Mathematics

https://www.aimspress.com/journal/Math

AIMS Mathematics, 9(8): 19911–19942.
DOI: 10.3934/math.2024972
Received: 09 March 2024
Revised: 04 June 2024
Accepted: 12 June 2024
Published: 19 June 2024

Research article

A hybrid framework for mean-CVaR portfolio selection under jump-diffusion processes: Combining cross-entropy method with beluga whale optimization

Guocheng Li[1,2,*], **Pan Zhao**[1], **Minghua Shi**[1,2] **and Gensheng Li**[1,*]

[1] School of Finance and Mathematics, West Anhui University, Lu'an 237012, China
[2] Anhui Provincial Key Laboratory of Philosophy and Social Sciences for Data Intelligence and Rural Revitalization of Dabie Mountains, Lu'an 237012, China

* **Correspondence:** Email: liguocheng@wxc.edu.cn; lgsheng2007@163.com.

Abstract: In this paper, a new hybrid meta-heuristic algorithm called CEBWO (cross-entropy method and beluga whale optimization) is presented to solve the mean-CVaR portfolio optimization problem based on jump-diffusion processes. The proposed CEBWO algorithm combines the advantages of the cross-entropy method and beluga whale optimization algorithm with the help of co-evolution technology to enhance the performance of portfolio selection. The method is evaluated on 29 unconstrained benchmark functions from CEC 2017, where its performance is compared against several state-of-the-art algorithms. The results demonstrate the superiority of the hybrid method in terms of solution quality and convergence speed. Finally, Monte Carlo simulation is employed to generate scenario paths based on the jump-diffusion model. Empirical results further confirm the effectiveness of the hybrid meta-heuristic algorithm for mean-CVaR portfolio selection, highlighting its potential for real-world applications.

Keywords: portfolio selection; conditional value-at-risk; jump-diffusion process; cross-entropy method; beluga whale optimization
Mathematics Subject Classification: 68T20, 91G10

1. Introduction

Markowitz (1952) [1] introduced the mean-variance (MV) model, which has significantly influenced modern investment theory. However, both theoretical research and practical applications have shown that variance has certain limitations and cannot adequately measure risk (Artzner et al. 1999) [2]. As an alternative, value-at-risk (VaR) was proposed to serve as a statistical estimation of risk (Morgan 1996) [3]. VaR represents the maximum expected loss of a financial asset or portfolio under a specified confidence level and time period (Chen 2011) [4]. VaR offers several advantages as a risk measurement method and allows for direct comparison of relative risk across different measurement tools facing various risks (Goh et al. 2012) [5]. Consequently, VaR has become the primary method for measuring financial risks and is widely employed by financial institutions and their business units for making investment decisions (Morgan 1996; Basak and Shapiro 2001, among others) [3, 6]. Nonetheless, VaR lacks sub-additivity, convexity, and consistency as risk measures (Morgan 1996) [3]. To address these limitations, conditional value-at-risk (CVaR), which represents the average level of excess losses beyond VaR, was introduced as a consistent risk measure with sub-additivity and convexity properties (Rockfeller and Uryasev 2000; 2002) [7, 8]. CVaR has gained popularity in portfolio and risk management research. Related articles are Alexander et al. (2006) [9], Zhu et al. (2009) [10], Yau et al. (2011) [11], Hong and Liu (2009) [12], Zhao et al. (2015) [13], and Ferreira and Cardoso (2021) [14].

However, in practical applications, the mean-CVaR portfolio optimization problem is non-smooth due to the non-differentiability of CVaR, rendering classical gradient methods inapplicable. Rockfeller and Uryasev (2002) [8] proposed a new method for solving this problem by introducing an auxiliary variable into linear programming. Fábián (2008) [15] addressed CVaR objectives in two-stage stochastic models based on decomposition frameworks. Hong and Liu (2009) [12] explored estimating the sensitivities of CVaR using Monte Carlo simulation. Liu et al. (2022) [16] investigated calculating CVaR through weighted kernel density estimation and achieved promising results. Abudurexiti et al. (2023) [17] focused on investigating the value at risk (VaR) and conditional value at risk (CVaR) risk measures for portfolios of returns. Specifically, their research considered cases where the underlying distribution of returns belongs to a broader class of normal mean-variance mixture models. In recent years, the development of various heuristic algorithms has provided new avenues for tackling this problem, such as particle swarm optimization (Lu et al. 2013) [18], the fireworks algorithm (Zhang and Liu 2017) [19], a genetic algorithm (Zhai et al. 2018) [20], the gray wolf optimization (Li et al. 2022) [21], the bi-level whale optimization algorithm (Lu et al. 2022) [22], the Runge–Kutta method (Danane et al. 2023) [23], and differential evolution (Song et al. 2023) [24]. These algorithms have shown promise in addressing mean-CVaR portfolio optimization problems.

In recent years, a significant number of swarm-based metaheuristic algorithms have been developed and investigated to solve complex optimization problems. These algorithms include the bat algorithm (Yang and Gandomi 2012) [25], krill herd algorithm (Gandomi and Alavi 2012) [26], grey wolf optimizer (Mirjalili et al. 2014) [27], crow search algorithm (Askarzadeh 2016) [28], whale optimization algorithm (Mirjalili and Lewis 2016) [29], grasshopper optimization algorithm (Saremi et al. 2017) [30], moth search algorithm (Wang 2018) [31], pity beetle algorithm (Kallioras et al. 2018) [32], Harris hawks optimization (Heidari et al. 2019) [33], squirrel search algorithm (Jain, 2019) [34], butterfly optimization algorithm (Arora and Singh 2019) [35], marine predator algorithm (Faramarzi et al. 2020) [36], chimp optimization algorithm (Khishe and Mosavi 2020) [37], slime mould algorithm (Li et al. 2020) [38], golden eagle optimizer (Mohammadi-Balani et al. 2021) [39], red fox optimization (Połap and Woźniak 2021) [40], hunger games search (Yang et al. 2021) [41], Runge Kutta method (Ahmadianfar et al. 2021a) [42], colony predation algorithm (Tu et al. 2021) [43], weighted mean of vectors (Ahmadianfar et al. 2021b) [44], African vultures optimization algorithm (Abdollahzadeh et al. 2021a) [45], artificial gorilla troops optimizer (Abdollahzadeh et al. 2021b) [46], honey badger algorithm (Hashim et al. 2022) [47], artificial hummingbird algorithm

(Zhao et al. 2022) [48], mountain gazelle optimizer (Abdollahzadeh et al. 2022) [49], Liver cancer algorithm (Houssein et al. 2023) [50], rime optimization algorithm (Su et al. 2023) [51], and others [52].

With advancements in meta-heuristic algorithms, it has been recognized that the performance of any single meta-heuristic algorithm is limited and specific to certain optimization problems (Wolpert and Macready, 1997) [53]. To address this issue, hybrid meta-heuristic algorithms have gained attention as a new research area. These algorithms combine two or more different meta-heuristic algorithms to leverage their strengths and overcome limitations. Some well-known hybrid meta-heuristic algorithms include WOASA (Mafarja and Mirjalili, 2017) [54] and HHOBSA (Abdel-Basset et al., 2021) [55] for feature selection, GWO-ABC (Gaidhane and Nigam, 2018) [56] for designing a fractional order PID controller, and HPG-SOS (Farnad et al. 2018) [57] combining GA, PSO, and SOS for continuous optimization problems. ALO-KHO (AkbaiZadeh et al. 2021) [58] is employed for energy management. These hybrid algorithms aim to improve exploration and exploitation capabilities, leading to better performance in solving complex optimization problems. They have been successfully applied in function optimization and engineering design optimization. In this paper, we present a novel hybrid algorithm called the cross-entropy-based beluga whale optimization (CEBWO) algorithm, which is an enhanced version of the beluga whale optimization algorithm proposed by Zhong et al. (2022) [52], to tackle the mean-CVaR portfolio optimization problem. The proposed CEBWO algorithm incorporates the cross-entropy method and Monte Carlo techniques to address the limitations of the original beluga whale optimization algorithm, such as susceptibility to local optima and unbalanced exploration of the search space. By leveraging the power of Monte Carlo sampling, the CEBWO algorithm improves the algorithm's effectiveness in exploring the entire search space and achieving a more balanced development.

The main contributions of this paper can be summarized as follows:

- Integration of BWO and CE: The paper introduces the CEBWO algorithm, a hybrid meta-heuristic approach that combines the cross-entropy (CE) method and beluga whale optimization (BWO) algorithm. The algorithm accelerates the convergence rate of the CE operator by utilizing excellent individuals obtained from the BWO operator.
- Improved exploration and exploitation: By leveraging the strengths of both algorithms through co-evolution, the CEBWO algorithm achieves a better balance between exploration and exploitation. It is evaluated on 29 unconstrained benchmark functions from CEC 2017 and compared against state-of-the-art algorithms. The results demonstrate the algorithm's superiority in terms of solution quality and convergence speed.
- Application to mean-CVaR portfolio optimization: The CEBWO algorithm is specifically designed for solving the mean-CVaR portfolio optimization problem based on jump-diffusion processes. Empirical results obtained through Monte Carlo simulation confirm the effectiveness of the CEBWO algorithm for mean-CVaR portfolio selection, highlighting its potential for real-world applications in portfolio management.

The remainder of this paper is organized as follows. Section 2 introduces the mean CVaR portfolio optimization problem and jump-diffusion model. Section 3 presents a comprehensive description of the proposed CEBWO algorithm. Section 4 evaluates the performance of our hybrid metaheuristic approach by conducting experiments on 30 benchmark functions from CEC 2017. The effectiveness of our method in addressing the mean-CVaR portfolio optimization problem is validated in Section 5. Finally, the conclusions drawn from our study are summarized in Section 6.

2. Preliminaries

2.1. Mean-CVaR portfolio optimization model

In this paper, we address the portfolio selection problem involving n risk assets. We consider a random vector $y = (y_1, y_2, \cdots, y_n)'$, with a density function denoted as $p(y)$, representing the uncertain returns of the n assets. The expected returns of these assets are denoted as $r = (r_1, r_2, \cdots, r_n)'$. A portfolio of the n assets is represented by $x = (x_1, x_2, \cdots, x_n)'$, and its loss function is defined as $f(x, y) = -x'y$. In the classic mean-variance portfolio optimization model, the risk measure is traditionally based on variance. However, in this study, we replace the risk measure with CVaR (conditional value-at-risk). Consequently, we formulate the mean-CVaR portfolio optimization model as follows:

$$\begin{cases} \min\ CVaR(x) \\ s.t.\ x'e = 1, x'r = u, x \in \mathcal{N}, \end{cases} \tag{2.1}$$

where $e = (1, 1, \cdots, 1)'$ and u is the given level of return. We assume that short selling is not allowed, then $\mathcal{N} = \{x \in R^n \mid x_i \geq 0, i = 1, 2, \cdots, n\}$ in Eq (2.1). For a confidence level $\alpha \in (0, 1)$, the VaR function, $\xi_\alpha(x)$, is given by the smallest number satisfying $\varphi(x, \xi_\alpha(x)) = \alpha$, where $\varphi(x, \xi)$ is the probability that the loss $f(x, y)$ does not exceed a threshold value ξ. Therefore, the objective function can be described as

$$\min\ CVaR(x) = (1 - \alpha)^{-1} \int_{f(x,y) \geq \xi_\alpha(x)} f(x, y) p(y) dy. \tag{2.2}$$

Rockfeller and Uryasev (2000) [7] showed that CVaR has the following equivalent definitions in Eq (2.3).

$$CVaR_\alpha(x) = \xi + (1 - \alpha)^{-1} \sum_{j=1}^{J} p(y_j)[f(x, y_j) - \xi]^+, \tag{2.3}$$

where $[x]^+$ represents $\max\{x, 0\}$. Furthermore, the mean-CVaR portfolio optimization model can be represented by the following equation.

$$\min_x CVaR_\alpha(x) = \min_{x,\xi} \{\xi + (1 - \alpha)^{-1} \sum_{j=1}^{J} p(y_j)[f(x, y_j) - \xi]^+\}, \tag{2.4}$$

$$s.t.\ x'r = u, \tag{2.5}$$

$$x'e = 1, x \in \mathcal{N}. \tag{2.6}$$

Equations (2.4)–(2.6) describe a constrained optimization problem. The mean-CVaR portfolio optimization model poses a challenge as it is a non-smooth optimization problem. To address this issue, Rockafellar and Uryasev (2000, 2002) [7, 8] proposed a transformation of the problem into a linear programming formulation by introducing some constraints, resulting in the addition of many decision variables. However, the effectiveness of this method decreases significantly as the sample size increases. To overcome this issue, many researchers are actively exploring solving these optimization problems with heuristic search algorithms (Najafi and Mushakhian 2015 [59]; Zhang and Liu 2017 [19]; Leung and Wang 2022 [60]). In this paper, we propose a novel hybrid meta-heuristics algorithm that combines the beluga whale optimization algorithm with the cross-entropy method to tackle this problem.

2.2. Jump-diffusion model and its parameters estimation

Let $S(t)$ be the stock price at time t, which follows a jump-diffusion process:

$$dS(t) = S(t-)\left\{\mu dt + \sigma dW(t) + d\left(\sum_{i=1}^{N(t)}(V_i - 1)\right)\right\}, \tag{2.7}$$

where μ and σ are the drift and volatility terms of the asset, $W(t)$ is a standard Brownian motion in Eq (2.7), $N(t)$ is a Poisson process with rate λ, and $\{V_i\}$ is a sequence of independent identically distributed ($i.i.d.$) nonnegative random variables, which satisfies $\ln V \sim N(\mu_J, \sigma_J^2)$. By solving the stochastic differential equation (2.7), we can obtain the dynamics of the asset price process:

$$S(t) = S(0)e^{\{(\mu-\frac{1}{2}\sigma^2)t+\sigma W(t)\}\prod_{i=1}^{N(t)} V_i}. \tag{2.8}$$

Additionally, we can represent the discretized logarithmic price process corresponding to Eq (2.8) as

$$S(t+\triangle t) = S(t)e^{(\mu-\frac{1}{2}\sigma^2)\triangle t+\sqrt{\triangle t}Z_t+\sum_{j=N(t)+1}^{N(t+\triangle t)} Y_j}, \tag{2.9}$$

where $Z_t \sim N(0,1)$, and $Y_j = \ln V_j$ is a normal distribution.

Based on the historical data of the stock price, the parameters of the price model can be estimated using maximum likelihood estimation with continuous compounding interest. Relevant articles for reference include Sorensen (2004) [61], Cont and Tankov (2004) [62], and Ardia et al. (2011) [63].

Algorithm 3.1 Pseudo-code of the CE method.

Begin

Initialize by setting a parameter vector $\hat{v}_0 = u$ and choosing a quantity ρ.

Set $t = 1$(iteration counter).

While $t < t_{max}$

Generate a sample $X_1, X_2, ..., X_N$ from the density $f(\cdot; v_{t-1})$ and calculate the performances $S(X_i)$ for all i, and sort them in ascending order, $S_{(1)} \leq S_{(2)} \leq \cdots \leq S_{(N)}$.

Let $\hat{\gamma}_t$ be $1-\rho$ sample quantile of the performances, and use the same sample $X_1, X_2, \cdots, X_N$ to solve Eq (3.3) .

Update $\hat{v}_t$ by Eq (3.4).

Set $t = t + 1$.

End While

Output the best solution and optimal value.

End

3. The proposed approach

3.1. The cross-entropy method

The cross-entropy (CE) method, based on the Monte Carlo technique, was introduced by Rubinstein (1997) [64] for estimating rare-event probabilities. This method exhibits good global search capability, excellent adaptability, and strong robustness. The CE method has been successfully applied to a wide range of complex optimization problems, including probability estimation (De Boer et al. 2004 [65]; Chan and Kroese 2012 [66]), buffer allocation problem (Alon et al. 2005 [67]), combination optimization (Rubinstein 1999 [64]; Caballero et al. 2015 [68]), continuous optimization (Rubinstein 1999 [69]; Kroese et al. 2006 [70]), multi-objective optimization (Bekker and Aldrich 2011 [71]; Caballero et al. 2015 [68]), scheduling and vehicle routing optimization problems (Chepuri and Homem-De-Mello et al. 2005 [72]), as well as other complex problems (Szita and Lörincz 2006 [73]; Laguna et al. 2009 [74]; Maher et al. 2013 [75]; Lamonica et al. 2020 [76]; Cardoso et al. 2022 [77]). The CE method for a general optimization problem is described as follows:

$$\min_{x\in\chi} = F(x), \tag{3.1}$$

where x is a decision variable and F is a real-valued function defined on the feasible domain χ. We can transform the optimization problem defined in Eq (3.1) into a probability distribution estimation problem by introducing the auxiliary problem

$$l(\gamma) = \frac{1}{N}\sum_{i=1}^{N} I_{F(X)\leq\gamma}\frac{f(x^i;v)}{g(x^i)}, \tag{3.2}$$

where I denotes the indicator function, N represents the sample size, x^i is a random sample drawn from a probability density function $f(x;v)$ using the importance sampling density $g(x)$, v is the probability distribution parameter, and γ is a threshold parameter. As a result, Eq (3.1) can be expressed in the following form (Kroese et al. 2006):

$$\min_{v} = \frac{1}{N}\sum_{i=1}^{N} I_{F(X)\leq\gamma} \ln f(x^i;v). \tag{3.3}$$

To enhance the convergence speed of the cross-entropy method for solving the optimization problem described in Eq (3.3), the adaptive smoothing parameter $\hat{v}$ is replaced with $\tilde{v}$. The updated equation (3.4) is as follows:

$$\hat{v}_{t+1} = \alpha\tilde{v} + (1-\alpha)\hat{v}_t. \tag{3.4}$$

The pseudocode of the CE method for optimization problems is presented in Algorithm 3.1.

3.2. Beluga whale optimization algorithm

The beluga whale optimization (BWO) algorithm is a recently proposed swarm-based metaheuristic algorithm introduced by Zhong et al. (2022) [52]. The inspiration for this algorithm is drawn from the social behaviors exhibited by beluga whales, including swimming, preying, and whale fall. The mathematical model of the BWO algorithm is formulated as follows.

3.2.1. Initialization

In the population mechanism of the BWO algorithm, each beluga whale is considered as a search agent. The movement of the search agent within the search space is achieved by modifying its position vector. The matrix representing the location of the search agents can be expressed by

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix}, \tag{3.5}$$

where n is the population size of beluga whales, d represents the dimension of the design variables, and the fitness values for each beluga whale are stored as follows:

$$F_X = \begin{bmatrix} f(x_{11}, x_{12}, \cdots, x_{1d}) \\ f(x_{21}, x_{22}, \cdots, x_{2d}) \\ \vdots \\ f(x_{n1}, x_{n2}, \cdots, x_{nd}) \end{bmatrix}. \tag{3.6}$$

The BWO algorithm can transfer from exploration to exploitation, depending on the balance factor B_f, which is modeled by

$$B_f = B_0(1 - 0.5 \times \frac{T}{T_{max}}), \tag{3.7}$$

where T is the current iteration, T_{max} is the maximum iterative number, and B_0 randomly changes between $(0, 1)$ at each iteration. The exploration phase happens when the balance factor $B_f > 0.5$ while the exploitation phase happens when $B_f \leq 0.5$. With the increasing of iteration T, the fluctuation range of B_f is reduced from 1 to 0.5, illustrating the significant change of probabilities for exploitation and exploration phase, while the probability of the exploitation phase is increased with the ever-increasing iteration T.

3.2.2. Exploration phase

In the exploration phase of the BWO algorithm, the position update is determined by the swimming behavior of beluga whales. The update rule for the beluga whales' positions is established based on the observed behaviors of beluga whales in human care, specifically their social-sexual behaviors performed under different postures. One such behavior is the pair swim, where two beluga whales swim closely together in a synchronized or mirrored manner. Therefore, the update rule for the positions of the beluga whales can be expressed as

$$\begin{cases} X_{ij}^{T+1} = X_{i,p_j}^{T} + (X_{r,p_1}^{T} - X_{i,p_j}^{T})(1 + r_1)\sin(2\pi r_2), & \text{if } j \text{ is even}, \\ X_{ij}^{T+1} = X_{i,p_j}^{T} + (X_{r,p_1}^{T} - X_{i,p_j}^{T})(1 + r_1)\cos(2\pi r_2), & \text{if } j \text{ is odd}, \end{cases} \tag{3.8}$$

where T is the current iteration, $X_{i,j}^{T+1}$ is the new position for the i-th beluga whale on the j-th dimension, p_j $(j = 1, 2, \cdots, d)$ is a random number selected from d-dimension, X_{i,p_j}^{T} and X_{r,p_1}^{T} are the current positions of the i-th and r-th beluga whale (r is a randomly selected beluga whale), r_1 and r_2 are random numbers between (0, 1), and $\sin(2\pi r_2)$ and $\cos(2\pi r_2)$ mean fins of the mirrored beluga whales are toward the surface. In the BWO algorithm, the updated position of the beluga whales is determined based on the dimension chosen as either odd or even. This selection of odd or even dimensions reflects the synchronous or mirror behaviors exhibited by beluga whales during swimming or diving.

Algorithm 3.2 Pseudo-code of the BWO algorithm.

Begin
 Set parameters and initialize population X (the positions of beluga whales).
 Calculate the fitness values of all beluga whales, and select the optimal solution.
 While $T < T_{max}$
  Calculate the balance factor B_f and whale fall probability W_f according to Eqs (3.7) and (3.15), respectively.
  For $i = 1 : N$
   If $B_f(i) > 0.5$
    Implement the exploitation phase and update the ith beluga whale's position by Eq (3.8).
   Else
    Implement the exploration phase and update the ith beluga whale's position by Eq (3.9).
   End If
  End For
  For $i = 1 : N$
   If $B_f(i) \leq W_f$
    Implement the whale fall phase and update the ith beluga whale's position by Eq (3.13).
   End If
   Calculate the fitness value, and update the optimal solution and optimal value.
  End For
  Calculate the fitness value, and update the optimal solution and optimal value.
 End While
 Output the best solution and optimal value.
End

3.2.3. Exploitation phase

Beluga whales in the BWO algorithm utilize a Lévy flight strategy to capture prey, which simulates their movement pattern during hunting. This strategy is represented by

$$X_i^{T+1} = r_3 X_{best}^T - r_4 X_i^T + C_1 \cdot L_F \cdot (X_r^T - X_i^T), \tag{3.9}$$

where X_i^T and X_i^{T+1} are the current and new position of the i-th beluga whale, X_r^T is the current position for a random beluga whale, X_{best}^T is the best position among beluga whales, $r3$ and $r4$ are random numbers between (0, 1), and C_1 is the random jump strength and can be calculated as

$$C_1 = 2r_4(1 - \frac{T}{T_{max}}), \tag{3.10}$$

where C_1 measures the intensity of Lévy flight. L_F is the Lévy flight function, defined as

$$L_F = 0.05 \times \frac{\mu \times \sigma}{|v|^{1/\beta}}), \tag{3.11}$$

where u and v are normally distributed random numbers, β is the default constant equal to 1.5, and σ is defined as

$$\sigma = \left[\frac{\Gamma(1+\beta) \times \sin(\pi\beta/2)}{\Gamma((1+\beta)/2) \times \beta \times 2^{(\beta-1)/2}} \right]^{1/\beta}. \tag{3.12}$$

3.2.4. Whale fall

To maintain a constant population size in the BWO algorithm, the updated positions of beluga whales are determined using the current positions and the step size of the whale fall. The formula for updating the positions is

$$X_i^{T+1} = r_5 X_i^T - r_6 X_r^T + r_7 X_{step}, \tag{3.13}$$

where r_5, r_6, and r_7 are random numbers in $(0, 1)$, and X_{step} is the step size of whale fall calculated as

$$X_{step} = (ub - lb)exp(-C_2 \frac{T}{T_{max}}), \tag{3.14}$$

where the step size factor C_2 is related to the probability of beluga whale fall W_f and population size n ($C_2 = 2nW_f$). ub and lb are the upper and lower bounds of variables, respectively. The probability of a whale fall is calculated as a linear function, and the formula is

$$W_f = 0.1 - 0.05 \times \frac{T}{T_{max}}. \tag{3.15}$$

The probability of whale fall decreases from the initial iteration 0.1 to the final iteration, which reflects the reduced risk of beluga whales dying as they approach the food source during the optimization process. The pseudo-code of the BWO is given in Algorithm 3.2.

3.3. Hybridization of cross-entropy method and beluga whale optimization algorithm

3.3.1. The procedure of CEBWO

This section presents the details of the proposed CEBWO algorithm, which combines the cross-entropy method (CE) and the beluga whale optimization algorithm (BWO). A successful meta-heuristic method should incorporate both exploitation and exploration functions while maintaining a proper balance between them for optimal performance (Eiben and Schipper 1998) [78]. Although the nature-inspired BWO has demonstrated advantages in solving global optimization problems, it can still suffer from issues such as local optima and unbalanced development, limiting its effectiveness in exploring the entire search space (Chen et al. 2023 [79]; Hussien et al. 2023 [80]).

To address these limitations, this paper introduces the CEBWO algorithm, which utilizes the Monte Carlo technique. The algorithm combines the BWO and CE optimization operators based on co-evolutionary technique and performs co-updates on the BWO population (Pop_{BWO}) and the CE sample (Pop_{CE}) in each iteration. By incorporating the CE method, the population diversity of the BWO algorithm is enhanced, leading to improved convergence rates. Meanwhile, the CE operator obtains initial probability parameters using the BWO population (Pop_{BWO}) to enhance its convergence rate.

Algorithm 3.3 Pseudo-code of the CEBWO algorithm.

```
Begin
  Set the parameters of the BWO and CE operators, and initialize population X .
  Calculate the fitness values of all beluga whales, and obstain the current best solution and fitness
        value.
 While T < T_max
    Calculate the balance factor B_f and whale fall probability W_f according to Eqs (3.7) and (3.15),
    respectively.
    For i = 1 : N
        If B_f(i) > 0.5
            Implement the exploitation phase and update the ith beluga whale's position by
            Eq (3.8).
        Else
            Implement the exploration phase and update the ith beluga whale's position by
            Eq (3.9).
        End If
    End For
    For i = 1 : N
        If B_f(i) ≤ W_f
            Implement the whale fall phase and update the ith beluga whale's position by
            Eq (3.13).
        End If
        Calculate the fitness value, and update the best solution and fitness value.
    End For
    While t < T_CE
        Calculate the probability distribution parameter μ̃ and σ̃, update ṽ_t based on X by Eq (3.4).
        Generate and evaluate sample Y_1, Y_2, ··· , Y_N.
        Co-update X using Y, update the current best solution and fitness value.
        Set t = t + 1
    End While
    Set T = T + 1
 End While
 Output the best solution and optimal value.
End
```

The pseudo-code of the CEBWO hybrid algorithm is described in Algorithm 3.3. Figure 1 presents the flowchart of the CEBWO algorithm, illustrating the co-evolutionary process between the CE operator and the BWO operator. The hybrid algorithm CEBWO combines the strengths of the CE and BWO methods to overcome the limitations of the original BWO algorithm. It achieves improved population diversity and convergence rates by performing co-evolutionary updates on the BWO population (Pop_{BWO}) and the CE sample (Pop_{CE}) in each iteration.

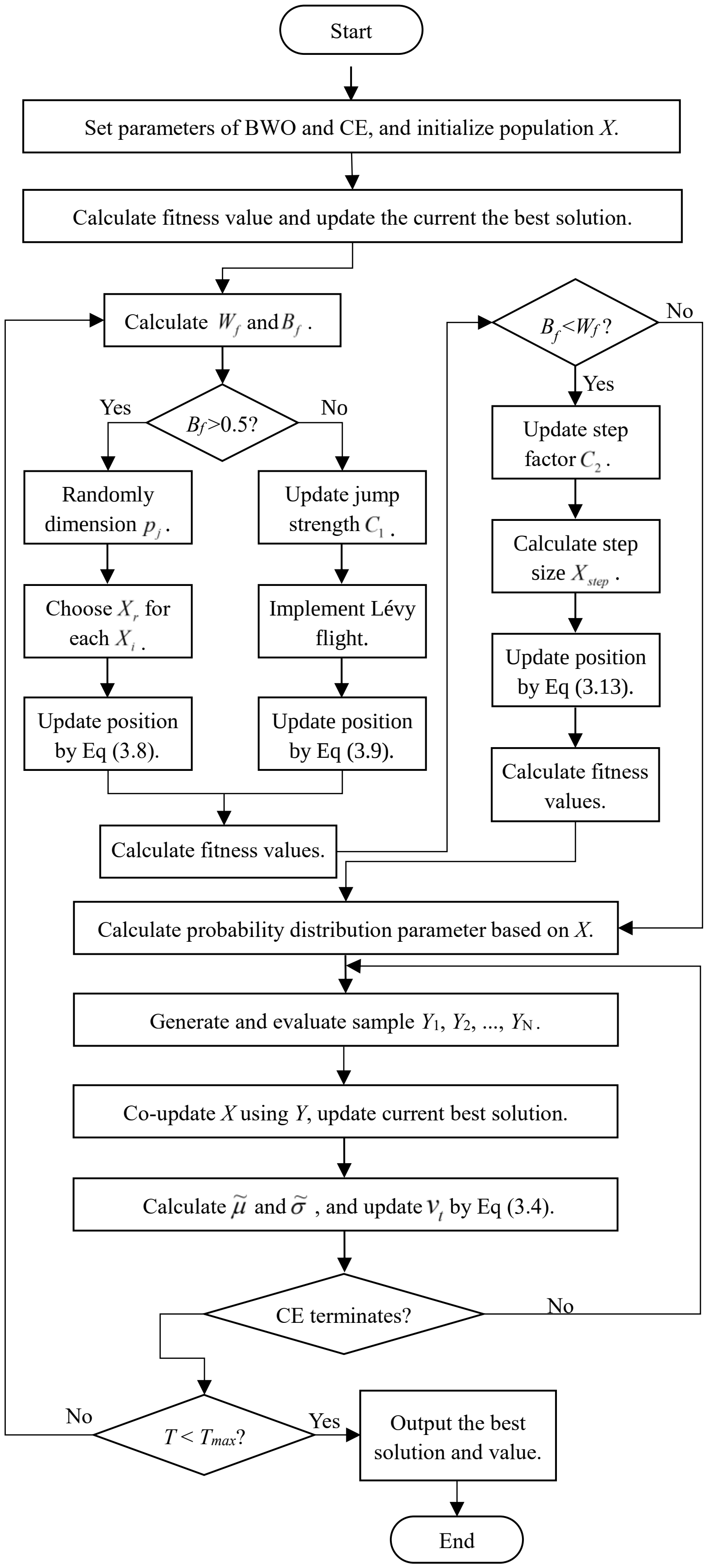

Figure 1. The flowchart of CEBWO.

3.3.2. Computational complexity

The computational complexity of the proposed CEBWO algorithm plays a crucial role in assessing its performance. The CEBWO algorithm consists of two main operators: BWO and CE. The BWO operator comprises three processes: initialization, fitness evaluation, and updating of the beluga whales. It is important to note that the computational complexity of the initialization process is $O(n_1)$, where n_1 represents the number of beluga whales. During the exploration and exploitation phase, the computational complexity is calculated as $O(n_1 \times T_{max})$, where T_{max} denotes the maximum number of iterations. In the whale fall phase, the computational complexity is influenced by the probability of whale fall W_f and the balance factor B_f. This phase can be approximated as $O(0.1 \times n_1 \times T_{max})$. On the other hand, the CE operator includes one inner loop for the sample size n_2 and two outer loops for iterations T_{max} and T_{CE}, respectively. Hence, the computational complexity of the CE operator can be approximated as $O(n_2 \times (1 + T_{max} \times T_{CE}))$. Therefore, the computational complexity of our proposed hybrid method, CEBWO, can be evaluated approximately as $O(n_1 \times (1 + 1.1 \times T_{max}) + n_2 \times (1 + T_{max} \times T_{CE}))$. It should be noted that the complexity is linear in terms of $T_{max} \times T_{CE}$, which represents the total number of iterations in the CEBWO algorithm.

3.3.3. Efficiency analysis of co-evolution

The proposed hybrid meta-heuristic algorithm CEBWO utilizes co-evolutionary technology to achieve a balance between exploration and exploitation. This is accomplished through collaborative updating of the optimal solution and value by the CE and BWO operators, iterative parameter updates based on the BWO population, and integration of the CE operator's results with the BWO population. These strategies enhance the algorithm's performance and search capabilities. By employing these co-evolutionary strategies, CEBWO effectively combines the strengths of both operators, resulting in improved optimization performance and a more balanced exploration-exploitation trade-off.

Figure 2 shows the specific process of co-evolution when the hybrid algorithm is used to solve F9 selected from the benchmark functions, where "o" is the optimal function value updated by the CE operator and "." is updated by the BWO operator. This fully demonstrates that the co-evolutionary technology can be well implemented in the proposed method and the optimal function value is collaboratively updated by the two operators CE and BWO during the iterative process.

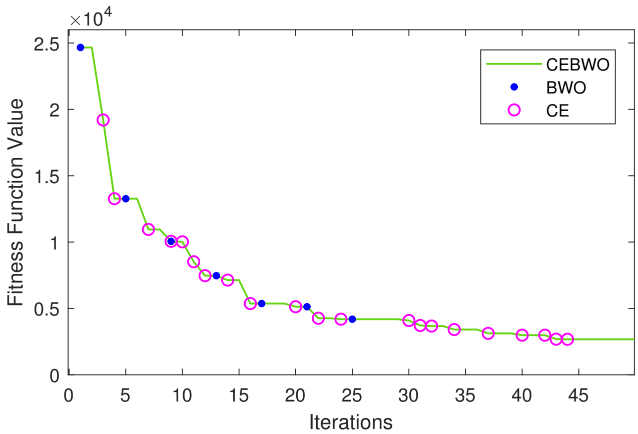

Figure 2. Efficiency analysis of co-evolution: CE and BWO co-update the current best in CEBWO's iterative process.

3.3.4. Performance of CEBWO algorithm in solving function optimization problems across different dimensions

To investigate the impact of search space dimension on the optimization performance and convergence rate of the CEBWO algorithm when tackling high-dimensional function optimization problems, we conducted comparative analyses. In addition to the original BWO algorithm (Zhong, 2022) [52], we included several recently proposed swarm intelligence optimization algorithms, namely WOA (Mirjalili, 2016) [29], AVOA (Abdollahzadeh, 2021) [45], HBA (Hashim, 2022) [47], and AHA (Zhao, 2022) [48], as reference benchmarks. The objective of our study was to assess the performance of the aforementioned algorithms on the more complex F9 function, which is part of the CEC 2017 benchmark function set. To comprehensively evaluate the algorithms' capabilities, we conducted a series of tests in which we varied the independent variable dimension across 2, 10, 30, 50, and 100 dimensions. The results of these evaluations are presented in Figure 3.

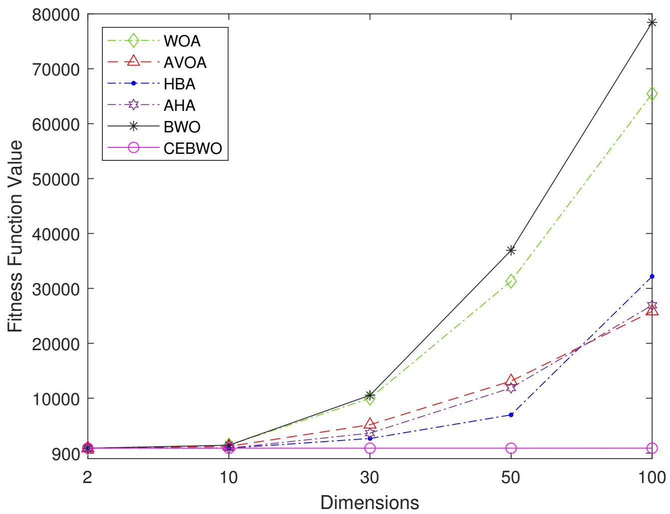

Figure 3. Comparison of optimization accuracy of different search space dimensions.

The analysis of Figure 3 reveals a remarkable characteristic of the proposed CEBWO algorithm: Its accuracy remains relatively robust even with an increase in search space dimension. This distinguishes CEBWO from the BWO, WOA, AVOA, HBA, and AHA algorithms, which exhibit a more pronounced decrease in accuracy as the dimensionality increases. These findings highlight the novelty and effectiveness of the CEBWO approach in tackling high-dimensional function optimization problems while mitigating the typical degradation in solution accuracy associated with dimensionality. Moreover, the results demonstrate that the optimization space dimension has minimal impact on the performance of the CEBWO algorithm, in contrast to the other algorithms, where dimensionality significantly affects their performance. This observation further underscores the suitability of the CEBWO algorithm for addressing high-dimensional function optimization problems.

3.4. Benchmark functions and experimental setup

To assess the performance of our novel hybrid algorithm, we conducted a rigorous evaluation using 29 unconstrained benchmark functions from CEC 2017. These functions encompass a diverse range of problem types, including two single-peak functions (F1 and F3), seven simple multi-peak functions (F4 to F10), nineteen mixed function sets (F11 to F29), and ten combined test functions (F21 to F30). The mathematical expressions, search intervals, and other parameters of these benchmark functions can be found in the comprehensive work by Awad et al. (2016) [81]. To provide a comprehensive comparison, we included the original BWO algorithm (Zhong, 2022) [52] along with several recently proposed swarm intelligence optimization algorithms, namely WOA (Mirjalili, 2016) [29], AVOA (Abdollahzadeh, 2021) [45], HBA (Hashim, 2022) [47], and AHA (Zhao, 2022) [48]. The parameter configurations of the meta-heuristic algorithms are presented in Table 1, where n represents the population size of AHA, while other relevant parameters were adopted from the respective original publications. Throughout the experiments, we employed a consistent setup with 40 populations, a maximum of 1000 iterations, and a fixed dimensionality of 30 for all benchmark functions. Specifically, the maximum number of iterations T_{max} and T_{CE} for the two operators BWO and CE in the CEBWO algorithm were set to 50 and 20, respectively. Each algorithm was independently executed 30 times on each benchmark problem to ensure result reliability and robustness. The experimental evaluations were conducted on a computational platform equipped with an Intel (R) Core (TM) i7-9700 CPU @ 3.00GHz and 16 GB RAM. The software environment employed was Windows 11, coupled with MATLAB 2021a for algorithm implementation and analysis.

Table 1. Algorithm parameter settings.

Algorithm	Parameter	Value
WOA	Probability of encircling mechanism, spiral factor	0.5,1
AVOA	$L_1, L_2, w, P_1, P_2, P_3$	0.8,0.2,2.5,0.6,0.4,0.6
HBA	Ability of a honey badger to get food β, C	6,2
AHA	Migration coefficient	$2n$
BWO	Probability of whale fall decreased at interval W_f	[0.05,0.1]
CEBWO	fixed and dynamic smoothing coefficient of CE α and β	0.8,0.7

4. Numerical experiment

4.1. Results and discussion

Table 2 displays the mean and standard deviation (STD) of fitness values achieved by each algorithm on the CEC 2017 test functions in a 30-dimensional space. The best mean and STD are highlighted in bold black. Analyzing Table 2, we observe several key findings.

First, it becomes evident that the proposed CEBWO algorithm consistently achieves highly competitive results compared to the other algorithms across the CEC 2017 test functions. For instance, considering the multi-modal function F6, with an optimal value of 600, CEBWO obtains an average fitness value of 6.00E+02, with a standard deviation of 1.99E-07, indicating proximity to the optimal value. In comparison, the average fitness values obtained by WOA, AVOA, HBA, AHA, and BWO are 6.87E+02, 6.48E+02, 6.08E+02, 6.04E+02, and 6.88E+02, respectively. Furthermore, examining the multi-modal function F9, with an optimal value of 900, CEBWO attains an average fitness value of 9.00E+02, with a standard deviation of 0.00E+00. The low standard deviation implies that all 30 independent experiments yield the optimal solution. Conversely, the mean fitness values for

WOA, AVOA, HBA, AHA, and BWO are 1.21E+04, 4.92E+03, 2.23E+03, 3.22E+03, and 1.03E+04, respectively.

Table 2. Comparison of results for CEC 2017 benchmark functions.

Fun.	Meas.	WOA	AVOA	HBA	AHA	BWO	CEBWO
F1	Mean	4.62E +10	5.42E +03	5.44E +03	8.66E +04	4.89E +10	**1.76E+03**
	STD	7.41E +09	6.27E +03	5.52E +03	9.60E +04	3.11E +09	**2.13E+03**
F3	Mean	8.68E +04	2.01E +04	1.12E +04	1.66E +04	7.62E +04	**5.88E+03**
	STD	6.12E +03	5.65E +03	4.37E +03	5.87E +03	5.37E +03	**2.20E+03**
F4	Mean	8.39E +03	5.14E +02	4.95E +02	**4.93E+02**	1.19E +04	5.13E +02
	STD	2.92E +03	3.33E +01	2.48E +01	3.01E +01	1.27E +03	**8.00E+00**
F5	Mean	9.30E+02	7.12E+02	**6.07E+02**	6.43E+02	9.19E+02	6.61E+02
	STD	3.76E+01	4.66E+01	2.82E+01	3.44E+01	1.63E+01	**9.52E+00**
F6	Mean	6.87E+02	6.48E+02	6.08E+02	6.04E+02	6.88E+02	**6.00E+02**
	STD	9.68E+00	8.17E+00	4.88E+00	6.19E+00	4.73E+00	**1.99E-07**
F7	Mean	1.46E+03	1.13E+03	**8.88E+02**	9.29E+02	1.36E+03	8.89E+02
	STD	5.81E+01	8.44E+01	4.61E+01	6.76E+01	3.45E+01	**1.02E+01**
F8	Mean	1.17E+03	9.62E+02	**8.96E+02**	9.21E+02	1.13E+03	9.59E+02
	STD	2.25E+01	2.80E+01	1.93E+01	2.58E+01	1.45E+01	**1.25E+01**
F9	Mean	1.21E+04	4.92E+03	2.23E+03	3.22E+03	1.03E+04	**9.00E +02**
	STD	1.99E+03	6.68E+02	9.07E+02	1.06E+03	7.88E+02	**0.00E+00**
F10	Mean	8.98E+03	5.35E+03	5.47E+03	**4.41E+03**	8.49E+03	7.82E+03
	STD	3.06E+02	5.93E+02	1.56E+03	7.20E+02	4.08E+02	**3.39E+02**
F11	Mean	9.38E+03	1.28E+03	1.25E+03	1.19E+03	7.02E +03	**1.18E+03**
	STD	3.12E+03	5.40E +01	7.59E+01	**3.70E +01**	8.88E +02	5.03E+00
F12	Mean	5.04E+09	4.36E+06	**2.37E+05**	1.62E+06	1.00E+10	7.86E+05
	STD	1.16E+09	3.67E+06	5.13E+05	1.20E+06	2.08E+09	**4.90E+05**
F13	Mean	1.27E+09	1.10E+05	4.04E+04	1.91E+04	5.94E+09	**9.95E+03**
	STD	2.50E +08	4.90E +04	3.59E +04	2.07E +04	1.25E +09	**7.81E+03**
F14	Mean	2.44E+06	1.99E+05	**1.04E+04**	1.90E+04	2.78E+06	1.80E+05
	STD	1.21E +06	1.92E +05	**7.90E+03**	2.30E +04	1.25E +06	1.83E +05
F15	Mean	4.66E+08	3.56E+04	1.10E+04	4.05E+03	2.00E+08	**3.64E+03**
	STD	2.87E +08	3.01E +04	9.98E +03	2.93E +03	1.06E +08	**2.50E+03**
F16	Mean	4.77E+03	2.99E+03	2.72E+03	2.60E+03	5.20E+03	**2.30E+03**
	STD	4.22E+02	3.56E+02	4.89E+02	**2.95E+02**	3.21E+02	4.49E+02
F17	Mean	3.15E+03	2.45E+03	2.21E+03	2.18E+03	3.72E+03	**1.84E+03**
	STD	2.78E +02	2.45E +02	2.11E +02	1.78E +02	4.07E +02	**9.81E+01**
F18	Mean	4.28E+07	1.24E+06	3.37E+05	**1.78E+05**	2.78E+07	4.29E+05
	STD	2.45E +07	1.36E +06	3.44E +05	**1.74E+05**	1.61E +07	4.04E +05
F19	Mean	5.99E+08	1.88E+04	1.31E+04	8.67E+03	3.12E+08	**5.55E+03**
	STD	3.20E +08	1.70E +04	1.56E +04	7.83E +03	1.30E +08	**3.86E+03**
F20	Mean	3.05E+03	2.68E+03	2.54E+03	2.42E+03	2.90E+03	**2.21E+03**
	STD	1.54E +02	2.18E +02	2.32E +02	1.51E +02	1.24E +02	**1.37E+02**

F21	Mean	2.71E+03	2.51E+03	**2.40E+03**	**2.40E+03**	2.71E+03	2.45E+03
	STD	4.47E +01	4.99E +01	2.81E +01	2.50E +01	3.18E +01	**9.59E+00**
F22	Mean	7.30E+03	5.31E+03	3.23E+03	**2.30E+03**	8.28E+03	2.74E+03
	STD	1.23E +03	2.39E +03	1.91E +03	**1.55E+00**	6.30E +02	1.69E +03
F23	Mean	3.30E+03	2.93E+03	2.78E+03	2.78E+03	3.29E+03	**2.77E+03**
	STD	1.01E +02	8.80E +01	3.38E +01	**2.75E+01**	5.28E +01	5.90E+01
F24	Mean	3.37E+03	3.14E+03	**2.94E+03**	2.96E+03	3.53E+03	2.97E+03
	STD	1.01E +02	1.01E +02	5.93E +01	4.46E +01	5.43E +01	**2.93E+01**
F25	Mean	5.24E+03	2.91E+03	2.90E+03	2.91E+03	4.32E+03	**2.89E+03**
	STD	4.55E +02	2.19E +01	1.56E +01	2.01E +01	1.26E +02	**2.64E+00**
F26	Mean	1.01E+04	6.02E+03	4.78E+03	**3.97E+03**	1.04E+04	4.04E+03
	STD	7.65E +02	1.74E +03	6.78E +02	1.46E +03	**5.35E+02**	6.22E+02
F27	Mean	3.20E+03	3.28E+03	3.28E+03	3.26E+03	3.95E+03	**3.21E+03**
	STD	7.20E-05	3.44E +01	7.48E +01	2.36E +01	1.32E +02	**1.01E+01**
F28	Mean	3.30E+03	3.26E+03	3.22E+03	3.25E+03	6.32E+03	**3.24E+03**
	STD	**6.05E-05**	3.41E +01	2.00E +01	2.66E +01	2.63E +02	1.58E+00
F29	Mean	5.95E+03	4.27E +03	4.03E+03	3.72E+03	6.58E+03	**3.41E+03**
	STD	5.12E +02	2.63E +02	3.33E +02	2.05E +02	5.25E +02	**7.28E+01**
F30	Mean	4.42E+08	3.12E+05	1.48E+05	1.02E+04	7.89E+08	**8.96E+03**
	STD	1.93E +08	1.79E +05	5.12E +05	2.90E +03	3.32E +08	**2.18E+03**

Second, in contrast to the original BWO algorithm, the enhanced CEBWO algorithm proposed in this study consistently outperforms in terms of mean fitness values across all 29 test functions. Furthermore, it achieves the lowest standard deviation (STD) among the remaining 28 test functions, with the exception of F26. These findings convincingly demonstrate that the integration of the CE method into BWO significantly improves its optimization performance, enhances accuracy, and augments robustness.

Finally, in comparison to the other four swarm intelligence optimization algorithms, CEBWO demonstrates superior performance over WOA and AVOA in terms of the mean indicator across all 29 test functions. It exhibits a minor deviation from WOA in the STD indicator for the test function F28, while showcasing comparable performance to HBA and AHA. Notably, CEBWO outperforms HBA in 22 test functions and outperforms AHA in 23 test functions.

These findings substantiate the superior optimization performance of the CEBWO algorithm, effectively circumventing local optima and attaining global optimal solutions. Additionally, the noteworthy performance on the STD index highlights the stability and robustness of the algorithm. Notably, the algorithm exhibits minimal sensitivity to variations in the initial population, indicating that the population has minimal influence on its overall performance.

To conduct a comprehensive comparison between CEBWO and the five alternative algorithms (WOA, AVOA, HBA, AHA, and BWO), we employed the Wilcoxon rank sum test. Table 3 presents the experimental results obtained from executing CEBWO and the aforementioned algorithms 30 times, utilizing the CEC2017 benchmark functions with a dimensionality of 30. These results, supported by the data presented in Table 3, strongly suggest that CEBWO exhibits notable distinctions when compared to the other five approaches.

Table 3. Wilcoxon rank sum test results of CEC2017 with dim = 30.

Fun.	WOA	AVOA	HBA	AHA	BWO
F1	3.02E-11	4.86E-03	2.39E-04	3.02E-11	3.02E-11
F3	3.02E-11	3.34E-11	1.11E-06	9.76E-10	3.02E-11
F4	3.02E-11	1.30E-01	2.13E-04	4.71E-04	3.02E-11
F5	3.02E-11	7.60E-07	3.82E-09	1.27E-02	3.02E-11
F6	1.72E-12	1.72E-12	1.72E-12	1.72E-12	1.72E-12
F7	3.02E-11	3.02E-11	**4.29E-01**	2.92E-02	3.02E-11
F8	3.02E-11	**8.77E-01**	4.50E-11	4.69E-08	3.02E-11
F9	1.21E-12	1.21E-12	1.21E-12	1.21E-12	1.21E-12
F10	3.02E-11	3.02E-11	7.69E-08	3.02E-11	6.01E-08
F11	3.02E-11	1.73E-07	2.25E-04	**5.30E-01**	3.02E-11
F12	3.02E-11	5.97E-09	7.77E-09	3.03E-03	3.02E-11
F13	3.02E-11	5.49E-11	8.20E-07	**9.93E-02**	3.02E-11
F14	6.07E-11	**5.69E-01**	1.25E-07	1.73E-06	5.49E-11
F15	3.02E-11	6.07E-11	8.29E-06	**4.12E-01**	3.02E-11
F16	3.02E-11	2.20E-07	2.62E-03	8.68E-03	3.02E-11
F17	3.02E-11	6.70E-11	3.20E-09	3.50E-09	3.02E-11
F18	3.02E-11	1.33E-02	**1.15E-01**	4.94E-05	3.02E-11
F19	3.02E-11	1.39E-06	3.78E-02	**5.37E-02**	3.02E-11
F20	3.02E-11	4.62E-10	8.35E-08	5.86E-06	3.02E-11
F21	3.02E-11	1.16E-07	8.10E-10	1.41E-09	3.02E-11
F22	4.63E-10	1.24E-09	1.24E-09	1.24E-09	6.24E-10
F23	3.02E-11	2.23E-09	**4.83E-01**	**5.40E-01**	3.02E-11
F24	3.02E-11	2.92E-09	7.74E-06	**7.73E-02**	3.02E-11
F25	3.02E-11	3.34E-03	**3.79E-01**	7.69E-08	3.02E-11
F26	3.02E-11	6.77E-05	2.84E-04	**1.09E-01**	3.02E-11
F27	8.48E-09	8.99E-11	4.31E-08	1.61E-10	3.02E-11
F28	3.02E-11	1.91E-02	4.12E-06	**5.94E-02**	3.02E-11
F29	3.02E-11	3.02E-11	4.08E-11	8.89E-10	3.02E-11
F30	3.02E-11	3.02E-11	7.04E-07	3.64E-02	3.02E-11

Figures 4–7 illustrate the convergence analysis of various algorithms, including CEBWO, WOA, AVOA, HBA, AHA, and BWO, on the CEC 2017 benchmark function in a 30-dimensional space. These plots present the convergence curves, providing valuable insights into the algorithm's performance. A careful examination of these graphs reveals several noteworthy observations.

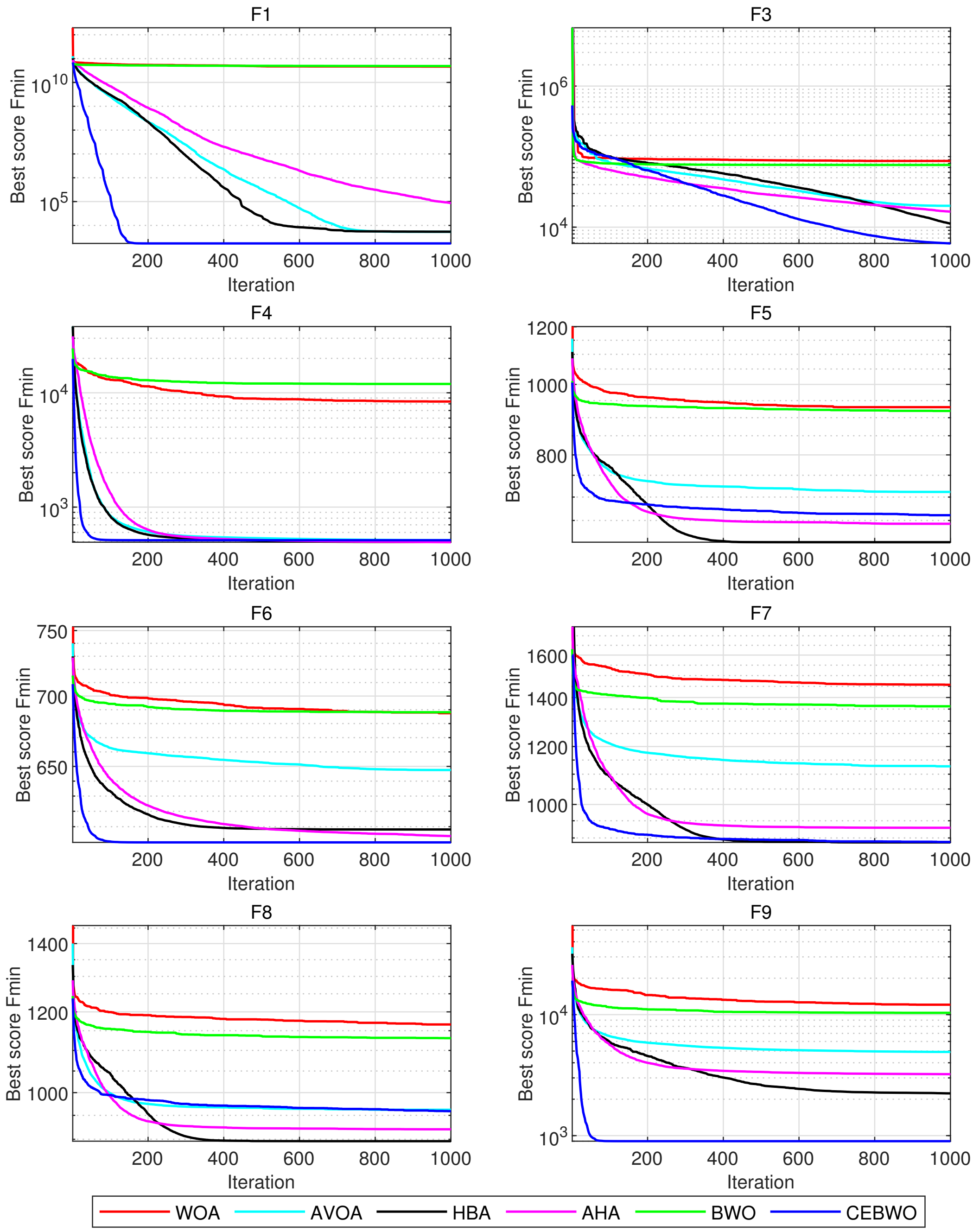

Figure 4. Convergence curves of CEBWO and comparison algorithm on CEC2017 (F1, F3 to F9).

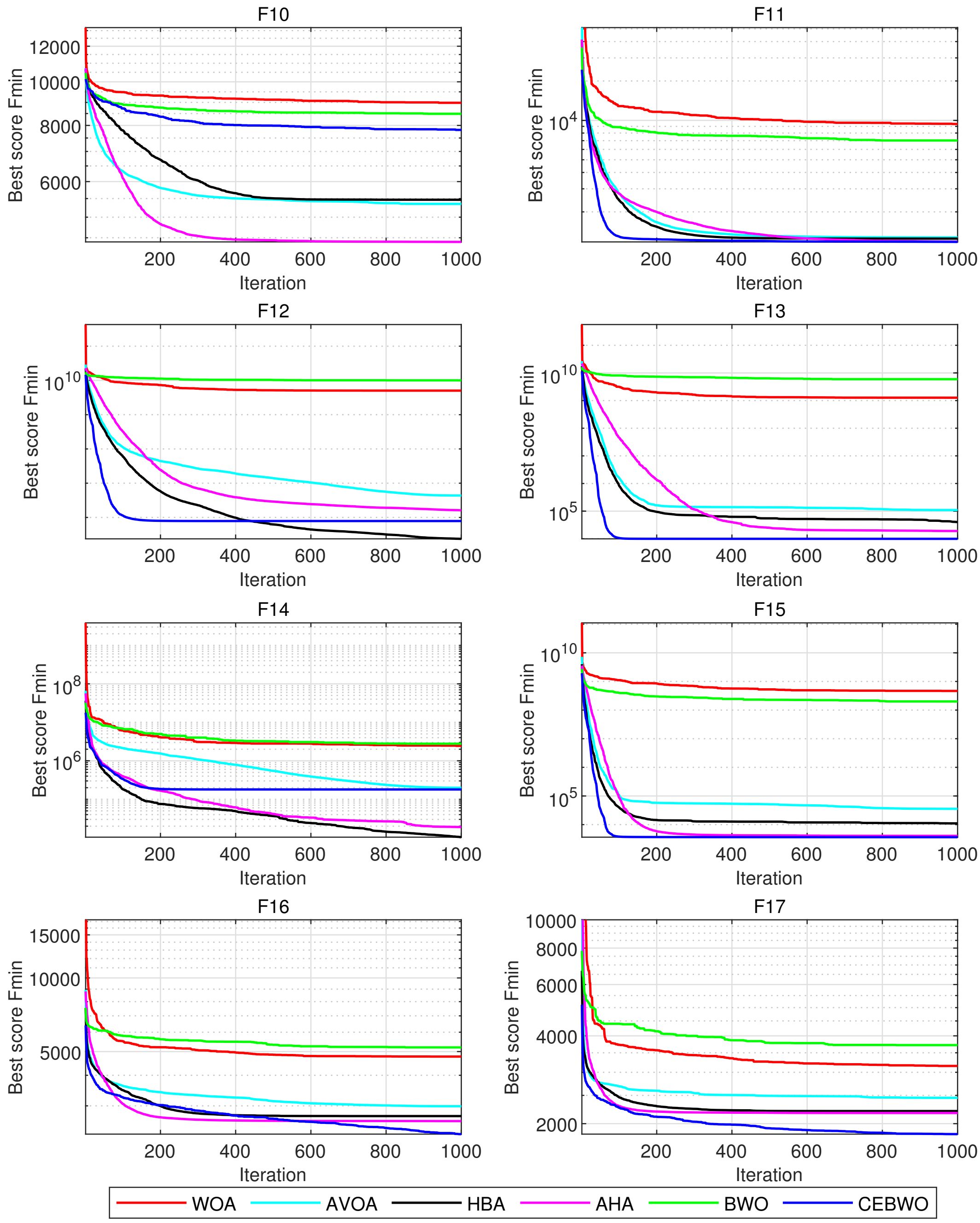

Figure 5. Convergence curves of CEBWO and comparison algorithm on CEC2017 (F10 to F17).

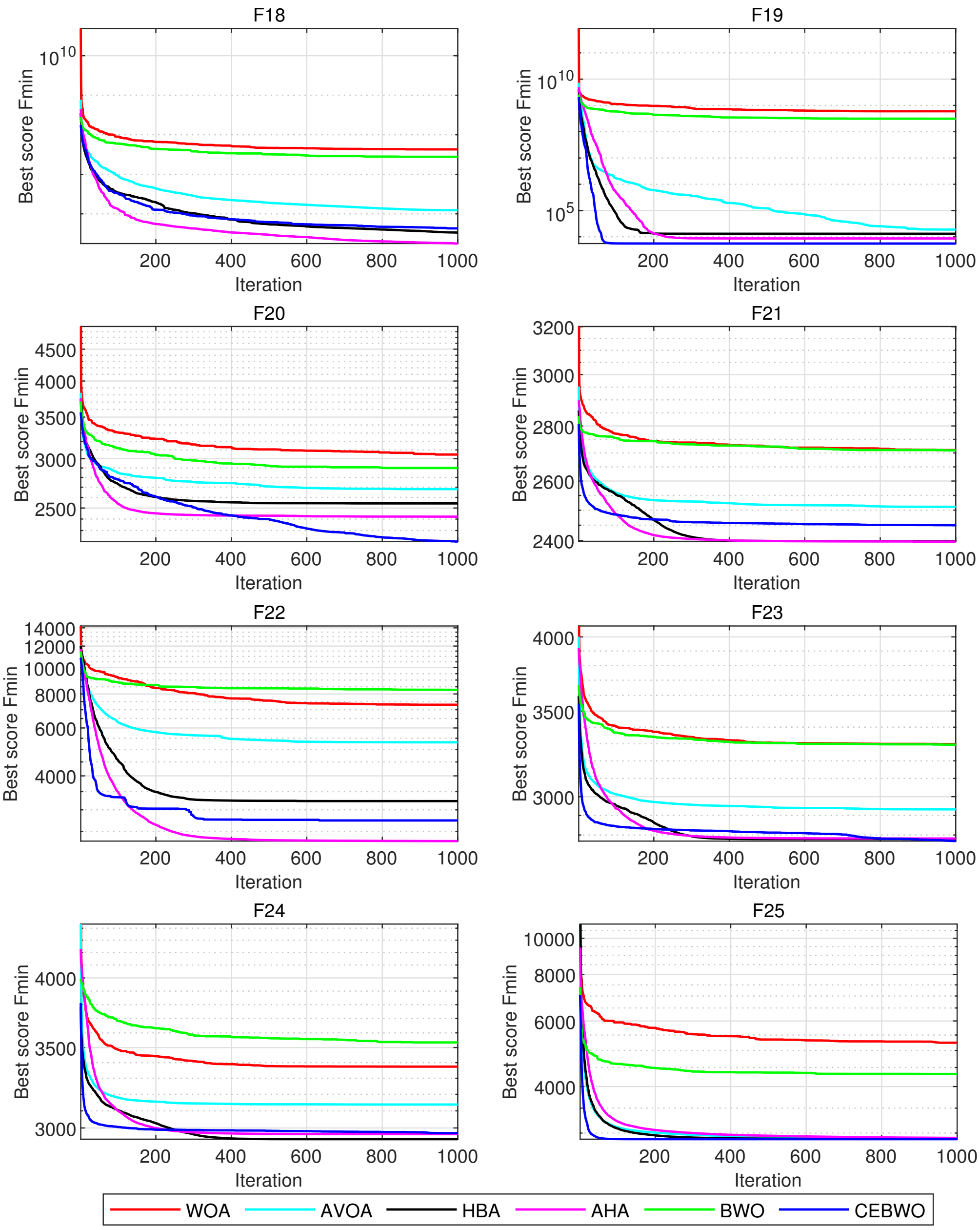

Figure 6. Convergence curves of CEBWO and comparison algorithm on CEC2017 (F18 to F25).

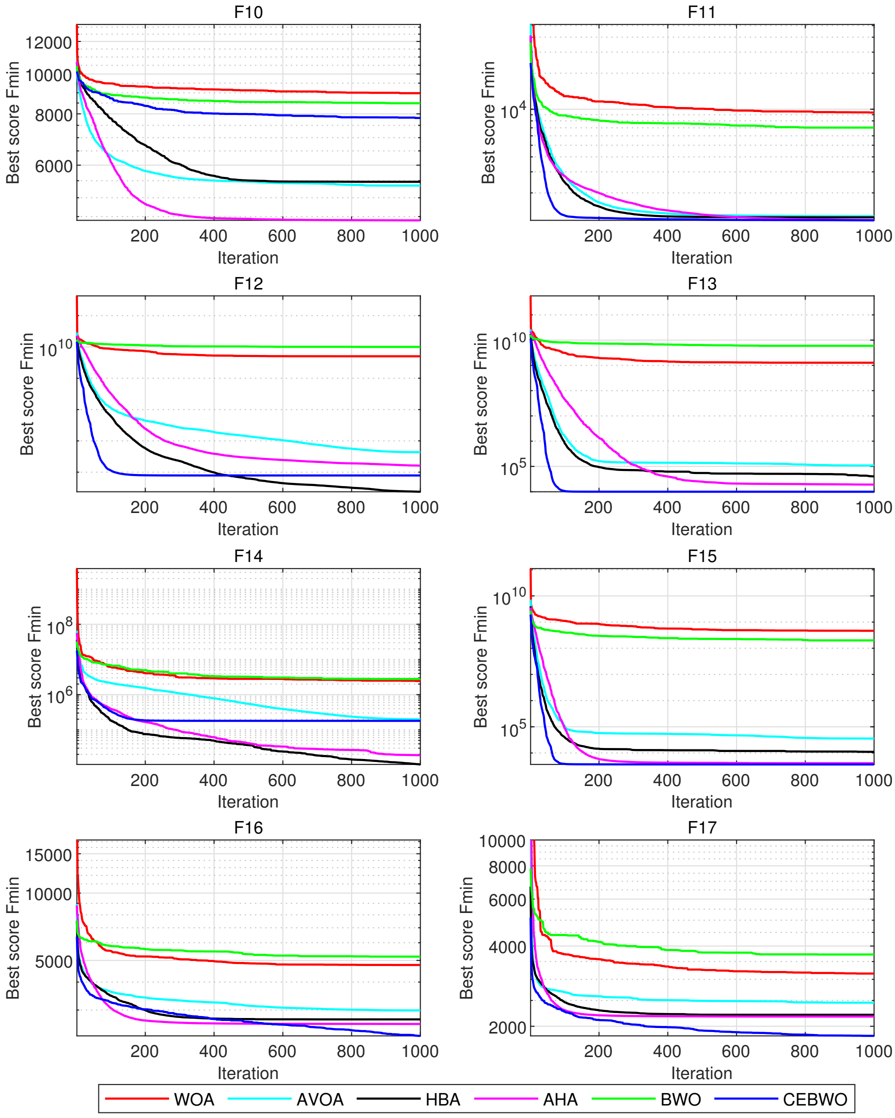

Figure 7. Convergence curves of CEBWO and comparison algorithm on CEC2017 (F26 to F30).

First, in comparison to the original BWO algorithm, the proposed CEBWO algorithm exhibits substantial enhancements in terms of convergence rate and solution accuracy across all 29 test functions. These findings clearly indicate that CEBWO effectively overcomes the limitations of premature convergence and low accuracy observed in BWO, particularly in complex optimization problems. Notably, CEBWO successfully mitigates the issues of local minima and demonstrates remarkable efficiency in rapidly converging towards the global optimal solution.

Furthermore, in comparison to WOA and the other four algorithms, CEBWO consistently demonstrates exceptional convergence characteristics and superior optimization performance. As illustrated in Figure 4, CEBWO exhibits rapid convergence in seven out of the eight functions, achieving high solution accuracy in five functions. Similarly, in Figure 5, CEBWO showcases rapid convergence in six functions while consistently attaining superior solution accuracy across all six functions. Moreover, in Figure 6, CEBWO demonstrates fast convergence in five functions and excels in solution accuracy in four functions. Notably, among the six functions depicted in Figure 7, CEBWO outperforms the other algorithms in terms of both convergence rate and solution accuracy in four functions.

These findings provide robust evidence supporting the conclusion that CEBWO demonstrates enhanced convergence characteristics, establishing it as a promising algorithm for optimization tasks. The observed performance enhancements, specifically in terms of convergence rate and solution accuracy, underscore the significant potential of CEBWO for effectively addressing complex optimization problems.

5. Application for mean-CVaR portfolio selection problem

5.1. Data selection and model parameters estimation

To evaluate the effectiveness of the proposed mean-CVaR portfolio optimization model, a portfolio is constructed using a selection of eight equity indices from the global market. The chosen indices include the SSE Composite Index (SSEC), DAX Performance Index (GDAXI), Nikkei 225 Index (N225), S&P 500 Index (SPX), FTSE 100 Index (UKX), Hang Seng Index (HSI), Mexico's IPC Index (MXX), and CAC 40 (FCHI). Data for the daily closing prices of these indices, denominated in USD, is obtained from Yahoo Finance for the period from January 1, 2019, to December 31, 2022. Each index comprises a total of 1042 daily closing price data points. The model parameters are estimated using the maximum likelihood estimation method, with specific details of the estimation procedure available in the study by Fortune (1999) [82]. The obtained results are reported in Table 4 and serve as the basis for the subsequent numerical experiments.

Table 4. Parameter estimates for the jump-diffusion model.

Para.	SSECI	GDAXI	N225	SPX	FTSE	HIS	MXX	FCHI
μ	0.1612	0.2461	0.1242	0.3109	0.2286	0.0001	0.1193	0.2641
σ	0.1416	0.1439	0.1637	0.1479	0.1205	0.1704	0.1588	0.1477
λ	10.6196	27.0351	12.9691	18.3318	24.2630	33.1185	6.1999	16.7630
μ_J	−0.0085	−0.0062	−0.0034	−0.0091	−0.0078	−0.0004	−0.0190	−0.0090
σ_J	0.0278	0.0316	0.0303	0.0358	0.0273	0.0274	0.0217	0.0352

Our results are obtained through maximum likelihood estimation of the five parameters in a jump diffusion model for stock returns. These parameters include the instantaneous mean return (μ) and volatility (σ) of a simple diffusion process, as well as three parameters related to the jump process: the mean frequency of jump events (λ), the mean size of a jump's impact on returns (μ_J), and the standard deviation (σ_J), representing the volatility of a jump's effect on returns. Based on the findings presented in Table 4, we utilize Monte Carlo simulation to generate 5000 price paths for the index by Eq (2.9). These simulated price paths are then employed in our mean-CVaR portfolio optimization model.

5.2. *Constraint satisfaction*

In this study, based on the Lagrangian multiplier method, we can incorporate a penalty term into the objective function expression to handle the constraint of satisfying the expected return described by Eq (2.5). This allows us to obtain the fitness function associated with each individual of the six intelligent algorithms used, which can be described in Eq (5.1):

$$fitness(x) = \xi + (1-\alpha)^{-1} \sum_{j=1}^{J} p(y_j)[f(x, y_j) - \xi, 0]^{+} + \lambda_L |x'r - u|, \tag{5.1}$$

where $\lambda_L$is the Lagrange multiplier penalty coefficient, which is set to 10^5 in this study. In addition, the constraints described by Eq (2.6) are handled as follows:

$$x_i^* = \begin{cases} x_i/S, & \text{if } S = 0, \\ 1/N, & \text{else.} \end{cases}, i = 1, 2, \cdots, N \tag{5.2}$$

5.3. *Results and discussion*

To assess the effectiveness of the proposed CEBWO algorithm in addressing the mean-CVaR portfolio optimization model, we conducted a comparative analysis with five state-of-the-art intelligent algorithms, namely WOA, AVOA, HBA, AHA, and BWO, which were used previously. Additionally, we compared the obtained efficient frontier with the standard efficient frontier derived using the linear programming model approach proposed by Rockafellar and Uryasev (2000;2002) [7, 8]. The MATLAB platform was utilized for implementing all algorithms individually. The parameter configurations for each algorithm were kept consistent with the earlier specifications, and a termination condition of a maximum of 1000 iterations was set. The comparison was conducted for multiple scenarios with 5000 price paths, considering different confidence levels, such as 0.90 and 0.95. We generated a set of 50 expected returns for the portfolio, ranging from the minimum to the maximum returns of the eight assets under consideration.

Table 5 presents the mean and standard deviation (STD) of returns and CVaR values obtained by CEBWO and the other five algorithms in solving the mean-CVaR portfolio optimization model. The experiments were conducted independently 30 times, and the best values are highlighted in bold black. The performance evaluation of each algorithm focuses on its ability to provide an optimal solution for a specified expected return. Table 5 provides a comprehensive overview of the 50 expected returns for the portfolio and the corresponding results obtained by the six considered algorithms in solving the mean-CVaR portfolio optimization model at two confidence levels. The results obtained demonstrate the effectiveness and competitiveness of the proposed CEBWO algorithm in solving the mean-CVaR portfolio optimization problem.

Upon meticulous analysis of Table 5, it becomes evident that among the six intelligent optimization algorithms, namely AVOA, AHA, CEBWO, BWO, WOA, and HBA, only the optimal solutions obtained by AVOA, AHA, and CEBWO satisfy the expected return conditions when solving the mean-CVaR model at a confidence level of 0.90. Notably, the CEBWO algorithm exhibits the smallest mean CVaR values and outperforms the other algorithms, with AHA and AVOA following suit, and BWO, WOA, and HBA ranking lower in performance. Similarly, at a confidence level of 0.95, only the optimal investment portfolios derived from the AHA and CEBWO algorithms fulfill the expected return conditions, with the CEBWO algorithm once again demonstrating the best mean CVaR values. The performance of the remaining algorithms, ranked from best to worst, is as follows: AHA, BWO, AVOA, WOA, and HBA. These findings accentuate the efficient capabilities of the proposed CEBWO algorithm in effectively addressing the mean-CVaR portfolio optimization problem.

Table 5. Results for the mean-CVaR portfolio optimization model.

Confidence Level		α = 0.90		α = 0.95	
Method	Measure	Return	CVaR	Return	CVaR
WOA	Mean	2.7734E-04	1.8854E-04	2.7399E-04	2.7768E-04
	STD	2.2431E-07	6.2675E-06	3.4440E-07	1.5972E-05
AVOA	Mean	**2.7719E-04**	1.8599E-04	2.7383E-04	2.7033E-04
	STD	1.2423E-08	7.1977E-06	6.1384E-08	3.1695E-05
HBA	Mean	2.7811E-04	2.0144E-04	2.7472E-04	3.1432E-04
	STD	2.6294E-06	2.8314E-05	2.7491E-06	8.4756E-05
AHA	Mean	**2.7719E-04**	1.8256E-04	**2.7381E-04**	2.5903E-04
	STD	2.3748E-09	3.9733E-07	1.1921E-09	4.4382E-07
BWO	Mean	2.7721E-04	1.8365E-04	2.7389E-04	2.6378E-04
	STD	8.4814E-08	1.5059E-06	1.5642E-07	6.2909E-06
CEBWO	Mean	**2.7719E-04**	**1.8246E-04**	**2.7381E-04**	**2.5883E-04**
	STD	**6.2803E-12**	**3.4196E-07**	**7.1460E-11**	**4.3511E-07**

Figures 8 and 9 present the effective frontiers of the mean-CVaR portfolio optimization model computed by the six algorithms for two distinct confidence levels, denoted as $alpha$ = 0.90 and $alpha$ = 0.95, while employing a fixed number of scenarios (J=5000). The outcomes depicted in both figures clearly demonstrate the superior accuracy of the solutions derived from the proposed CEBWO algorithm. Across the two confidence levels, the efficient frontier curves generated by the CEBWO algorithm align more closely with the standard efficient frontier, which is obtained through the utilization of the linear programming model. The observations derived from Figures 8 and 9 furnish empirical evidence substantiating the efficacy of the proposed CEBWO algorithm in effectively addressing the mean-CVaR portfolio optimization problem.

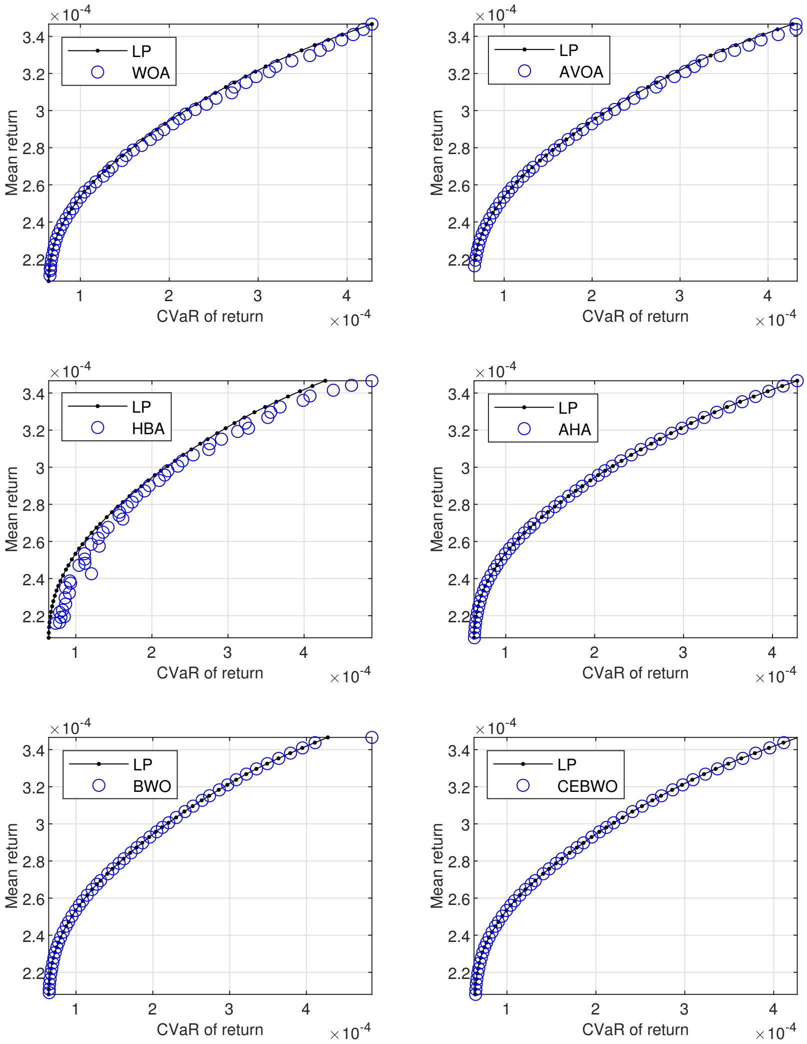

Figure 8. Comparison of effective frontiers obtained by six algorithms when $\alpha = 0.90$.

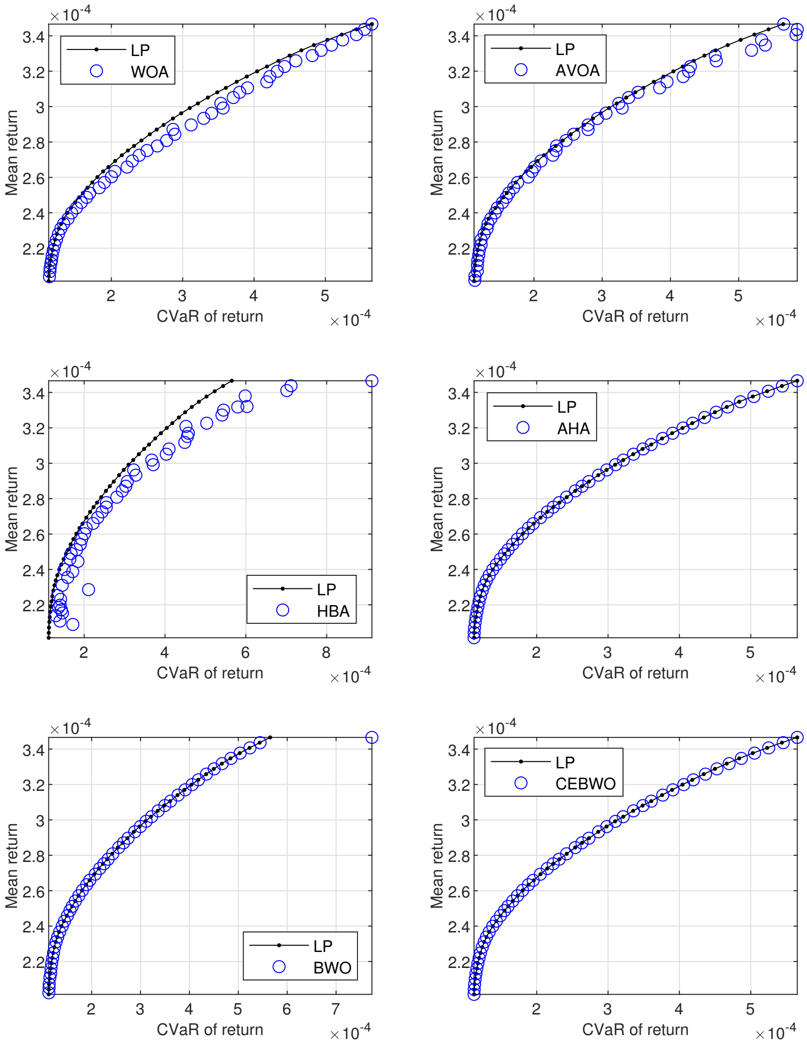

Figure 9. Comparison of effective frontiers obtained by six algorithms when $\alpha = 0.95$.

6. Conclusions

This paper presents a hybrid meta-heuristics algorithm that combines the cross-entropy method (CE) and the beluga whale optimization algorithm (BWO) for solving the mean-CVaR portfolio optimization model. The algorithm's performance is evaluated on 29 benchmark functions from CEC 2017, comparing it against several state-of-the-art intelligent algorithms, including WOA, AVOA, HBA, AHA, and BWO. Experimental results demonstrate the superior solution quality and convergence speed of the proposed hybrid method. To further evaluate the proposed method for solving the mean-CVaR portfolio optimization model, a real-world portfolio comprising eight equity indices from the global market is constructed. Monte Carlo simulation based on the jump-diffusion model is employed to generate scenario paths. The empirical results provide additional evidence of the effectiveness of the hybrid meta-heuristic algorithm for mean-CVaR portfolio selection. The proposed algorithm demonstrates both feasibility and effectiveness in solving the mean-CVaR portfolio optimization model, surpassing the accuracy of the other five intelligent algorithms. Moreover, it exhibits applicability to portfolio problems with diverse real-world trading conditions and other risk measures. These findings underscore the potential of the hybrid algorithm for practical applications in portfolio optimization.

Author contributions

Conceptualization, G.C.L.; methodology, G.C.L.; software, G.C.L.; validation, P.Z., and M.H.S.; formal analysis, P.Z.; investigation, M.H.S.; resources, M.H.S.; data curation, G.S.L.; writing—original draft preparation, G.C.L.; writing—review and editing, G.C.L., and G.S.L.; visualization, G.C.L. supervision, G.C.L.; project administration, G.C.L.; funding acquisition, G.C.L., and G.S.L.. All authors have read and agreed to the published version of the manuscript.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research was funded by the Humanities and Social Sciences Key Grant Foundation of the Education Department of Anhui Province of P.R. China (No. SK2021ZD0075, 2022AH040240), the Special Program for Science and Technology Innovation Strategy and Soft Science of Anhui Province (No. 202106f01050041), the Humanities and Social Sciences Foundation of Ministry of Education of P. R. China (No. 23YJA630143).

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. H. Markowitz, Portfolio selection, *J. Fin.*, **7** (1952), 77–91. https://doi.org/10.2307/2975974
2. P. Artzner, F. Delbaen, J. Eber, D. Heath, Coherent measures of risk, *Math. Financ.*, **9** (1999), 203–228. https://doi.org/10.1111/1467-9965.00068
3. J. Longerstaey, M. Spencer, *Riskmetricstm–Technical document*, 4 Eds., New York: Morgan Guaranty Trust Company of New York, 1996.
4. F. Y. Chen, Analytical VaR for international portfolios with common jumps, *Comput. Math. Appl.*, **62** (2011), 3066–3076. https://doi.org/10.1016/j.camwa.2011.08.018
5. J. W. Goh, K. G. Lim, M. Sim, W. Zhang, Portfolio value-at-risk optimization for asymmetrically distributed asset returns, *Eur. J. Oper. Res.*, **221** (2012), 397–406. https://doi.org/10.1016/j.ejor.2012.03.012

6. S. Basak, A. Shapiro, Value-at-risk based risk management: Optimal policies and asset prices, *Rev. Fin. Stud.*, **14** (2001), 371–405. https://doi.org/10.1093/rfs/14.2.371
7. T. Rockfeller, S. Uryasev, Optimization of conditional value-at-risk, *J. Risk*, **2** (2000), 21–41. https://doi.org/10.21314/JOR.2000.038
8. R. T. Rockfeller, S. Uryasev, Conditional value-at-risk for general loss distribution, *J. Bank. Financ.*, **26** (2002), 1443–1471. https://doi.org/10.1016/S0378-4266(02)00271-6
9. S. Alexander, T. F. Coleman, Y. Li, Minimizing CVaR and VaR for a portfolio of derivatives, *J. Bank. Finan.*, **30** (2006), 583–605. https://doi.org/10.1016/j.jbankfin.2005.04.012
10. S. Zhu, M. Fukushima, Worst-case conditional value-at-risk with application to robust portfolio management, *Oper. Res.*, **57** (2009), 1155–1168. https://doi.org/10.1287/opre.1080.0684
11. S. Yau, R. H. Kwon, J. S. Rogers, D. Wu, Financial and operational decisions in the electricity sector: Contract portfolio optimization with the conditional value-at-risk criterion, *Int. J. Prod. Econ.*, **134** (2011), 67–77. https://doi.org/10.1016/j.ijpe.2010.10.007
12. L. J. Hong, G. Liu, Simulating sensitivities of conditional value at risk, *Manage. Sci.*, **55** (2009), 281–293. https://doi.org/10.1287/mnsc.1080.0901
13. S. Zhao, Q. Lu, L. Han, Y. Liu, F. Hu, A mean-CVaR-skewness portfolio optimization model based on asymmetric Laplace distribution, *Ann. Oper. Res.*, **226** (2015), 727–739. https://doi.org/10.1007/s10479-014-1654-y
14. F. G. Ferreira, R. T. Cardoso, Mean-CVaR portfolio optimization approaches with variable cardinality constraint and rebalancing process, *Arch. Comput. Method. Eng.*, **28** (2021), 3703–3720. https://doi.org/10.1007/s11831-020-09522-1
15. C. I. Fábián, Handling CVaR objectives and constraints in two-stage stochastic models, *Eur. J. Oper. Res.*, **191** (2008), 888–911. https://doi.org/10.1016/j.ejor.2007.02.052
16. W. Liu, L. Yang, B. Yu, Kernel density estimation based distributionally robust mean-CVaR portfolio optimization, *J. Glob. Optim.*, **84** (2022), 1053–1077. https://doi.org/10.1007/s10898-022-01177-5
17. N. Abudurexiti, K. He, D. Hu, S. T. Rachev, H. Sayit, R. Sun, Portfolio analysis with mean-CVaR and mean-CVaR-skewness criteria based on mean–variance mixture models, *Ann. Oper. Res.*, **336** (2024), 945–966. https://doi.org/10.1007/s10479-023-05396-1
18. F. Q. Lu, M. Huang, W. K. Ching, T. K. Siu, Credit portfolio management using two-level particle swarm optimization, *Inform. Sci.*, **237** (2013), 162–175. https://doi.org/10.1016/j.ins.2013.03.005
19. T. Zhang, Z. Liu, Fireworks algorithm for mean-VaR/CVaR models, *Physica A: Stat. Mech. Appl.*, **483** (2017), 1–8. https://doi.org/10.1016/j.physa.2017.04.036
20. J. Zhai, M. Bai, H. Wu, Mean-risk-skewness models for portfolio optimization based on uncertain measure, *Optimization*, **67** (2018), 701–714. https://doi.org/10.1080/02331934.2018.1426577
21. Y. Li, B. Zhou, Y. Tan, Portfolio optimization model with uncertain returns based on prospect theory, *Complex Intell. Syst.*, **8** (2022), 4529–4542. https://doi.org/10.1007/s40747-021-00493-9
22. F. Lu, T. Yan, H. Bi, M. Feng, S. Wang, M. Huang, A bilevel whale optimization algorithm for risk management scheduling of information technology projects considering outsourcing, *Knowl.-Based Syst.*, **235** (2022), 107600. https://doi.org/10.1016/j.knosys.2021.107600

23. J. Danane, M. Yavuz, M. Yıldız, Stochastic modeling of three-species prey–predator model driven by Lévy Jump with Mixed Holling-II and Beddington–DeAngelis functional responses, *Fractal Fract.*, **7** (2023), 751. https://doi.org/10.3390/fractalfract7100751
24. Y. Song, G. Zhao, B. Zhang, H. Chen, W. Deng, W. Deng, An enhanced distributed differential evolution algorithm for portfolio optimization problems, *Eng. Appl. Artif. Intel.*, **121** (2023), 106004. https://doi.org/10.1016/j.engappai.2023.106004
25. X. S. Yang, A. H. Gandomi, Bat algorithm: A novel approach for global engineering optimization, *Eng. Computation.*, **29** (2012), 464–483. https://doi.org/10.1108/02644401211235834
26. A. H. Gandomi, A. H. Alavi, Krill herd: A new bio-inspired optimization algorithm, *Commun. Nonlinear Sci.*, **12** (2012), 4831–4845. https://doi.org/ 10.1016/j.cnsns.2012.05.010
27. S. Mirjalili, S. M. Mirjalili, A. Lewis, Grey wolf optimizer, *Adv. Eng. Softw.*, **69** (2014), 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
28. A. Askarzadeh, A novel metaheuristic method for solving constrained engineering optimization problems: Crow search algorithm, *Comput. Struct.*, **169** (2016), 1–12. https://doi.org/10.1016/j.compstruc.2016.03.001
29. S. Mirjalili, A. Lewis, The whale optimization algorithm, *Adv. Eng. Softw.*, **95** (2016), 51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
30. S. Saremi, S. Mirjalili, A. Lewis, Grasshopper optimisation algorithm: Theory and application, *Adv. Eng. Softw.*, **105** (2017), 30–47. https://doi.org/10.1016/j.advengsoft.2017.01.004
31. G. Wang, Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems, *Memetic Comp.*, **10** (2018), 151–164. https://doi.org/10.1007/s12293-016-0212-3
32. N. A. Kallioras, N. D. Lagaros, D. N. Avtzis, Pity beetle algorithm–A new metaheuristic inspired by the behavior of bark beetles, *Adv. Eng. Softw.*, **147** (2018), 147–166. https://doi.org/10.1016/j.advengsoft.2018.04.007
33. A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, H. Chen, Harris hawks optimization: Algorithm and applications, *Future Gener. Comp. Syst.*, **97** (2019), 849–872. https://doi.org/10.1016/j.future.2019.02.028
34. M. Jain, V. Singh, A. Rani, A novel nature-inspired algorithm for optimization: Squirrel search algorithm, *Swarm Evol. Comput.*, **44** (2019), 148–175. https://doi.org/10.1016/j.swevo.2018.02.013
35. S. Arora, S. Singh, Butterfly optimization algorithm: A novel approach for global optimization, *Soft Comput.*, **23** (2019), 715–734. https://doi.org/10.1007/s00500-018-3102-4
36. A. Faramarzi, M. Heidarinejad, S. Mirjalili, A. H. Gandomi, Marine Predators Algorithm: A nature-inspired metaheuristic, *Expert Syst. Appl.*, **152** (2020), 113377. https://doi.org/10.1016/j.eswa.2020.113377
37. M. Khishe, M. R. Mosavi, Chimp optimization algorithm, *Expert Syst. Appl.*, **149** (2020), 113338. https://doi.org/10.1016/j.eswa.2020.113338
38. S. Li, H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Gener. Comp. Syst.*, **111** (2020), 300–323. https://doi.org/10.1016/j.future.2020.03.055

39. A. Mohammadi-Balani, M. D. Nayeri, A. Azar, M. Taghizadeh-Yazdi, Golden eagle optimizer: A nature-inspired metaheuristic algorithm, *Comput. Ind. Eng.*, **152** (2021), 107050. https://doi.org/10.1016/j.cie.2020.107050
40. D. Połap, M. Woźniak, Red fox optimization algorithm, *Expert Syst. Appl.*, **166** (2021), 114107. https://doi.org/10.1016/j.eswa.2020.114107
41. Y. Yang, H. Chen, A. A. Heidari, A. H. Gandomi, Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts, *Expert Syst. Appl.*, **177** (2021), 114864. https://doi.org/10.1016/j.eswa.2021.114864
42. I. Ahmadianfar, A. A. Heidari, A. H. Gandomi, X. Chu, H. Chen, RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method, *Expert Syst. Appl.*, **181** (2021), 115079. https://doi.org/10.1016/j.eswa.2021.115079
43. J. Tu, H. Chen, M. Wang, A. H. Gandomi, The colony predation algorithm, *J. Bionic Eng.*, **181** (2021), 674–710. https://doi.org/10.1007/s42235-021-0050-y
44. I. Ahmadianfar, A. A. Heidari, S. Noshadian, H. Chen, A. H Gandomi, INFO: An efficient optimization algorithm based on weighted mean of vectors, *Expert Syst. Appl.*, **195** (2021), 116516. https://doi.org/10.1016/j.eswa.2022.116516
45. B. Abdollahzadeh, F. S. Gharehchopogh, S. Mirjalili, African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems, *Comput. Ind. Eng.*, **158** (2021), 107408. https://doi.org/10.1016/j.cie.2021.107408
46. B. Abdollahzadeh, F. S. Gharehchopogh, S. Mirjalili, Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems, *Int. J. Intell. Syst.*, **36** (2021), 5887–5958. https://doi.org/10.1002/int.22535
47. F. A. Hashim, E. H. Houssein, K. Hussain, M. S. Mabrouk, W. Al-Atabany, Honey Badger Algorithm: New metaheuristic algorithm for solving optimization problems, *Math. Comput. Simulat.*, **192** (2022), 84–110. https://doi.org/10.1016/j.matcom.2021.08.013
48. W. Zhao, L. Wang, S. Mirjalili, Artificial hummingbird algorithm: A new bio-inspired optimizer with its engineering applications, *Comput. Method. Appl. Mech. Eng.*, **388** (2022), 114194. https://doi.org/10.1016/j.cma.2021.114194
49. B. Abdollahzadeh, F. S. Gharehchopogh, N. Khodadadi, S. Mirjalili, Mountain gazelle optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems, *Adv. Eng. Softw.*, **174** (2022), 103282. https://doi.org/10.1016/j.advengsoft.2022.103282
50. E. H. Houssein, D. Oliva, N. A. Samee, N. F. Mahmoud, M. M. Emam, Liver Cancer Algorithm: A novel bio-inspired optimizer, *Comput. Biol. Med.*, **165** (2023), 107389. https://doi.org/10.1016/j.compbiomed.2023.107389
51. H. Su, D. Zhao, A. A. Heidari, L. Liu, X. Zhang, M. Mafarja, et al., RIME: A physics-based optimization, *Neurocomputing*, **532** (2023), 183–214. https://doi.org/10.1016/j.neucom.2023.02.010
52. C. Zhong, G. Li, Z. Meng, Beluga whale optimization: A novel nature-inspired metaheuristic algorithm, *Knowl.-Based Syst.*, **251** (2022), 109215. https://doi.org/10.1016/j.knosys.2022.109215
53. D. H. Wolpert, W. G. Macready, No free lunch theorems for optimization, *IEEE T. Evolut. Comput.*, **1** (1997), 67–82. https://doi.org/10.1109/4235.585893

54. M. M. Mafarja, S. Mirjalili, Hybrid whale optimization algorithm with simulated annealing for feature selection, *Neurocomputing*, **260** (2017), 302–312. https://doi.org/10.1016/j.neucom.2017.04.053
55. M. Abdel-Basset, W. Ding, D. El-Shahat, A hybrid Harris Hawks optimization algorithm with simulated annealing for feature selection, *Artif Intell. Rev.*, **54** (2021), 593–637. https://doi.org/10.1007/s10462-020-09860-3
56. P. J. Gaidhane, M. J. Nigam, A hybrid grey wolf optimizer and artificial bee colony algorithm for enhancing the performance of complex systems, *J. Comput. Sci.*, **27** (2018), 284–302. https://doi.org/10.1016/j.jocs.2018.06.008
57. B. Farnad, A. Jafarian, D. Baleanu, A new hybrid algorithm for continuous optimization problem, *Appl. Math. Model.*, **55** (2018), 652–673. https://doi.org/10.1016/j.apm.2017.10.001
58. M. AkbaiZadeh, T. Niknam, A. Kavousi-Fard, Adaptive robust optimization for the energy management of the grid-connected energy hubs based on hybrid meta-heuristic algorithm, *Energy*, **235** (2021), 121171. https://doi.org/10.1016/j.energy.2021.121171
59. A. A. Najafi, S. Mushakhian, Multi-stage stochastic mean–semivariance–CVaR portfolio optimization under transaction costs, *Appl. Math. Comput.*, **256** (2015), 445–458. https://doi.org/10.1016/j.amc.2015.01.050
60. M. F. Leung, J. Wang, Cardinality-constrained portfolio selection based on collaborative neurodynamic optimization, *Neural Networks*, **145** (2022), 68–79. https://doi.org/10.1016/j.neunet.2021.10.007
61. H. Sorensen, Parametric inference for diffusion processes observed at discrete points in time: A survey, *Int. Stat. Rev.*, **72** (2004), 337–354. https://doi.org/10.1111/j.1751-5823.2004.tb00241.x
62. R. Cont, P. Tankov, *Financial modelling with jump processes*, 1 Eds., New York: Chapman and Hall/CRC, 2003. https://doi.org/10.1201/9780203485217
63. D. Ardia, J. David, O. Arango, N. D. G. Gómez, Jump-diffusion calibration using differential evolution, *Wilmott*, **55** (2011), 76–79. https://doi.org/10.1002/wilm.10034
64. R. Y. Rubinstein, Optimization of computer simulation models with rare events, *Eur. J. Oper. Res.*, **99** (1997), 89–112. https://doi.org/10.1016/S0377-2217(96)00385-2
65. P. T. de Boer, D. P. Kroese, R. Y. Rubinstein, A fast cross-entropy method for estimating buffer overflows in queueing networks, *Manage. Sci.*, **50** (2004), 883–895. https://doi.org/10.1287/mnsc.1030.0139
66. J. C. Chan, D. P. Kroese, Improved cross-entropy method for estimation, *Stat. Comput.*, **22** (2012), 1031–1040. https://doi.org/10.1007/s11222-011-9275-7
67. G. Alon, D. P. Kroese, T. Raviv, R. Y. Rubinstein, Application of the cross-entropy method to the buffer allocation problem in a simulation-based environment, *Ann. Oper. Res.*, **134** (2005), 137–151. https://doi.org/10.1007/s10479-005-5728-8
68. R. Caballero, A. G. Hernández-Díaz, M. Laguna, J. Molina, Cross entropy for multiobjective combinatorial optimization problems with linear relaxations, *Eur. J. Oper. Res.*, **243** (2015), 362–368. https://doi.org/10.1016/j.ejor.2014.07.046
69. R. Rubinstein, The cross-entropy method for combinatorial and continuous optimization, *Methodol. Comput. Appl.*, **1** (1999), 127–190. https://doi.org/10.1023/A:1010091220143

70. D. P. Kroese, S. Porotsky, R. Y. Rubinstein, The cross-entropy method for continuous multi-extremal optimization, *Methodol. Comput. Appl.*, **8** (2006), 383–407. https://doi.org/10.1007/s11009-006-9753-0
71. J. Bekker, C. Aldrich, The cross-entropy method in multi-objective optimisation: An assessment, *Eur. J. Oper. Res.*, **211** (2011), 112–121. https://doi.org/10.1016/j.ejor.2010.10.028
72. K. Chepuri, T. Homem-De-Mello, Solving the vehicle routing problem with stochastic demands using the cross-entropy method, *Ann. Oper. Res.*, **134** (2005), 153–181. https://doi.org/10.1007/s10479-005-5729-7
73. I. Szita, A. Lörincz, Learning Tetris using the noisy cross-entropy method, *Neural Comput.*, **18** (2006), 2936–2941. https://doi.org/10.1162/neco.2006.18.12.2936
74. M. Laguna, A. Duarte, R. Marti, Hybridizing the cross-entropy method: An application to the max-cut problem, *Comput. Oper. Res.*, **36** (2009), 487–498. https://doi.org/10.1016/j.cor.2007.10.001
75. M. Maher, R. Liu, D. Ngoduy, Signal optimization using the cross entropy method, *Transport. Res. C: Emer. Technol.*, **27** (2013), 76–88. https://doi.org/10.1016/j.trc.2011.05.018
76. G. R. Lamonica, M. C. Recchioni, F. M. Chelli, L. Salvati, The efficiency of the cross-entropy method when estimating the technical coefficients of input-output tables, *Spat. Econ. Anal.*, **15** (2020), 62–91. https://doi.org/10.1080/17421772.2019.1615634
77. M. L. Cardoso, L. F. Venturini, Y. L. Baracy, I. M. B. Ulisses, L. E. Bremermann, A. P. Grilo-Pavani, et al., Fault indicator placement optimization using the cross-entropy method and traffic simulation data, *Electr. Pow. Syst. Res.*, **212** (2022), 108391. https://doi.org/10.1016/j.epsr.2022.108391
78. A. E. Eiben, C. A. Schipper, On evolutionary exploration and exploitation, *Fund. Inform.*, **35** (1998), 35–50. https://doi.org/10.3233/FI-1998-35123403
79. H. Chen, Z. Wang, D. Wu, H. Jia, C. Wen, H. Rao, et al., An improved multi-strategy beluga whale optimization for global optimization problems, *Math. Biosci. Eng.*, **20** (2023), 13267–13317. https://doi.org/10.3934/mbe.2023592
80. A. G. Hussien, R. A. Khurma, A. Alzaqebah, M. Amin, F. A. Hashim, Novel memetic of beluga whale optimization with self-adaptive exploration–exploitation balance for global optimization and engineering problems, *Soft Comput.*, **27** (2023), 13951–13989. https://doi.org/10.1007/s00500-023-08468-3
81. N. H. Awad, M. Z. Ali, P. N. Suganthan, J. J. Liang, B. Y. Qu, *Problem defnitions and evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization*, 2016. Available from: `https://github.com/P-N-Suganthan/CEC2017-BoundContrained/blob/master/Definitions%20of%20%20CEC2017%20benchmark%20suite%20final%20version%20updated.pdf`.
82. P. Fortune, Are stock returns different over weekends? A jump diffusion analysis of the weekend effect, In: *New England Economic Review*, **10** (1999), 3–19. `https://fedinprint.org/item/fedbne/41360`

AIMS Press