO (Hazra and Sinha, 2011), and NSGA (Qu et al., 2016). Other algorithms, namely MOEA/D (Abido, 2009), SPEA (Basu and Basu, 2011), MBFA (Hota et al., 2010), FSBF (Panigrahi et al., 2010), and NGPSO (Zou et al., 2017), also successfully implemented a cost-matching and compromise solution.

Table 6 presents the results of metaheuristic algorithms in terms of minimization of costs and emissions in case 1. In the case of cost minimization, HKSOPSO-CP and SOA (Dhiman and Kumar, 2018) achieve the same optimal result of 605.8791 ($/h), which outperforms all the other algorithms, particularly MOEA/D (Abido, 2009), SPEA (Basu and Basu, 2011), MBFA (Hota et al., 2010), FSBF (Panigrahi et al., 2010), and NGPSO (Zou et al., 2017), which all achieve comparable compromise solutions. Compared with the second-best result, namely that of KSO, HKSOPSO-CP reduces fuel costs by .017 ($/h). In the case of minimization of emissions, HKSOPSO-CP achieves the optimal result among all the algorithms. These results show that HKSOPSO-CP produces a solution equivalent to or better solution than that provided by existing technology. Therefore, the HKSOPSO-CP algorithm can be considered to exhibit excellent performance in solving low-dimensionality problems.

4.2.2 Case 2 comparison

In case 2, an optimization algorithm such as HKSOPSO-CP is applied in the same way to a ten-unit system with a load demand of 2,000 MW, and its results are compared to those of other optimization algorithms handling ten generators, considering the valve point effect.

Table 7 displays the results of the best solution for case 2 with a range of values for weight factor *w* between 0 and 1 and a step size of .1. As shown in this table, when the only objective function is to minimize fuel costs, HKSOPSO-CP achieves an optimal value of 11,497 ($/h). When only emissions are minimized, HKSOPSO-CP achieves an optimal value of 3,932 (ton/h).

In case 2, the HKSOPSO-CP algorithm is applied to obtain the common optimal values of fuel cost *C* and pollution emissions *E*. Figure 5 depicts the Pareto optimal values and also plots the optimal solutions provided by other algorithms. Taking the graphs and tables together, it is clear that HKSOPSO-CP achieves a better cost and overall a better compromise solution compared to FPA (Abdelaziz et al., 2016b), NSGA (Qu et al., 2016), and SPEA2 (Basu and Basu,

TABLE 11 Case 3 comparison between different algorithms optimizing for fuel cost or emission rate.

Algorithm	Emission minimization (w = 0)		Fuel cost minimization (w = 1)	
	Fuel cost ($/h)	Emissions (ton/h)	Fuel cost ($/h)	Emissions ($/h)
IABC	129995	176682	121415	356422
IABC-LS	129995	176682	121413	359901
HPSOGSA	129997	176684	121413	360228
MBFA	129995	176682	121416	356424
PSOGSA	129987	176678	121461	358155
MODE	129956	176683	121837	374791
DE-HS	129994	176682	121415	356433
MA0-PSO	129995	176682	121413	359902
SOA	130312	179119	124704	361448
TSA	131171	189754.8	124854	382458
ISMA	129949	176683	121547	359501
SMA	129943	176687	121622	362707
KSO	129955	176682	121376	356336
HKSOPSO-CP	129955	176682	121374	356390

2011). Other algorithms, namely BSA (Modiri-Delshad and Abd Rahim, 2015), OGHS (Singh and Dhillon, 2016), NGPSO (Zou et al., 2017), SMA (Hassan et al., 2021), and ISMA (Hassan et al., 2021), also successfully implemented a cost-matching and compromise solution.

Table 8 and Table 9 present the results of metaheuristic algorithms in terms of minimization of costs and emissions in case 2. In the case of cost minimization, HKSOPSO-CP achieves the best optimal result of 111496.97 ($/h) compared with other algorithms, particularly SOA (Dhiman and Kumar, 2018), TSA (Kaur et al., 2020), OGHS (Singh and Dhillon, 2016), SMA (Hassan et al., 2021), and ISMA (Hassan et al., 2021), which all achieve comparable compromise solutions. Compared with the second-best result, namely that of KSO, HKSOPSO-CP reduces fuel costs by .3 ($/h). In the case of minimization of emissions, HKSOPSO-CP achieves the optimal result among all the algorithms. This shows that HKSOPSO-CP generates comparable or better quality solutions as compared to existing techniques in case 2.

4.2.3 Case 3 comparison

A better illustration of the effectiveness of the HKSOPSO-CP algorithm is to implement it for case 3. Case 3 can be used to study the applicability of the proposed methods in the case of a 40-unit system with a load demand of 10,500 MW.

Table 10 displays the results of the best solution for case 3 with a range of values for weight factor w between 0 and 1 and a step size of .1. As shown in this table, when the only objective function is to minimize fuel costs, HKSOPSO-CP achieves an optimal value of 176682 (ton/h). When only emissions are minimized, HKSOPSO-CP achieves an optimal value of 121374($/h).

In case 3, the HKSOPSO-CP algorithm is applied to obtain the common optimal values of fuel cost C and pollution emissions E. Figure 6 depicts the Pareto optimal values and also plots the optimal solutions provided by other algorithms. Taking the graphs and tables together, it is clear that HKSOPSO-CP achieves a better cost and overall a better compromise solution compared to NSGA (Modiri-Delshad and Abd Rahim, 2015), GSA (Özyön et al., 2015), and NSPSO (Zou et al., 2017). Other algorithms, namely QOTLBO (Roy and Bhui, 2013), LMPSO (Jadoun et al., 2015), SMA (Hassan et al., 2021), and ISMA (Hassan et al., 2021), also successfully implemented a cost-matching and compromise solution.

Table 11 presents the results of metaheuristic algorithms in terms of minimization of costs and emissions in case 3. In the case of cost minimization, HKSOPSO-CP achieves the best optimal result of 1,21,374 ($/h) in comparison to all the other algorithms, particularly LMPSO (Jadoun et al., 2015), SOA (Dhiman and Kumar, 2018), TSA (Kaur et al., 2020), SMA (Hassan et al., 2021), and ISMA (Hassan et al., 2021), which all achieve comparable compromise solutions. Compared with the second-best result, namely that of KSO, HKSOPSO-CP reduces fuel costs by 2.0 ($/h). In the case of minimization of emissions, HKSOPSO-CP achieves the optimal result among all the algorithms. The results show that HKSOPSO-CP produces a solution that is equivalent to or better than those provided by existing technology in case 3.

4.2.4 Case 4 comparison

In case 4, HKSOPSO-CP is applied, along with several other optimization algorithms for comparison, to a 110-unit system with a load demand of 15,000 MW, with a valve point effect and no transmission loss.

Table 12 presents the results of metaheuristic algorithms in terms of minimization of costs and emissions in case 4. In the case of cost minimization, HKSOPSO-CP achieves the best result of 198526.78 ($/h) when compared with all other algorithms, particularly SSA and TSA, which achieve comparable solutions on the benchmark function test. Compared with the second-best

TABLE 12 Case 4 comparison between different algorithms optimizing for fuel cost.

Algorithm	SSA	TSA	SMA	ISMA	PSO	KSO	HKSOPSO-CP
P_1	2.622565	4.2086	2.6226	2.4005	5.4292	2.7931	2.7600
P_2	6.319552	2.4000	2.4000	2.4022	5.9845	2.4396	2.4190
P_3	2.580771	6.2549	2.4347	5.0556	5.2657	2.4585	2.4378
P_4	7.980262	3.6530	2.9926	2.6290	5.2151	2.4999	2.4629
P_5	2.742143	6.8452	2.4000	5.1486	4.9833	2.5295	2.5482
P_6	6.97131	4.0000	4.0000	4.0156	8.7832	4.0000	4.0000
P_7	8.421912	4.0000	4.0000	4.0000	15.7124	4.0000	4.0000
P_8	5.671668	6.6529	4.0000	4.0000	12.2462	4.0000	4.0000
P_9	17.12558	4.2596	4.0000	4.0000	12.6635	4.0000	4.0000
P_{10}	75.08748	17.1692	31.2006	31.0802	46.4525	59.5908	65.7108
P_{11}	73.64287	71.9809	74.7964	37.1873	63.9151	57.3409	62.4228
P_{12}	43.16686	49.8780	62.7589	68.8219	54.1720	63.1323	38.2387
P_{13}	39.80309	56.4164	54.1428	36.4421	23.1388	68.5117	56.6228
P_{14}	53.33065	40.8009	25.0714	25.0000	69.9086	26.3418	26.0582
P_{15}	85.65221	25.0000	25.0060	25.1327	63.1699	34.2506	25.1015
P_{16}	61.21406	26.7609	25.0047	25.1051	72.7267	26.6554	25.0842
P_{17}	159.6986	154.0009	155.0000	154.9465	103.6746	153.2279	154.0870
P_{18}	61.77792	154.5085	154.9952	154.9773	147.5522	152.1607	155.0000
P_{19}	98.76287	60.1401	154.9864	155.0000	134.0495	148.8504	153.8775
P_{20}	164.0439	155.0000	102.5721	154.9998	135.3134	148.9845	152.7656
P_{21}	179.6693	68.9000	68.9020	68.9008	84.5990	69.3850	69.1493
P_{22}	204.2383	69.6055	68.9013	68.9273	73.9599	70.1787	69.4531
P_{23}	118.2412	74.1791	68.9000	68.9061	68.9000	69.7400	69.2334
P_{24}	367.2122	349.4898	349.9845	349.9944	332.0594	344.6627	348.9680
P_{25}	170.9869	394.8134	399.9994	399.9994	372.7378	399.1524	400.0000
P_{26}	375.7379	400.0000	400.0000	399.9996	380.5251	395.7907	398.8397
P_{27}	506.9935	460.2447	499.9935	499.9994	377.1125	467.5401	476.0545
P_{28}	560.0383	500.0000	499.7582	500.0000	465.3444	470.0665	458.6294
P_{29}	154.2269	200.0000	200.0000	200.0000	141.5889	189.1772	198.9552
P_{30}	81.24869	25.3328	99.9998	65.2420	76.4075	99.2711	100.0000
P_{31}	28.46584	13.7067	10.0020	10.0000	21.3520	10.6923	10.2708
P_{32}	4.17562	5.0000	12.3444	9.8453	16.3002	18.0406	19.4716
P_{33}	25.8594	20.8379	65.5620	43.5313	32.8140	68.9326	74.2058
P_{34}	182.6466	240.3992	249.9988	247.1876	224.3301	241.9766	247.1635
P_{35}	264.1791	360.0000	359.9986	359.9998	313.3566	346.9819	348.0956
P_{36}	239.4743	400.0000	400.0000	399.9987	336.1917	376.1370	388.2775
P_{37}	14.25777	10.0000	37.6758	29.2966	30.3013	38.1247	38.1490
P_{38}	54.35035	67.7939	56.8364	22.7896	46.7252	66.4007	66.4558
P_{39}	45.80205	49.3139	69.8013	56.9889	70.7727	84.8536	97.5929

(Continued on following page)

TABLE 12 (*Continued*) Case 4 comparison between different algorithms optimizing for fuel cost.

Algorithm	SSA	TSA	SMA	ISMA	PSO	KSO	HKSOPSO-CP
P_{40}	60.68091	35.2239	120.0000	73.8338	91.5161	96.3301	117.5323
P_{41}	89.27466	142.8969	136.6560	152.8516	127.1847	136.5202	170.6824
P_{42}	228.9135	206.5694	199.7630	216.2521	171.7525	196.9808	203.9003
P_{43}	317.3063	440.0000	440.0000	440.0000	439.5246	440.0000	439.4754
P_{44}	627.9953	560.0000	560.0000	559.8904	539.2588	543.0548	559.2631
P_{45}	605.9089	650.7611	659.9800	659.9003	659.6707	658.2462	657.8982
P_{46}	326.3605	643.2216	627.7291	665.0369	699.1032	589.6224	618.4499
P_{47}	13.13005	6.3869	5.4000	5.4005	8.8969	6.0343	5.5034
P_{48}	10.21097	11.5386	5.4000	5.4014	15.2255	5.4330	5.6497
P_{49}	19.87718	9.2598	8.4000	8.4024	16.4760	8.5327	8.5030
P_{50}	19.64045	15.4321	8.4037	8.4000	17.3728	9.2383	8.4494
P_{51}	43.54884	9.1478	29.3681	9.1439	20.7017	9.1278	8.6124
P_{52}	28.27613	17.0171	12.0003	12.0000	16.1034	12.0000	12.0000
P_{53}	27.05809	12.0000	12.0000	12.0016	18.0873	12.0000	12.0000
P_{54}	55.34711	15.7346	12.0000	12.0000	12.0000	12.0000	12.0000
P_{55}	38.63337	13.7635	12.0000	12.0259	12.0000	12.0000	12.0000
P_{56}	73.37852	67.8326	50.6110	57.8652	66.1334	60.2415	26.7997
P_{57}	77.61656	38.5086	48.5226	31.4578	90.1559	75.2052	54.5181
P_{58}	53.23581	47.1231	78.0355	35.7310	66.2796	53.9123	42.5551
P_{59}	52.87958	54.9464	43.8139	68.2357	55.1443	73.4760	50.0174
P_{60}	127.236	49.9604	45.0000	45.0000	69.9769	45.0850	45.5746
P_{61}	81.50426	48.0535	45.0018	45.6812	46.7225	45.8513	45.9457
P_{62}	85.56415	52.5856	45.0000	45.0000	80.2357	46.3309	45.6714
P_{63}	89.97434	175.9284	172.4597	171.6163	124.0940	164.2997	176.8378
P_{64}	119.4536	139.5947	183.8930	184.9205	131.7357	177.2498	167.4040
P_{65}	120.0246	167.5547	169.6174	140.2055	102.7506	171.0727	150.9309
P_{66}	125.3839	172.3518	183.8618	178.9734	95.3875	165.8698	166.7102
P_{67}	148.0305	75.2563	70.0000	70.0000	91.5468	71.0786	70.4225
P_{68}	137.279	70.0000	70.0000	70.0003	71.1494	70.9655	70.1265
P_{69}	125.5039	72.2721	70.0386	70.0000	70.0014	70.9380	70.2708
P_{70}	215.2831	360.0000	341.4428	360.0000	357.0100	306.6718	321.9450
P_{71}	410.9247	356.0329	400.0000	399.9998	399.7669	394.9507	400.0000
P_{72}	291.4381	400.0000	400.0000	399.9935	400.0000	382.6996	397.9505
P_{73}	253.5676	189.8934	68.5842	123.9319	205.3575	200.5920	188.3936
P_{74}	246.3212	154.8527	152.5601	218.2584	207.1537	187.0232	206.5440
P_{75}	92.91994	88.5956	90.0000	88.5949	66.3923	90.0000	88.6105
P_{76}	27.14028	18.2314	50.0000	49.9246	18.6559	50.0000	49.4185
P_{77}	360.4447	444.2485	217.3226	213.4715	398.0960	215.4140	235.5136
P_{78}	338.8554	521.2429	475.6029	409.1935	470.3499	365.0429	292.1456

(Continued on following page)

TABLE 12 (*Continued*) Case 4 comparison between different algorithms optimizing for fuel cost.

Algorithm	SSA	TSA	SMA	ISMA	PSO	KSO	HKSOPSO-CP
P_{79}	109.2578	148.8751	164.0484	118.1070	131.3772	152.7913	175.0241
P_{80}	51.46387	93.4857	45.0855	59.6734	120.0000	97.2393	106.4033
P_{81}	12.65586	10.7962	10.0000	10.0001	27.0205	10.5244	10.2743
P_{82}	29.14222	18.9508	24.3424	14.1862	28.8276	34.5700	21.2688
P_{83}	62.85363	20.2725	35.5953	20.0866	65.4708	56.8222	62.1335
P_{84}	180.5718	193.7249	93.7117	190.3161	108.9955	166.2470	170.7884
P_{85}	275.5625	325.0000	317.0912	303.3981	252.2418	266.8946	318.8481
P_{86}	442.9358	440.0000	439.8854	439.2509	322.0197	374.6612	430.4750
P_{87}	15.63212	18.5699	34.7954	15.6084	24.2937	32.4656	29.2856
P_{88}	31.43107	37.1090	21.6903	38.7410	41.2079	46.8143	37.4321
P_{89}	84.46915	59.7780	95.8253	87.1381	85.4221	38.9726	68.7526
P_{90}	167.388	74.5533	98.8315	185.9027	155.8520	128.0841	112.2484
P_{91}	86.82912	52.2730	61.0916	91.4037	92.1016	116.9602	73.9098
P_{92}	80.75758	42.7131	73.3633	58.7979	80.5661	85.0206	93.1928
P_{93}	298.0457	440.0000	439.9993	440.0000	408.8525	434.8370	437.1723
P_{94}	353.1111	490.0781	500.0000	499.9633	484.6226	490.0770	491.2568
P_{95}	585.3382	600.0000	599.9999	599.7575	599.8339	600.0000	598.2580
P_{96}	406.2559	496.6494	510.9935	485.6981	565.4470	530.2636	426.3165
P_{97}	12.5741	6.8992	3.6289	3.6022	10.6764	3.6437	3.7794
P_{98}	6.362006	5.6272	3.6002	3.6046	10.9636	3.7086	3.8711
P_{99}	5.104551	4.4000	4.4000	4.6846	10.5587	4.4129	4.5395
P_{100}	9.16614	4.4000	4.4002	4.4517	12.4563	4.4209	4.6211
P_{101}	59.38305	13.1696	11.3826	10.7957	26.3353	15.0028	10.0499
P_{102}	32.9138	10.3038	12.7267	20.2634	32.5020	14.7123	10.6984
P_{103}	40.39806	30.1760	20.0002	20.0002	20.0000	24.4985	20.1270
P_{104}	54.45426	21.0474	20.0000	20.0149	20.0000	20.3921	20.4855
P_{105}	82.65469	41.3085	40.0000	40.0048	42.4803	40.0000	40.0000
P_{106}	141.2189	40.0000	40.0000	40.0000	40.0081	40.0000	40.0000
P_{107}	139.9624	50.0000	50.0002	50.0005	50.0000	50.0000	50.0000
P_{108}	156.2254	30.0000	30.0000	30.0000	43.5982	30.0000	30.0000
P_{109}	245.3367	40.0000	40.0004	40.0000	58.5707	40.0000	40.0000
P_{110}	20.00183	22.3258	20.0000	20.0011	23.1355	20.0000	20.0000
V(MV)	.0000	5.22E−02	1.25E−05	2.33E−06	1.42E−01	.0000	.0000
Fuel cost ($/h)	227057.8	200832.10	198618.51	198565.94	204974.80	199147.53	198526.78

result, namely that of ISMA (Hassan et al., 2021), HKSOPSO-CP reduces fuel costs by 39.12 ($/h). These results show that HKSOPSO-CP obtains solutions that are comparable to or better than those provided by existing techniques in case 4. Therefore, the HKSOPSO-CP algorithm can be considered to provide improved performance in solving high-dimensionality EED problems.

Overall, SOA offers nearly the same performance as HKSOPSO when dealing with low-dimensionality problems, but it is less effective when dealing with high-dimensionality problems. HKSOPSO-CP provides the optimal results in solving problems of different dimensionalities, which indicates that this method offers better performance and versatility than other algorithms; for this reason, it can be applied to more fields, such as hybrid motion models (Wu et al., 2022), structured sparsity optimization (Zhang et al., 2022b), recommender systems (Li et al., 2014; Li et al., 2017), image-to-image translation (Zhang et al., 2022c), human activity recognition (Qiu et al., 2022), dynamic module detection (Li et al., 2021), location-based services (Wu et al., 2020b; Wu et al., 2021b), smart contract vulnerability detection (Zhang et al., 2022d), power flow optimization (Cao et al., 2022b), image denoising (Zhang et al., 2020), and medical data processing (Guo et al., 2022).

5 Conclusion

A novel hybrid approach, in the form of the HKSOPSO-CP optimization algorithm, is presented in this paper and is successfully applied to solve the EED problem in power systems, accounting for the valve point effect. The HKSOPSO-CP method is applied in four real-world cases, involving 6, 10, 40, and 110 generating units with various power system constraints. It has been demonstrated that the proposed algorithm is capable of providing better solutions than conventional KSO and other algorithms used as points of comparison. The results also show that the proposed algorithm is robust and flexible. In the future, this method can be used to solve other complex optimization problems. Moreover, different kernel functions could be used to solve some optimization problems to achieve better results; however, to enable this, it will be necessary to focus on computational tricks for kernel mapping. Meanwhile, research will focus on applying the HKSOPSO-CP approach and its variants to other real-world problems, such as medical data classification and complex multi-objective problems in engineering.

Data availability statement

The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding authors.

Author contributions

RD: Writing—original draft, writing—review and editing, software, visualization, investigation. LM: Writing—Original draft, writing—Review and editing, software, visualization, investigation. HC: Conceptualization, methodology, formal analysis, investigation, writing—Review and editing, funding acquisition, supervision. AH: Writing—original draft, writing—review and editing, software, visualization, investigation. GL: Conceptualization, methodology, formal analysis, investigation, writing—review and editing, funding acquisition, supervision.

Funding

This research was supported by the Science and Technology Development Project of Jilin Province, China (YDZJ202201ZYTS555), the Science and Technology Research Project of the Education Department of Jilin Province, China (JJKH20220244KJ), and the Zhejiang Provincial Philosophy and Social Sciences Planning Project (23NDJC393YBM).

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

References

Abdelaziz, A., Ali, E., and Abd-Elazim, S. (2016). Combined economic and emission dispatch solution using flower pollination algorithm. *Int. J. Electr. Power Energy Syst.* 80, 264–274. doi:10.1016/j.ijepes.2015.11.093

Abdelaziz, A., Ali, E., and Abd-Elazim, S. (2016). Implementation of flower pollination algorithm for solving economic load dispatch and combined economic emission dispatch problems in power systems. *Energy* 101, 506–518. doi:10.1016/j.energy.2016.02.041

Abido, M. A. (2009). "Multiobjective evolutionary algorithms for electric power dispatch problem," in *Computational intelligence* (Springer), 47–82.

Abualigah, L., Elaziz, M. A., Sumari, P., Geem, Z. W., and Gandomi, A. H. (2021). Reptile search algorithm (RSA): A nature-inspired meta-heuristic optimizer. *Expert Syst. Appl.* 191, 116158. doi:10.1016/j.eswa.2021.116158

Ahmadianfar, I., Heidari, A. A., Gandomi, A. H., Chu, X., and Chen, H. (2021). RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Syst. Appl.* 181, 115079. doi:10.1016/j.eswa.2021.115079

Ahmadianfar, I., Heidari, A. A., Noshadian, S., Chen, H., and Gandomi, A. H. (2022). INFO: An efficient optimization algorithm based on weighted mean of vectors. *Expert Syst. Appl.* 195, 116516. doi:10.1016/j.eswa.2022.116516

Basu, M., and Basu, M. (2011). Economic environmental dispatch using multi-objective differential evolution. *Appl. Soft Comput.* 11 (2), 28452845–28532853. doi:10.1016/j.asoc.2010.11.014

Cao, B., Gu, Y., Lv, Z., Yang, S., Zhao, J., and Li, Y. (2020). RFID reader anticollision based on distributed parallel particle swarm optimization. *IEEE Internet Things J.* 8 (5), 3099–3107. doi:10.1109/jiot.2020.3033473

Cao, B., Yan, Y., Wang, Y., Liu, X., Lin, J. C. W., Sangaiah, A. K., et al. (2022). A multiobjective intelligent decision-making method for multistage placement of PMU in power grid enterprises. *IEEE Trans. Industrial Inf.*, 1–9. doi:10.1109/tii.2022.3215787

Cao, X., Wang, J., and Zeng, B. (2022). A study on the strong duality of second-order conic relaxation of AC optimal power flow in radial networks. *IEEE Trans. Power Syst.* 37 (1), 443–455. doi:10.1109/tpwrs.2021.3087639

Chen, H.-L., Wang, G., Cai, Z. N., Liu, W. B., and Wang, S. J. (2016). An efficient hybrid kernel extreme learning machine approach for early diagnosis of Parkinson's disease. *Neurocomputing* 184, 131–144. doi:10.1016/j.neucom.2015.07.138

Chen, Y. (2022). Research on collaborative innovation of key common technologies in new energy vehicle industry based on digital twin technology. *Energy Rep.* 8, 15399–15407. doi:10.1016/j.egyr.2022.11.120

Cima, J. A., Matheson, A. L., and Ross, W. T. (2006). *The Cauchy transform*. American Mathematical Soc.

de Athayde Costa e Silva, M., Klein, C. E., Mariani, V. C., and dos Santos Coelho, L. (2013). Multiobjective scatter search approach with new combination scheme applied to solve environmental/economic dispatch problem. *Energy* 53, 14–21. doi:10.1016/j.energy.2013.02.045

Deng, W., Ni, H., Liu, Y., Chen, H., and Zhao, H. (2022). An adaptive differential evolution algorithm based on belief space and generalized opposition-based learning for resource allocation. *Appl. Soft Comput.* 127, 109419. doi:10.1016/j.asoc.2022.109419

Deng, W., Xu, J., Gao, X. Z., and Zhao, H. (2022). An enhanced MSIQDE algorithm with novel multiple strategies for global optimization problems. *IEEE Trans. Syst. Man, Cybern. Syst.* 52 (3), 1578–1587. doi:10.1109/tsmc.2020.3030792

Deng, W., Xu, J., Zhao, H., and Song, Y. (2020). A novel gate resource allocation method using improved PSO-based QEA. *IEEE Trans. Intelligent Transp. Syst.* 23, 1737–1745. doi:10.1109/TITS.2020.3025796

Deng, W., Zhang, L., Zhou, X., Zhou, Y., Sun, Y., Zhu, W., et al. (2022). Multi-strategy particle swarm and ant colony hybrid optimization for airport taxiway planning problem. *Inf. Sci.* 612, 576–593. doi:10.1016/j.ins.2022.08.115

Deng, W., Zhang, X., Zhou, Y., Liu, Y., Zhou, X., Chen, H., et al. (2022). An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems. *Inf. Sci.* 585, 441–453. doi:10.1016/j.ins.2021.11.052

Dhiman, G., and Kumar, V. (2018). Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowledge-Based Syst.* 165, 169–196. doi:10.1016/j.knosys.2018.11.024

Dong, R., Chen, H., Heidari, A. A., Turabieh, H., Mafarja, M., and Wang, S. (2021). Boosted kernel search: Framework, analysis and case studies on the economic emission dispatch problem. *Knowledge-Based Syst.* 233, 107529. doi:10.1016/j.knosys.2021.107529

Dong, R., and Wang, S. (2020). New optimization algorithm inspired by kernel tricks for the economic emission dispatch problem with valve point. *IEEE Access* 8, 16584–16594. doi:10.1109/access.2020.2965725

El-Keib, A. A., and Ding, H. (1994). Environmentally constrained economic dispatch using linear programming. *Electr. Power Syst. Res. - ELEC POWER SYST RES* 29, 155–159. doi:10.1016/0378-7796(94)90010-8

Gao, D., Wang, G.-G., and Pedrycz, W. (2020). Solving fuzzy job-shop scheduling problem using DE algorithm improved by a selection mechanism. *IEEE Trans. Fuzzy Syst.* 28 (12), 3265–3275. doi:10.1109/tfuzz.2020.3003506

Ge, L., Du, T., Li, Y., Yan, J., and Rafiq, M. U. (2022). Virtual collection for distributed photovoltaic data: Challenges, methodologies, and applications. *Energies* 15 (23), 8783. doi:10.3390/en15238783

Ge, L., Li, Y., Li, Y., Yan, J., and Sun, Y. (2022). Smart distribution network situation awareness for high-quality operation and maintenance: A brief Review. *Energies* 15 (3), 828. doi:10.3390/en15030828

Goudarzi, A., Li, Y., and Xiang, J. (2019). A hybrid non-linear time-varying double-weighted particle swarm optimization for solving non-convex combined environmental economic dispatch problem. *Appl. Soft Comput.* 86, 105894. doi:10.1016/j.asoc.2019.105894

Guo, K., Chen, T., Ren, S., Li, N., Hu, M., and Kang, J. (2022). Federated learning empowered real-time medical data processing method for smart healthcare. *IEEE/ACM Trans. Comput. Biol. Bioinforma.*, 1–12. doi:10.1109/tcbb.2022.3185395

Han, X., Han, Y., Chen, Q., Li, J., Sang, H., Liu, Y., et al. (2021). Distributed flow shop scheduling with sequence-dependent setup times using an improved iterated greedy algorithm. *Complex Syst. Model. Simul.* 1 (3), 198–217. doi:10.23919/csms.2021.0018

Hassan, M. H., Kamel, S., Abualigah, L., and Eid, A. (2021). Development and application of slime mould algorithm for optimal economic emission dispatch. *Expert Syst. Appl.* 182, 115205. doi:10.1016/j.eswa.2021.115205

Hayyolalam, V., and Pourhaji Kazem, A. A. (2019). Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems ☆. *Eng. Appl. Artif. Intell.* 87, 103249. doi:10.1016/j.engappai.2019.103249

Hazra, J., and Sinha, A. (2011). A multi-objective optimal power flow using particle swarm optimization. *Eur. Trans. Electr. power* 21 (1), 1028–1045. doi:10.1002/etep.494

He, Z., Yen, G. G., and Ding, J. (2020). Knee-based decision making and visualization in many-objective optimization. *IEEE Trans. Evol. Comput.* 25 (2), 292–306. doi:10.1109/tevc.2020.3027620

Heidari, A. A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., and Chen, H. (2019). Harris hawks optimization: Algorithm and applications. *Future Generation Comput. Systems-the Int. J. Escience* 97, 849–872. doi:10.1016/j.future.2019.02.028

Horn, J., Nafpliotis, N., and Goldberg, D. E. (1994). "A niched Pareto genetic algorithm for multiobjective optimization," in Proceedings of the first IEEE conference on evolutionary computation. IEEE world congress on computational intelligence, Orlando, FL, USA, 27-29 June 1994 (IEEE).

Hosny, M., Kamel, S., Abualigah, L., and Eid, A. (2021). Development and application of slime mould algorithm for optimal economic emission dispatch. *Expert Syst. Appl.* 182, 115205. doi:10.1016/j.eswa.2021.115205

Hota, P., Barisal, A., and Chakrabarti, R. (2010). Economic emission load dispatch through fuzzy based bacterial foraging algorithm. *Int. J. Electr. Power & Energy Syst.* 32, 794–803. doi:10.1016/j.ijepes.2010.01.016

Hu, J., Gui, W., Heidari, A. A., Cai, Z., Liang, G., Chen, H., et al. (2022). Dispersed foraging slime mould algorithm: Continuous and binary variants for global optimization and wrapper-based feature selection. *Knowledge-Based Syst.* 237, 107761. doi:10.1016/j.knosys.2021.107761

Hua, Y., Liu, Q., Hao, K., and Jin, Y. (2021). A survey of evolutionary algorithms for multi-objective optimization problems with irregular Pareto fronts. *IEEE/CAA J. Automatica Sinica* 8 (2), 303–318. doi:10.1109/jas.2021.1003817

Hussien, A. G., Heidari, A. A., Ye, X., Liang, G., Chen, H., and Pan, Z. (2022). Boosting whale optimization with evolution strategy and Gaussian random walks: An image segmentation method. *Eng. Comput.* doi:10.1007/s00366-021-01542-0

Jadhav, H. T., Raj, S., and Roy, R. (2013). Solution to economic emission load dispatch problem using modified artificial bee colony algorithm. 1–6.

Jadoun, V. K., Gupta, N., Niazi, K. R., and Swarnkar, A. (2015). Modulated particle swarm optimization for economic emission dispatch. *Int. J. Electr. Power & Energy Syst.* 73, 80–88. doi:10.1016/j.ijepes.2015.04.004

Kaur, S., Dhiman, G., and Sangal, A. (2020). Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization. *Eng. Appl. Artif. Intell.* 90, 103541. doi:10.1016/j.engappai.2020.103541

Kennedy, J., and Eberhart, R. (1995). "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks* (IEEE).

Kumar, R., Sadu, A., and Panda, S. (2012). A novel multiobjective directed bee colony optimization algorithm for multiobjective emission constrained economic power dispatch. *Int. J. Electr. Power & Energy Syst.* 43, 1241–1250. doi:10.1016/j.ijepes.2012.06.011

Li, D., Zhang, S., and Ma, X. (2021). Dynamic module detection in temporal attributed networks of cancers. *IEEE/ACM Trans. Comput. Biol. Bioinforma.* 19, 2219–2230. doi:10.1109/TCBB.2021.3069441

Li, J.-Y., Zhan, Z. H., Wang, C., Jin, H., and Zhang, J. (2020). Boosting data-driven evolutionary algorithm with localized data generation. *IEEE Trans. Evol. Comput.* 24 (5), 923–937. doi:10.1109/tevc.2020.2979740

Li, J., Chen, C., Chen, H., and Tong, C. (2017). Towards context-aware social recommendation via individual trust. *Knowledge-Based Syst.* 127, 58–66. doi:10.1016/j.knosys.2017.02.032

Li, J., Zheng, X. L., Chen, S. T., Song, W. W., and Chen, D. r. (2014). An efficient and reliable approach for quality-of-service-aware service composition. *Inf. Sci.* 269, 238–254. doi:10.1016/j.ins.2013.12.015

Li, S., Chen, H., Wang, M., Heidari, A. A., and Mirjalili, S. (2020). Slime mould algorithm: A new method for stochastic optimization. *Future Gener. Comput. Syst.* 111, 300–323. doi:10.1016/j.future.2020.03.055

Lin, Y., Song, H., Ke, F., Yan, W., Liu, Z., and Cai, F. (2022). Optimal caching scheme in D2D networks with multiple robot helpers. *Comput. Commun.* 181, 132–142. doi:10.1016/j.comcom.2021.09.027

Liu, Y., Heidari, A. A., Cai, Z., Liang, G., Chen, H., Pan, Z., et al. (2022). Simulated annealing-based dynamic step shuffled frog leaping algorithm: Optimal performance design and feature selection. *Neurocomputing* 503, 325–362. doi:10.1016/j.neucom.2022.06.075

Mavrovouniotis, M., and Yang, S. (2011). A memetic ant colony optimization algorithm for the dynamic travelling salesman problem. *Soft Comput.* 15, 1405–1425. doi:10.1007/s00500-010-0680-1

Modiri-Delshad, M., and Abd Rahim, N. (2015). Multi-objective backtracking search algorithm for economic emission dispatch problem. *Appl. Soft Comput.* 40.

Özyön, S., Yasar, C., Durmus, B., and Temurtas, H. (2015). Opposition-based gravitational search algorithm applied to economic power dispatch problems consisting of thermal units with emission constraints. *Turkish J. Electr. Eng. Comput. Sci.* 23, 2278–2288. doi:10.3906/elk-1305-258

Panigrahi, B., Ravikumar Pandi, V., Das, S., and Das, S. (2010). Multiobjective fuzzy dominance based bacterial foraging algorithm to solve economic emission dispatch problem. *Energy* 35, 4761–4770. doi:10.1016/j.energy.2010.09.014

Pierezan, J., and Coelho, L. (2018). Coyote optimization algorithm: A new metaheuristic for global optimization problems. 1–8.

Qiu, S., Zhao, H., Jiang, N., Wang, Z., Liu, L., An, Y., et al. (2022). Multi-sensor information fusion based on machine learning for real applications in human activity recognition: State-of-the-art and research challenges. *Inf. Fusion* 80, 241–265. doi:10.1016/j.inffus.2021.11.006

Qu, B.-Y., Liang, J., Zhu, Y., Wang, Z., and Suganthan, P. (2016). Economic emission dispatch problems with stochastic wind power using summation based multi-objective evolutionary algorithm. *Inf. Sci.* 351, 48–66. doi:10.1016/j.ins.2016.01.081

Roy, P., and Bhui, S. (2013). Multi-objective quasi-oppositional teaching learning based optimization for economic emission load dispatch problem. *Int. J. Electr. Power & Energy Syst.* 53, 937–948. doi:10.1016/j.ijepes.2013.06.015

Sierpina, V. (2013). *Steven PrattSuperFoods rx for PregnancyRefstyled*. Canada: John Wiley & Sons Limited. *Canada.* EXPLORE: The Journal of Science and Healing 9.

Singh, M., and Dhillon, J. S. (2016). Multiobjective thermal power dispatch using opposition-based greedy heuristic search. *Int. J. Electr. Power & Energy Syst.* 82, 339–353. doi:10.1016/j.ijepes.2016.03.016

Song, Y., Cai, X., Zhou, X., Zhang, B., Chen, H., Li, Y., et al. (2023). Dynamic hybrid mechanism-based differential evolution algorithm and its application. *Expert Syst. Appl.* 213, 118834. doi:10.1016/j.eswa.2022.118834

Tu, J., Huiling, C., Mingjing, W., and Amir, H. G. (2021). The colony predation algorithm. *J. Bionic Eng.* 18 (3), 674–710. doi:10.1007/s42235-021-0050

Wu, D., Zhao, H., Song, Y., and Xu, J. (2020). An effective improved co-evolution ant colony optimisation algorithm with multi-strategies and its application. *Int. J. Bio-Inspired Comput.* 16 (3), 158–170. doi:10.1504/ijbic.2020.10033314

Wang, G.-G., Gao, D., and Pedrycz, W. (2022). Solving multi-objective fuzzy job-shop scheduling problem by a hybrid adaptive differential evolution algorithm. *IEEE Trans. Industrial Inf.* 18, 8519–8528. doi:10.1109/tii.2022.3165636

Wang, M., Chen, H., Yang, B., Zhao, X., Hu, L., Cai, Z., et al. (2017). Toward an optimal kernel extreme learning machine using a chaotic moth-flame optimization strategy with applications in medical diagnoses. *Neurocomputing* 267, 69–84. doi:10.1016/j.neucom.2017.04.060

Wang, Y., Wen, X., Gu, B., and Gao, F. (2022). Power scheduling optimization method of wind-hydrogen integrated energy system based on the improved AUKF algorithm. *Mathematics* 10 (22), 4207. doi:10.3390/math10224207

Wu, S.-H., Zhan, Z.-H., and Zhang, J. (2021). Safe: Scale-adaptive fitness evaluation method for expensive optimization problems. *IEEE Trans. Evol. Comput.* 25 (3), 478–491. doi:10.1109/tevc.2021.3051608

Wu, Y., Sheng, H., Zhang, Y., Wang, S., Xiong, Z., and Ke, W. (2022). Hybrid motion model for multiple object tracking in mobile devices. *IEEE Internet Things J.*, 1. doi:10.1109/jiot.2022.3219627

Wu, Z., Li, G., Shen, S., Lian, X., Chen, E., and Xu, G. (2021). Constructing dummy query sequences to protect location privacy and query privacy in location-based services. *World Wide Web* 24 (1), 25–49. doi:10.1007/s11280-020-00830-x

Wu, Z., Wang, R., Li, Q., Lian, X., Xu, G., Chen, E., et al. (2020). A location privacy-preserving system based on query range cover-up or location-based services. *IEEE Trans. Veh. Technol.* 69, 5244–5254. doi:10.1109/tvt.2020.2981633

Xu, Y., Chen, H., Heidari, A. A., Luo, J., Zhang, Q., Zhao, X., et al. (2019). An efficient chaotic mutative moth-flame-inspired optimizer for global optimization tasks. *Expert Syst. Appl.* 129, 135–155. doi:10.1016/j.eswa.2019.03.043

Xue, J.-K., and Shen, B. (2020). A novel swarm intelligence optimization approach: Sparrow search algorithm. *Syst. Sci. Control Eng.* 8, 22–34. doi:10.1080/21642583.2019.1708830

Yang, Y., Chen, H., Heidari, A. A., and Gandomi, A. H. (2021). Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts. *Expert Syst. Appl.* 177, 114864. doi:10.1016/j.eswa.2021.114864

Yaşar, C., and Özyön, S. (2012). Solution to scalarized environmental economic power dispatch problem by using genetic algorithm. *Int. J. Electr. Power & Energy Syst.* 38, 54–62. doi:10.1016/j.ijepes.2011.12.020

Yu, H., Song, J., Chen, C., Heidari, A. A., Liu, J., Chen, H., et al. (2022). Image segmentation of Leaf Spot Diseases on Maize using multi-stage Cauchy-enabled grey wolf algorithm. *Eng. Appl. Artif. Intell.* 109, 104653. doi:10.1016/j.engappai.2021.104653

Zhan, J., Wu, Q. H., Guo, C. X., and Zhou, X. X. (2014). Fast λ-Iteration method for economic dispatch with prohibited operating zones. *IEEE Trans.*29, 990–991. doi:10.1109/tpwrs.2013.2287995

Zhang, K., Wang, Z., Chen, G., Zhang, L., Yang, Y., Yao, C., et al. (2022). Training effective deep reinforcement learning agents for real-time life-cycle production optimization. *J. Petroleum Sci. Eng.* 208, 109766. doi:10.1016/j.petrol.2021.109766

Zhang, L., Wang, J., Wang, W., Jin, Z., Su, Y., and Chen, H. (2022). Smart contract vulnerability detection combined with multi-objective detection. *Comput. Netw.* 217, 109289. doi:10.1016/j.comnet.2022.109289

Zhang, X., Fan, C., Xiao, Z., Zhao, L., Chen, H., and Chang, X. (2022). Random reconstructed unpaired image-to-image translation. *IEEE Trans. Industrial Inf.*, 1. doi:10.1109/TII.2022.3160705

Zhang, X., Zheng, J., Wang, D., Tang, G., Zhou, Z., and Lin, Z. (2022). Structured sparsity optimization with non-convex surrogates of $l_{2,0}$-norm: A unified algorithmic framework. *IEEE Trans. Pattern Analysis Mach. Intell.*, 1–18. doi:10.1109/tpami.2022.3213716

Zhang, X., Zheng, J., Wang, D., and Zhao, L. (2020). Exemplar-based denoising: A unified low-rank recovery framework. *IEEE Trans. Circuits Syst. Video Technol.* 30 (8), 2538–2549. doi:10.1109/tcsvt.2019.2927603

Zhang, Y., Liu, R., Heidari, A. A., Wang, X., Chen, Y., Wang, M., et al. (2021). Towards augmented kernel extreme learning models for bankruptcy prediction: Algorithmic behavior and comprehensive analysis. *Neurocomputing* 430, 185–212. doi:10.1016/j.neucom.2020.10.038

Zheng, W., Liu, X., Ni, X., Yin, L., and Yang, B. (2021). Improving visual reasoning through semantic representation. *IEEE access* 9, 91476–91486. doi:10.1109/access.2021.3074937

Zheng, W., Liu, X., and Yin, L. (2021). Sentence representation method based on multi-layer semantic network. *Appl. Sci.* 11 (3), 1316. doi:10.3390/app11031316

Zheng, W., Tian, X., Yang, B., Liu, S., Ding, Y., Tian, J., et al. (2022). A few shot classification methods based on multiscale relational networks. *Appl. Sci.* 12 (8), 4059. doi:10.3390/app12084059

Zheng, W., Yin, L., Chen, X., Ma, Z., Liu, S., and Yang, B. (2021). Knowledge base graph embedding module design for Visual question answering model. *Pattern Recognit.* 120, 108153. doi:10.1016/j.patcog.2021.108153

Zheng, W., Zhou, Y., Liu, S., Tian, J., Yang, B., and Yin, L. (2022). A deep fusion matching network semantic reasoning model. *Appl. Sci.* 12 (7), 3416. doi:10.3390/app12073416

Zou, D., Li, S., Li, Z., and Kong, X. (2017). A new global particle swarm optimization for the economic emission dispatch with or without transmission losses. *Energy Convers. Manag.* 139, 45–70. doi:10.1016/j.enconman.2017.02.035