Artificial Intelligence for Engineering Design, Analysis and Manufacturing

www.cambridge.org/aie

CNN–NSDBO–EWTOPSIS: A hybrid multi-objective optimization approach for concrete mixture proportion design problem

Qifang Luo[1,2], Jiang Wu[1], Yongquan Zhou[1,2,3,4] and Yuanfei Wei[3,4]

[1]College of Artificial Intelligence, Guangxi Minzu University, Nanning, China; [2]Guangxi Key Laboratories of Hybrid Computation and IC Design Analysis, Nanning, China; [3]Xiangsihu College of Guangxi Minzu University, Nanning, China and [4]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, Bangi, Malaysia

Research Article

Cite this article: Luo Q, Wu J, Zhou Y and Wei Y (2025). CNN–NSDBO–EWTOPSIS: A hybrid multi-objective optimization approach for concrete mixture proportion design problem. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, **39**, e29, 1–22
https://doi.org/10.1017/S0890060425100115

Received: 08 November 2024
Revised: 12 May 2025
Accepted: 02 July 2025

Keywords:
CNN; NSDBO; EWTOPSIS; concrete mixture proportion; multi-objective optimization; metaheuristic

Corresponding author:
Yongquan Zhou;
Email: yongquanzhou@126.com

Abstract

The conventional design method for high-performance concrete (HPC) mixture proportion requires a large amount of trial mixing work to obtain the desired HPC mixture proportion, which consumes a lot of manpower, material resources, and time resources during the trial mixing process. In recent years, an intelligent scheme for HPC mixture proportion design has been developed. To more effectively optimize HPC mixture proportions, this article proposes a novel intelligent HPC mixture proportion design method. Firstly, this article establishes a hybrid multi-objective optimization (MOO) method for HPC mixture proportion design problem, called CNN–NSDBO–EWTOPSIS. In this MOO framework, there are three objective functions, namely the compressive strength (CS) of concrete, cost, and carbon dioxide emissions. Among them, based on the various components of concrete, this article constructs a convolutional neural network (CNN) regression prediction model for predicting the CS of concrete. The calculation of cost and carbon dioxide emissions involves the utilization of two polynomials. Additionally, dung beetle optimizer (DBO) is used to optimize the hyperparameters of the CNN. Furthermore, this article incorporates the constructed CNN regression prediction model and two polynomials as the three objective functions for HPC mixture proportion design problem. This three-objective optimization problem is solved using a non-dominated sorting dung beetle optimizer (NSDBO). Finally, based on the obtained Pareto front, this article obtains a good solution using the entropy weight technique for order preference by similarity to an ideal solution (EWTOPSIS) method. The experimental results indicate that the proposed CNN–NSDBO–EWTOPSIS approach can achieve HPC mixture proportion design.

Introduction

Background of HPC mixture proportion problem

Concrete, as an indispensable building material in construction engineering, has a significant impact on ensuring the stability, durability, and overall quality of structures. With the continuous advancement in construction technology, the demand for high-strength, durable, and weather-resistant concrete has surged, especially in projects like bridges, dams, and high-rise buildings. Various factors influence concrete performance, with the mix ratio standing out as a key determinant of its mechanical properties, durability, and constructability. Rational mix design can optimize concrete performance to its fullest potential, thereby enhancing overall project quality. Traditionally, concrete mix design heavily relied on empirical knowledge and test data, where engineers adjusted and refined mixtures based on past experiences and test results. Despite its simplicity, this method has limitations and often fails to fully exploit the inherent potential of concrete materials, falling short of meeting the evolving demands of engineering projects. Furthermore, traditional approaches may result in the overuse of components like cement, leading to increased project costs and environmental impact.

With the continuous advancement in artificial intelligence and machine learning technologies, modern concrete mix design methods have garnered attention. Leveraging big data and sophisticated algorithms, these methods achieve intelligent optimization of concrete mix proportions by conducting in-depth analysis and modeling of material properties and engineering requirements. Machine learning algorithms such as support vector machine (SVM) (Suthaharan and Suthaharan, 2016), random forest (RF) (Belgiu and Drăguţ, 2016), and back-propagation neural network (BPNN) (Goh, 1995) are extensively employed in concrete mix design, substantially enhancing design accuracy and efficiency. Modern concrete mix design methods not only meet engineering requirements but also effectively mitigate project costs and environmental impacts. By utilizing machine learning models to accurately forecast material properties and integrating optimization algorithms to intelligently adjust mix proportions, concrete performance can be maximized. This approach minimizes the overconsumption of resources like cement, aligning with the principles of sustainable construction.

© The Author(s), 2025. Published by Cambridge University Press. This is an Open Access article, distributed under the terms of the Creative Commons Attribution licence (http://creativecommons.org/licenses/by/4.0), which permits unrestricted re-use, distribution and reproduction, provided the original article is properly cited.

Metaheuristic optimization algorithm

Although the methods mentioned above can greatly improve the accuracy of predicting the compressive strength (CS) of high-performance concrete (HPC), relying on a single model to solve this problem still has limitations. Currently, a popular contemporary approach is to combine neural networks with intelligent optimization algorithms. Intelligent optimization algorithms are efficient tools for tackling complex optimization problems. They are typically flexible and versatile, capable of finding near-optimal solutions across a variety of problem scenarios. Unlike traditional optimization methods such as linear programming (Dantzig, 2002) and integer programming (Vinod, 1969), intelligent optimization algorithms do not depend on the specific characteristics of a problem. Instead, they guide the search process heuristically, balancing global exploration and local exploitation to find high-quality solutions within an acceptable time. Intelligent optimization algorithms can be mainly divided into four categories: evolutionary-based, physics- and chemistry-based, swarm intelligence-based, and human behavior-based. Classic intelligent optimization algorithms include genetic algorithm (GA) (Holland, 1992), particle swarm optimization (PSO) (Kennedy and Eberhart, 1995), simulated annealing (SA) (Bertsimas and Tsitsiklis, 1993), ant colony optimizer (ACO) (Dorigo and Di Caro, 1999), and differential evolution (DE) (Pant et al., 2020). In recent years, some new metaheuristic optimization algorithms have been proposed, such as whale optimization algorithm (WOA) (Mirjalili and Lewis, 2016), grey wolf optimizer (GWO) (Mirjalili et al., 2014), sparrow search algorithm (Xue and Shen, 2020), and dung beetle optimizer (DBO) (Xue and Shen, 2023). These algorithms have demonstrated strong performance across various application fields, but the no free lunch theorem (NFL) reminds us that no single algorithm can effectively solve all problems. Therefore, continuous improvement and innovation in algorithms are crucial. Feng et al. (2024) proposed the first complex-valued version of the artificial hummingbird algorithm (CAHA) and optimized the parameters of the artificial neural network (ANN) using CAHA for the short-term wind speed prediction problem. Bui et al. (2018) built an expert system based on an ANN model combined with an improved firefly algorithm to predict the performance of HPC. Jangir and Jangir (2017) developed a multi-objective variant of the WOA and utilized it to address a range of standard problems, including unconstrained, constrained, and engineering design challenges. Mirjalili et al. (2017) proposed the multi-objective ant lion optimization algorithm (MOALO) and utilized it to address various multi-objective engineering design problems. Zhang et al. (2023) proposed a multi-objective bald eagle search algorithm (MOBES) and applied it to solve two-objective, three-objective, and four-objective engineering design problems in the real world. Jangir et al. (2024) proposed a differential evolution algorithm based on depth information for the complex nonlinear optimization problem of proton exchange membrane fuel cell (PEMFC) parameter estimation. The optimized parameters of the proposed method resulted in a sum of squared errors (SSE) as low as 0.00002 in some cases, indicating better accuracy and stability. In order to optimize the parameter estimation of PEMFC, Jangir et al. (2024) used an advanced version of artificial rabbit optimization called mutated northern goshawk and elite opposition learning-based artificial rabbit optimizer (MNEARO). The experimental results showed that MNEARO outperformed other methods in terms of computational cost and solution quality. Jangir et al. (2024) proposed an artificial hummingbird algorithm based on Levy chaotic horizontal vertical crossing for the accurate estimation of PEMFC parameters. The experimental results showed that the combination of this method with PEMFC parameters significantly improved performance. Jangir et al. (2024) proposed a collaborative strategy-based differential evolution (CS-DE) algorithm for robust PEMFC parameter estimation, and experimental results showed the robustness and adaptability of CS-DE to complex PEMFC modeling tasks. Agrawal et al. (2024) proposed a quick crisscross sine cosine algorithm (QCSCA) to address the optimal power flow (OPF) problem in power systems that integrate renewable energy and flexible AC transmission system (FACTS) equipment. Experimental results showed that QCSCA outperformed various Sine Cosine Algorithm (SCA) variants, consistently minimizing generation costs, power losses, and total costs. This study focuses on the recently popular algorithm DBO and HPC mixture proportion design problem. Eventually, the main contributions and innovation of this paper are summarized as follows:

1. We utilized a convolutional neural network (CNN) to develop a model predicting the CS of concrete.
2. We optimized CNN hyperparameters through DBO and selected the optimal model as one of the objective functions in the HPC mixture proportion design problem.
3. We employed the non-dominated sorting dung beetle optimizer (NSDBO) algorithm to obtain the Pareto solution set and Pareto front (PF) for a three-objective optimization problem.
4. We utilized the entropy weight technique for order preference by similarity to an ideal solution (TOPSIS) multi-objective optimization (MOO) analysis method to comprehensively evaluate the obtained PF and perform decision analysis.
5. We proposed a hybrid MOO method, called CNN–NSDBO–EWTOPSIS.
6. We employed CNN–NSDBO–EWTOPSIS to solve the three-objective optimization problem of HPC mixture proportion design.

The remaining part of this paper is organized as follows: In Section "Literature review", an overview of pertinent studies on concrete mix design and relevant literature on DBO is provided. Section "Preliminaries" introduces the theoretical basis of the methods used in this article. Section "Methodology" provides a detailed description of the establishment of the proposed hybrid CNN–NSDBO–EWTOPSIS framework, while Section "Experimental result and analysis" focuses on an experiment to validate the effectiveness of the proposed model. Finally, Section "Conclusion and future work" presents the conclusion of this paper and proposes the following advice and potential directions for future work.

Literature review

Research on HPC mixture proportion design

HPC, known for its excellent mechanical properties and durability, has become a widely used material in modern construction engineering. The proper mixture proportion design of concrete is crucial to ensuring the performance of HPC. Traditional mixture proportion design methods often rely on experience and experimental data, leading to inefficiencies and high costs. With the rapid advancement of artificial intelligence and machine learning technologies, data-driven optimization methods have introduced new approaches and tools for the mixture proportion design of HPC. Chen et al. (2023) proposed a MOO hybrid intelligent framework combining RF and non-dominated sorting genetic algorithm version II (NSGA-II) to effectively predict concrete durability and optimize concrete mixture proportions. They applied this proposed method to a practical highway project. Liu et al. (2023) proposed a method for optimizing

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

the mixture proportion of recycled aggregate concrete (RAC) using machine learning and metaheuristic techniques. They developed six machine learning models to predict the CS of RAC. Based on the best predictive model, they employed a multi-objective PSO algorithm with a competition mechanism to optimize three scenarios involving four objective functions of RAC: CS, material cost, carbon footprint, and energy intensity. Tipu et al. (2023) aimed to maximize CS while minimizing costs and carbon dioxide emissions. They proposed using the XGBoost model and NSGA-II to achieve these objectives. The XGBoost model was employed to predict the CS of concrete, and NSGA-II was utilized for three-objective optimization. Zhang et al. (2020) proposed a MOO method combining machine learning and metaheuristic algorithms to optimize concrete mixture proportions. They utilized a multi-objective PSO algorithm to refine the mixing ratios and achieve optimal goals. Their experiments revealed that BPNNs perform better on continuous data (such as strength), while RF algorithms yield higher prediction accuracy for more discrete data (such as slump). Using their MOO model, they successfully obtained the PF for the MOO of both HPC and plastic concrete mixtures. Zhang et al. (2025) proposed an innovative method for mix proportion design and optimization. It pioneers the application of response surface methodology in the multi-parameter mix design and optimization of manufactured sand concrete, significantly enhancing the efficiency of the design process and providing more accurate solutions for the diversified production of manufactured sand concrete. Mandal and Rajput (2025) explored the application of machine learning techniques in optimizing ceramic waste concrete. They discussed a range of computational paradigms, including decision trees, RFs, XGBoost, ANNs, bagging, adaptive boosting (AdaBoost), gradient boosting, regression models, and SVMs. Zhang et al. (2025) developed an intelligent predictive model for the CS of RAC using machine learning techniques, based on 1,255 mix design datasets compiled from published literature and laboratory experiments. Furthermore, they established an intelligent mix proportioning model capable of generating optimal mixtures to meet specified CS targets. Wang et al. (2025) proposed a method based on the characterization of aggregate physical properties to address the varying requirements for CS, flexural strength, and cost-effectiveness in different types of RAC. Neural networks were employed to capture the complex nonlinear relationships between input parameters and target performance indicators. The algorithm with the highest fitting accuracy was selected as the objective function and integrated with NSGA-II to obtain the PF. Finally, the ideal point method was applied to identify Pareto-optimal solutions, enabling the selection of optimal mix proportion schemes tailored to different performance preferences. Concrete CS is a critical objective in the concrete mixture design process, and it is a primary focus of our attention. Asteris and Kolovos (2019) researched the prediction of CS in self-compacting concrete using ANNs. Kumar and Pratap (2024) investigated the use of machine learning methods to accurately predict the CS of high-strength concrete. Feng et al. (2020) proposed an intelligent prediction method for concrete CS based on machine learning techniques. This method utilizes the AdaBoost algorithm. Naderpour et al. (2018) utilized ANNs to predict the CS of RAC. Hariri-Ardebili et al. (2024) applied the principle of AutoML solutions to predict the most important mechanical property of various concrete datasets, namely CS, for benchmark testing.

Research on dung beetle optimizer

Since its proposal at the end of 2022, DBO has attracted widespread attention from many scholars due to its fast convergence speed, high solving accuracy, and strong optimization ability. So far, there have been some improvements and applications. Li et al. (2024) proposed an improved DBO to optimize the parameters of the BiLSTM model, thereby improving the predictive performance of the dual-optimization wind speed prediction model. Zhu et al. (2024) introduced a quantum-inspired hybrid dung beetle optimizer (QHDBO), which combines quantum computing techniques with multiple strategy integration, effectively applying it to address engineering challenges. (He et al. (2024) developed an improved dung beetle optimizer (IDBO) and utilized it to optimize the parameters of variation mode decomposition. Jiachen and Li-hui (2024) proposed an IDBO combined with the dynamic window approach (DWA) for path planning problems in both static and dynamic environments. (Cai et al. (2024) combined an IDBO with two deep learning models for predicting groundwater depth. To enhance the accuracy of specific dynamic subject recognition, Li et al. (2024) proposed a model that integrates an IDBO to optimize a long short-term memory (LSTM) network for specific dynamic subject identification. Wang et al. (2023) introduced a quasi-adversarial learning-based DBO (QOLDBO) that integrates Q-learning, applying it effectively to address classical engineering problems. Mai et al. (2024) introduced a novel maximum power point tracking (MPPT) technology for photovoltaic systems, utilizing the DBO to maximize output power across diverse weather conditions. Tu et al. (2024) designed a Q-learning-based multi-swarm beetle optimizer (MODBO-QL) to solve the optimal scheduling model for Hybrid Integrated Energy System (HIES). During the machining process, parts are prone to generating higher surface residual stresses in the cutting direction (CD), thereby reducing production costs. To address this issue, Xue et al. (2023) employed a MOO method that combines DBO–BPNN and improved particle swarm optimization (IPSO) algorithm. Wen-Chao et al. (2023) proposed a method combining variation mode decomposition (VMD) and gated recursive unit (GRU) in DBO, where DBO is used to optimize the parameters of GRU. Zhang et al. (2024) improved the hybrid kernel extreme learning machine model on the basis of the DBO to measure and analyze the Water-energy-food nexus (WEFN) recovery ability of China's Beidahuang Group. Table 1 lists additional studies on DBO covering various improvements and widespread applications, demonstrating the significant potential of DBO.

Preliminaries

CNN multi-input regression prediction

CNNs not only excel in image processing, but also demonstrate powerful capabilities in multi-input regression prediction tasks. Multi-input regression prediction involves gathering data from multiple input sources to predict one or more continuous output variables. This approach is particularly suitable for modeling complex systems, such as forecasting the CS of concrete, as it can manage the nonlinear relationships between multiple input variables.

By employing a multi-input CNN model, various types of input data (such as material composition, environmental conditions, etc.) can be integrated and analyzed. Convolution layers automatically extract features from these inputs, while pooling layers reduce dimension and prevent overfitting. Finally, the extracted features are transformed into the final predicted values through fully connected layers. This method effectively captures the complex characteristics of the input data, thereby enhancing the accuracy and robustness of the predictions.

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

Table 1. Literature survey on the DBO

References	Method	Application
Wei et al. (2023)	SA-based hybrid dung beetle optimizer (SDBO) algorithm	Seismic wave inversion and imaging endeavors
Xu et al. (2024)	DVMD–DBO–LSSVM–EC	Monthly runoff prediction
Zhang et al. (2024)	An improved dung beetle optimizer (IDBO) and a deep reinforcement learning-based twin delayed deep deterministic policy gradient (TD3) algorithm	A novel edge offloading system based on unmanned aerial vehicles (UAVs) and mobile edge service vehicles (MESVs)
Huang et al. (2024)	CMDBO–TSVM	Parameter selection of TSVM
Liu and Jiang (2024)	IDBO	Parameter optimization of the improved complete ensemble empirical modal decomposition with adaptive noise (ICEEMDAN)
Zhao et al. (2023)	DBO	Parameter optimization of SVM, RF, and KELM
Jiang et al. (2023)	DBO and SVR	A quantitative method for determining fatty acids in stored wheat using an olfactory sensor technique
Bai et al. (2023)	WPD and IDBO–KELM	Traditional diagnostic models for laser gyroscopes
He et al. (2024)	MIDBO	Optimal scheduling of flood control in reservoir groups
Yuan et al. (2023)	DBO–CNN	Water-body detection in synthetic aperture radar (SAR) images
Fu et al. (2024)	VMD–PE–IDBO–TCN	State of health (SOH) in lithium-ion batteries
Li et al. (2024)	A cascaded extended state observer sliding mode control (CE-SMC) method based on the osprey-dung beetle fusion optimization algorithm	Wind-thermal coordinated system

Related research has demonstrated that using CNNs for multi-input regression prediction has broad application prospects in the fields of engineering and science. For instance, Pourdaryaei et al. (2024) proposed a suitable technique for forecasting electricity prices using a multi-head self-attention and CNN-based approach. Similarly, Buchanan and Crawford (2024) proposed a model that combines feature recognition and many-to-one mapping capabilities of classical CNN with the probability characteristics of Gaussian process regression (GPR) for predicting battery health status. Wan et al. (2024) explored the application of CNN in predicting the partitioned homogeneous properties (PHPs) of electronic product wiring structures and found that CNN models can accurately predict the performance of previously unpaired wiring structures, which can be directly applied to product-level reliability finite element analysis (FEA) and improve the efficiency of reliability assessment. Bhadra et al. (2024) developed an end-to-end 3D CNN model for predicting soybean yield using RGB images based on multi-temporal drones and approximately 30,000 sample plots.

Dung beetle optimizer

Based on the distinct social roles among dung beetles, the DBO categorizes the population into four groups: ball-rolling dung beetles, spawning dung beetles, small dung beetles, and thief dung beetles. The dung beetles' location is represented as follows:

$$X = \{X_i | i = 1, 2, \ldots, N_{all}\} \tag{1}$$

For different purposes, the positions of the four dung beetle subpopulations are represented by distinct symbols as follows:

$$R = \{R_e | e = 1, 2, \ldots, N_{roll}\} \tag{2}$$

$$S = \{S_m | m = 1, 2, \ldots, N_{spawn}\} \tag{3}$$

$$L = \{L_h | h = 1, 2, \ldots, N_{little}\} \tag{4}$$

$$T = \{T_z | z = 1, 2, \ldots, N_{thief}\} \tag{5}$$

$$X = R \cup S \cup L \cup T \tag{6}$$

$$N_{roll} + N_{spawn} + N_{little} + N_{thief} = N_{all} \tag{7}$$

If the dimension of the optimization problem is D and the corresponding fitness function is f, then the position of each beetle is represented as a solution to the problem as follows:

$$X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,D}\} \tag{8}$$

The individual fitness value is represented as $f(X_i)$. The representations of R_e, S_m, L_h, T_z, and their corresponding fitness values are similar to X_i. The algorithm updates the positions of dung beetles according to different strategies, evaluates their survival ability based on fitness, and assumes that a smaller fitness value indicates better optimization results. The mathematical expressions for the optimal and worst positions are as follows:

$$\begin{cases} X^b = \{X_i \in X, i = 1, 2, \ldots, N_{all} | \forall X_j, f(X_i) \le f(X_j)\} \\ X^w = \{X_i \in X, i = 1, 2, .., N_{all} | \forall X_j, f(X_j) \le f(X_i)\} \end{cases} \tag{9}$$

The final optimal position X^b represents the best solution to the problem.

(1) Update the position of the roller-ball dung beetles

The position update for roller-ball dung beetles is divided into two scenarios: obstacle-free and obstacle-present.

When $\delta < \gamma$, roller-ball dung beetles are in an obstacle-free state, where δ is a random number and $\delta \in [0, 1]$, $\gamma = 0.9$. The position update formula is as follows:

$$R_{new,e}^{t+1} = R_e^t + \alpha \times k \times R_e^{t-1} + u \times \Delta x \tag{10}$$

$$\Delta x = \left| R_e^t - X^w \right| \tag{11}$$

Here, t represents the current iteration number, and R_e^t is the e-th dung beetle's position after the t-th iteration. k is a constant and $k \in (0, 0.2]$ represents the positional deviation coefficient. α is a

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

natural coefficient used to model the impact of environmental factors on the direction of movement. When $\lambda < \eta$, α is 1, otherwise α is -1, where λ is a random number and $\lambda \in [0,1]$, $\eta = 0.1$, u is a constant, and $u \in [0,1]$. X^W represent the worst global position, while Δx is used to simulate changes in lighting intensity.

When $\delta \geq \gamma$ is in an obstacle state, the position is updated as follows:

$$R_{new,e}^{t+1} = R_e^t + \tan(\theta)\left|R_e^t - R_e^{t-1}\right| \tag{12}$$

where θ is the deflection angle represented by radians, θ is a random number, and $\theta \in [0,\pi]$. If θ is $0, \pi/2, \pi$, the position is not updated. In the above position update formula, $R_{new,e}^{t+1}$ is the candidate position obtained from the individual's $t+1$ iteration, which will be compared with the previously best-known optimal position and retained; that is, if $f\left(R_{new,e}^{t+1}\right) > f\left(R_e^t\right)$, $R_e^{t+1} = R_e^t$, otherwise $R_e^{t+1} = R_{new,e}^{t+1}$.

(2) Update the position of the spawning dung beetles

Some of the dung balls collected by the beetles are used as food, while the rest are transported to a secure location for spawning, where they serve as brooding balls to nurture the next generation. The boundaries of the area where the brooding balls are located are strictly limited in Eq. (13):

$$\begin{aligned} Lb^* &= Max\left(X^{b*} \times (1-Q), Lb\right) \\ Ub^* &= Min\left(X^{b*} \times (1-Q), Ub\right) \end{aligned} \tag{13}$$

where Ub^* and Lb^* represent the upper and lower boundaries of the spawning area, respectively. X^{b*} represents the current best position of all beetles in the population, $Q = 1 - t/T$, T is the maximum iteration number, and Ub and Lb are the upper and lower boundaries, respectively.

After transporting the brooding ball to the designated spawning area, spawning dung beetles will lay one egg inside it during each iteration. The brooding ball's position is then updated as follows:

$$B_{new,m}^{t+1} = X^{b*} + a_1 \times \left(B_m^t - Lb^*\right) + a_2 \times \left(B_m^t - Ub^*\right) \tag{14}$$

where B_m^t represents the m-th brooding ball's position after the t-th iteration, a_1 and a_2 are two random vectors of size $1 \times D$, and D is the dimension.

(3) Update the position of the little dung beetles

Once the juvenile dung beetles inside the brooding ball reach maturity, they emerge from it to forage for food. Therefore, to achieve the goal of space exploration, it is essential to define an optimal foraging area to guide their search. The boundary of this optimal foraging area is defined as follows:

$$\begin{aligned} Lb' &= Max\left(X^b \times (1-Q), Lb\right) \\ Ub' &= Min\left(X^b \times (1+Q), Ub\right) \end{aligned} \tag{15}$$

where X^b represents the global best position and Ub' and Lb' represent the upper and lower bounds of the optimal foraging area, respectively. The definitions of Q, Lb, and Ub are the same as in Eq. (13). The formula for updating the little dung beetles' position is as follows:

$$L_{new,h}^{t+1} = L_h^t + C_1 \times \left(L_h^t - Lb'\right) + C_2 \times \left(L_h^t - Ub'\right) \tag{16}$$

where L_h^t represents the h-th little beetle's position after the t-th iteration, C_1 is a random number that follows a normal distribution, $C_1 \in [0,1]$, and C_2 is a random vector, with each component ranging between 0 and 1.

(4) Update the position of the thief dung beetles

Not all dung beetles will actively push the ball; some will instead steal the dung balls collected by others. The formula for updating the thief dung beetles' position is as follows:

$$T_{new,z}^{t+1} = X^b + S \times g \times \left(\left|T_z^t - X^{b*}\right| + \left|T_z^t - X^b\right|\right) \tag{17}$$

where X^b represents the global best position and X^{b*} holds the same meaning as defined in Eq. (13). T_z^t denotes the z-th thief dung beetle's position after the t-th iteration, while S is a constant set to 0.5. Finally, g is a D dimensional random row vector, with each component between 0 and 1.

The flowchart of DBO is shown in Figure 1, and the pseudo-code of DBO is shown in Algorithm 1.

Algorithm 1 The pseudo-code of DBO algorithm.

Require: The maximum iterations T, the size of the population N_{all}.
Ensure: Optimal position X^b and its fitness value F_b.

```
Initialize the particle's population i =1, 2, …, N_all and define its relevant parameters.
while (t < = T ) do.
for i = 1 to N_all do.
    if i == roller-ball dung beetle then.
        δ =rand(1);
        if δ < 0.9 then.
            Select α value;
            Use Eq. (10) to update the roller-ball dung beetle's position;
        else.
            Use Eq. (12) to update the roller-ball dung beetle's position;
        end if.
    end if.
    if i == spawning dung beetle then.
        Use Eq. (14) to update the spawning dung beetle's position;
    end if.
    if i == foraging dung beetle then.
        Use Eq. (16) to update the foraging dung beetle's position;
    end if.
    if i == stealing dung beetle then.
        Use Eq. (17) to update the stealing dung beetle's position;
    end if.
end for.
    if the newly generated position is better than before then.
        Update it;
    end if.
t = t + 1;
end while.
return X^b and its fitness value F_b.
```

Methodology

In this article, we propose a MOO approach to enhance the design of concrete mixture proportions. An application program based on Matlab has been developed to facilitate this method. The detailed operational procedures of the proposed method are depicted in Figure 3. Within this development framework, a hybrid CNN–

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

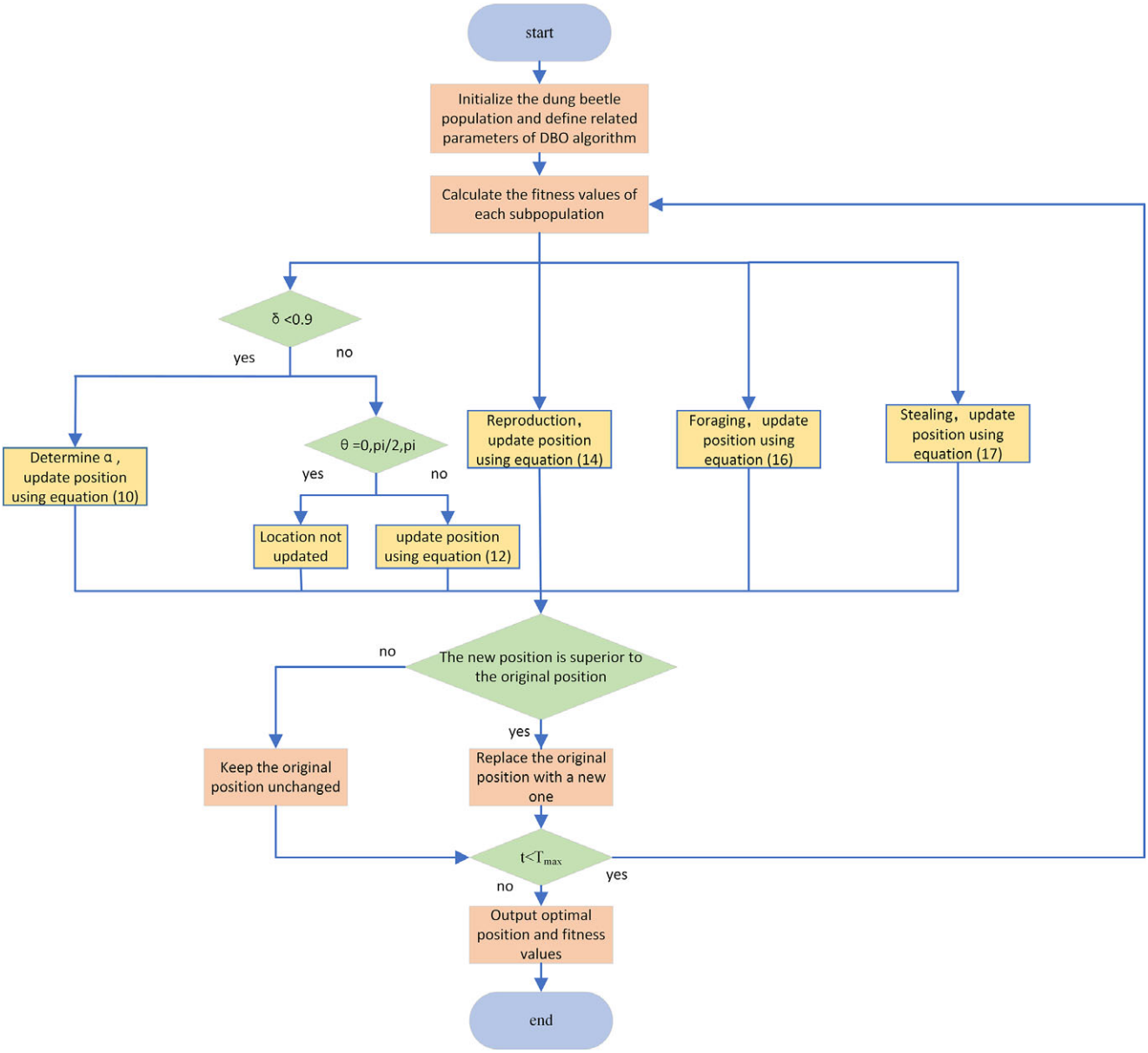


Figure 1. Flowchart of DBO.

NSDBO–TOPSIS intelligent method is employed to improve the CS, cost-effectiveness, and reduce carbon dioxide emissions of concrete. This method comprises the following four main steps.

Development of a CNN-based regression prediction model for CS in HPC

Data processing

Based on practical engineering and prior studies, the mixture proportion parameters for HPC include the amounts of cement (C), mineral powder (MP), fly ash (FLA), water (W), superplasticizer (SP), coarse aggregate (CA), and fine aggregate (FA). These components significantly impact the performance of concrete. Therefore, these key components are used as input variables for the CNN model. Running a CNN model requires setting crucial hyperparameters, specifically the learning rate and the number of convolutional kernels. It is essential to recognize that the concrete mixture proportion factors vary in dimensions and value ranges. To prevent any input quantity from being disproportionately large and skewing the output results, the samples are scaled to fit within [0,1]. This ensures that each parameter contributes equally. The formula for normalizing the input variables is as follows:

$$X_{norm} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{18}$$

where X_{norm} is the normalized data value and X is the initial value, with $X_{\max}$ and $X_{\min}$ being the initial maximum and minimum values, respectively.

To assess the forecasting capability of CNN, the dataset is typically split into a training set and a test set. CNN is then employed to train a model using the training set, which is subsequently used to make predictions on the testing set. The disparity between the forecast outcomes and the true data is computed to assess the predictive efficacy of the model.

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

CNN hyperparameter selection

In this study, we chose to optimize the learning rate and the number of convolution kernels in the CNN for several important reasons:

(1) The Importance of Learning Rate

The learning rate is a crucial hyperparameter that significantly affects the training process and the results of CNN models. It determines the step size for each weight update, directly influencing the convergence speed and final performance of the model. If the learning rate is too high, the model may oscillate and fail to converge; if it is too low, the training process will be slow and may get stuck in local optima. Therefore, selecting an appropriate learning rate is essential for enhancing the training efficiency and prediction accuracy of the model.

(2) The Importance of the Number of Convolution Kernels

The number of convolution kernels determines the richness of the features extracted by each convolution operation. More convolution kernels can capture more image details and features, thereby enhancing the model's expressive power. However, too many convolution kernels can increase computational complexity and the risk of overfitting, while too few kernels may result in the model failing to learn important features in the data. Thus, finding a balanced number of convolution kernels is key to optimizing CNN performance.

(3) Complementary Optimization Effects

The learning rate and the number of convolution kernels have complementary effects on the training and prediction capabilities of CNN models. The learning rate adjusts the pace of model weight updates, while the number of convolution kernels affects the depth and breadth of feature extraction. Optimizing these two hyperparameters together can comprehensively improve the training efficiency and prediction performance of the model.

(4) Balance Between Practicality and Computational Cost

Compared to other hyperparameters, such as the number of network layers or batch size, adjusting the learning rate and the number of convolution kernels requires relatively lower computational resources, but can significantly improve model performance. Therefore, optimizing these two hyperparameters can yield substantial performance gains at a reasonable computational cost, making it an efficient and practical optimization strategy. By using intelligent optimization algorithms (such as DBO) to optimize the learning rate and the number of convolution kernels in CNN models, the optimal configuration can be automatically found. This ensures that the model achieves optimal performance during both training and prediction phases.

Validation metrics for regression predictive model

The CNN regression prediction model is an algorithm based on CNN designed for predictive tasks. The primary concept of the CNN model is to extract features from input data using convolution operations and then progressively process these features through pooling layers and fully connected layers to achieve prediction objectives. In this study, the CNN model was utilized to predict the CS of HPC. To validate the efficacy of the CNN regression prediction model, four widely employed statistical metrics were employed to evaluate its prediction accuracy. For detailed information on these indicators, please refer to Table 2. These indicators allow us to comprehensively evaluate the accuracy and effectiveness of the CNN model in predicting the CS of HPC.

Table 2. The four performance metrics adopted in the experiment section

Metrics	Explanation	Equations
R^2	R^2, also known as determination coefficient, evaluates the goodness of fit between the model and the data, ranging from 0 to 1. The closer the value is to 1, the better the model fits the data.	$R^2 = 1 - \frac{\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{N}(y_i - \bar{y}_i)^2}$
RMSE	Root mean squared error measures the average deviation between the predicted values and the true values. A smaller RMSE value denotes higher prediction accuracy.	$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$
MAE	Mean absolute error is determined by calculating the absolute differences between the predicted and actual values and then averaging these differences across the dataset. A smaller MAE indicates higher predictive accuracy of the model.	$MAE = \frac{1}{N}\sum_{i=1}^{N}\lvert y_i - \hat{y}_i \rvert$
MAPE	Mean absolute percentage error is calculated by dividing the difference between the actual value and the predicted value by the actual value. The smaller the MAPE, the better the model's performance.	$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left\lvert \frac{y_i - \hat{y}_i}{y_i} \right\rvert * 100\%$

MOO of the HPC mixture proportion

Definition of the objective function

(1) Objective function of concrete CS based on the DBO–CNN model

To enhance both the calculation rate and accuracy of the optimization procedure, the regression model obtained from the optimized CNN and the mixture proportion is utilized as one of the fitness functions of NSDBO. The regression function based on the CNN is as follows:

$$CS = CNNRegress(\text{ConcreteCompos}) \tag{19}$$

where CS represents the CS of concrete, while *ConcreteCompos* refers to the various components that constitute the concrete mixture.

(2) Cost function of concrete

The raw material prices in this article are based on the prices in the Pearl River Delta region. There are 7 types of raw materials used, including C price (P_C), FLA price (P_{FLA}), MP price (P_{MP}), FA price (P_{FA}), CA price (P_{CA}), W price (P_W), and SP price (P_{SP}). The average prices of various raw materials obtained through market research are shown in Table 3.

The unit cost of concrete in the objective function is calculated using Eq. (20).

$$Cost = C_C P_C + C_{FLA} P_{FLA} + C_{MP} P_{MP} + C_{FA} P_{FA} + C_{CA} P_{CA} + C_W P_W + C_{SP} P_{SP} \tag{20}$$

In Eq. (20), C_C, C_{FLA}, C_{MP}, C_{FA}, C_{CA}, C_W, and C_{SP}, respectively, represent the amount of C, FLA, MP, FA, CA, W, and SP per cubic

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

Table 3. Unit price of each raw material

Raw Material	C (P_C)	MP (P_{MP})	FLA (P_{FLA})	W (P_W)	SP (P_{SP})	CA (P_{CA})	FA (P_{FA})
Unit price (Yuan/kg)	0.5	0.4	0.28	0.005	0.16	0.11	0.125

meter of concrete (kg/m3), and Cost represents the unit cost of concrete (Yuan/m^3).

(3) Carbon dioxide emissions function of concrete

Cement production is one of the main sources of carbon dioxide emissions. In the process of cement production, limestone (calcium carbonate) is heated to high temperature to form lime (calcium oxide), which releases carbon dioxide. Therefore, the cement production process directly leads to a large amount of carbon dioxide emissions. In order to reduce carbon dioxide emissions, cement should be used less. Therefore, it is necessary to establish another objective function (related to carbon dioxide emissions), and Eq. (21) has been used.

$$CO_2 = 0.9 \times C_C \tag{21}$$

Variable constraints

(1) Constraints on ingredient ranges

The research integrates upper and lower bounds for concrete ingredients, delineated by Eq. (22). These constraints define the acceptable search space for all variables, ensuring compliance with predefined limits. Typically, the upper and lower bounds for ingredients are determined by the regulatory standards governing concrete mixture proportion design in the respective country.

$$A_{i,\min} \le A_i \le A_{i,\max} \tag{22}$$

where A_i is the *ith* variables value and $A_{i,\max}$ and $A_{i,\min}$ are the upper and lower bounds of the *ith* variable, respectively.

Table 4 presents the range considered in this investigation, detailing the permissible values for each ingredient. Adhering to these constraints ensures that the mix design remains within acceptable bounds and complies with regulatory standards.

(2) Constraints on ingredient volumes

The overall volume of concrete is fixed at 1 m^3, calculated according to Eq. (23).

$$V_{total} = 1m^3 = \frac{C_C}{\rho_C} + \frac{C_{MP}}{\rho_{MP}} + \frac{C_{FLA}}{\rho_{FLA}} + \frac{C_W}{\rho_W} + \frac{C_{SP}}{\rho_{SP}} + \frac{C_{CA}}{\rho_{CA}} + \frac{C_{FA}}{\rho_{FA}} \tag{23}$$

where $\rho_C, \rho_{FLA}, \rho_{MP}, \rho_{FA}, \rho_{CA}, \rho_W$, and ρ_{SP} are density in kg/m^3, whereas $C_C, C_{FLA}, C_{MP}, C_{FA}, C_{CA}, C_W$, and C_{SP} are quality in kg of *C, FLA, MP, FA, CA, W*, and *SP*, respectively.

Table 4. Variable weights and range restrictions

Variables Unit	C	MP	FLA	W kg/m^3	SP	CA	FA	Age Days
Upper Limit	540	360	260	247	33	1145	993	365
Lower Limit	102	0	0	121	0	708	594	1
Density(kg/m^3)	1440	2800	2500	1000	1200	2700	2600	–

Table 5. Constraints on the ratios between variables

Constraints	Lower limit	Upper limit
SCM/C	0	0.35
W/C	0.3	0.6
C/CA	0.2	0.6
C/FA + CA	0.05	0.33
CA/FA + CA	0.4	0.65

(3) Constraints on ingredient proportion

In the proportion constraint, relationships among variables are defined by setting specific ratios. Ratio constraints SCM/C, W/C, C/CA, C/(FA + CA), and CA/(FA + CA) are considered. Eq. (24) defines the boundaries for the ratio constraint values in Table 5.

$$B_{i,\min} \le B_i \le B_{i,\max} \tag{24}$$

where B_i is ratio constraints and $B_{i,\min}$ and $B_{i,\max}$ are the minimum and maximum values of the *ith* ratio constraint.

Non-dominated sorting dung beetle optimizer

Once the fitness function and constraints are defined, NSDBO (Zhu et al., 2023) is employed to perform MOO of HPC mixture proportion. This aims to get a Pareto solution set that achieves the optimal mixture proportion with the lowest economic cost and minimal carbon dioxide emissions while maximizing the CS of the concrete. Non-dominated sorting is an efficient and powerful method commonly used in MOO algorithms. It ranks solutions based on the degree of their Pareto optimal. Solutions that are not dominated by any other solutions are given rank 1. Those dominated by only one solution are given rank 2, those dominated by two others are given rank 3, and this pattern continues accordingly. The solutions are then selected based on their ranks to improve the quality of the population. The main steps required to obtain PF through NSDBO are as follows:

(1) First, define the population size, dimension, maximum iterations, upper and lower limits of decision variables, and Pareto archive size, randomly initialize the dung beetles population, and store them into matrices. Calculate the fitness value for each individual in the population, identify the non-dominated solutions in the initial population, and store them in the Pareto archive. Then, compute the crowding distance for each member of the Pareto archive.
(2) Then, update the positions of the roller-ball dung beetles, spawning dung beetles, little dung beetles, and thief dung beetles. Therefore, a new generation of offspring has been obtained.
(3) Update population fitness. The parent and offspring populations are merged together. Perform the non-dominated sorting, and calculate the crowding distance. Start adding each front based on rank and crowing distance until the whole Pareto archive is filled.

This process continues until the algorithm reaches its maximum number of iterations or another stopping criterion is satisfied. At that point, the Pareto-optimal solution set is produced. The flowchart of NSDBO is shown in Figure 2.

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

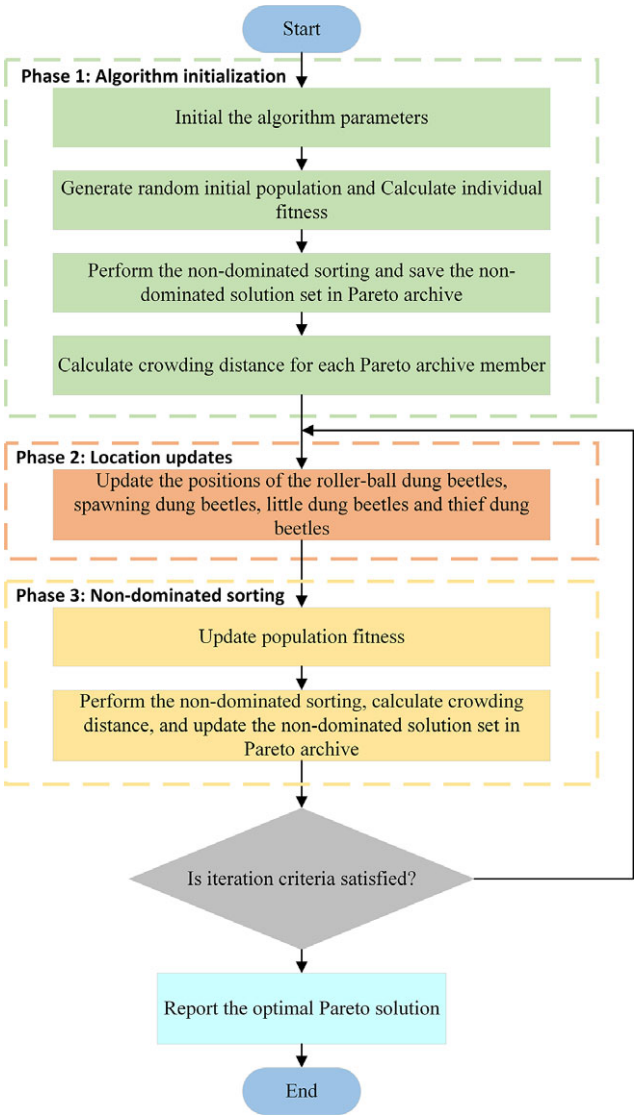


Figure 2. Flowchart of NSDBO.

Intelligent decision-making through EWTOPSIS method

Employing the EWTOPSIS method, we conducted a thorough evaluation and decision analysis for 60 individuals within the PF, representing 60 mixture proportion optimization schemes, to pinpoint the optimal solution within this frontier. This method, using a normalized decision matrix, is a MOO analysis technique. It incorporates the principle of information entropy to ascertain the objective weights of the indicators, computes the Euclidean distance between each assessed object and the ideal target, ranks the schemes according to their relative proximity, and identifies the optimal solution (Lv et al., 2023). This approach accurately captures the significance of the evaluated indicators, thereby enhancing the objectivity of the evaluation results. The calculation method is outlined as follows:

(1) Establish the initial assessment matrix

Define the initial assessment matrix as the set of assessment criteria for the 60 mixture proportion optimization schemes in the PF. It can be represented as follows:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1j} \\ a_{21} & a_{22} & \cdots & a_{2j} \\ \cdots & \cdots & \cdots & \cdots \\ a_{i1} & a_{i2} & \cdots & a_{ij} \end{pmatrix} \tag{25}$$

where A is the initial assessment matrix, a_{ij} represents the j-th evaluation index value in the i-th mix proportion optimization scheme, $i = 1,2,\ldots, m, j = 1,2,\ldots, n$, m is the number of optimized mix proportion schemes, and n is the number of assessment indicators.

(2) Build a standardized decision matrix

Perform dimensionless processing on matrix A and normalize all indicators. The dimensionless formulas corresponding to each indicator type are shown in Eqs. (26) and (27).

Positive indicators:

$$b_{ij} = \frac{a_{ij} - \min(a_{ij})}{\max(a_{ij}) - \min(a_{ij})} \tag{26}$$

Reverse indicators:

$$b_{ij} = \frac{\max(a_{ij}) - a_{ij}}{\max(a_{ij}) - \min(a_{ij})} \tag{27}$$

where b_{ij} represents the elements in the normalized decision matrix.

(3) Calculate the entropy weight of evaluation indicators

The formula for calculating entropy weight is shown in Eq. (28).

$$w_j = \frac{1 - e_j}{\sum_{j=1}^{n}(1 - e_j)} \tag{28}$$

where w_j is the entropy weight of the j-th assessment indicator; $e_j = -k\left[\sum_{i=1}^{m} f_{ij} \ln f_{ij}\right]$, representing the information entropy of the j-th assessment indicator; $k = 1/\ln m$; $f_{ij} = (1 + b_{ij}) / \sum_{i=1}^{m}(1 + b_{ij})$.

(4) Construct weighted decision matrix

$$c_{ij} = w_j \times b_{ij} \tag{29}$$

where c_{ij} is the element in the weighted decision matrix.

(5) Determine positive and negative ideal solutions

$$\begin{cases} C^+ = \{c_1^+, c_2^+, \ldots, c_n^+\} \\ C^- = \{c_1^-, c_2^-, \ldots, c_n^-\} \end{cases} \tag{30}$$

where C^+ and C^- represent the positive ideal solution set and the negative ideal solution set, respectively; c_n^+ and c_n^- represent positive ideal solutions and negative ideal solutions, respectively.

(6) Determine the Euclidean distance between the optimization scheme of each mix proportion and the positive and negative ideal solutions

$$\begin{cases} D_i^+ = \sqrt{\sum_{j=1}^{n}\left(c_{ij}^+ - c_{ij}\right)^2} \\ D_i^- = \sqrt{\sum_{j=1}^{n}\left(c_{ij}^- - c_{ij}\right)^2} \end{cases} \tag{31}$$

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

where D_i^+ and D_i^-, respectively, represent the distance between the i-th mix ratio optimization scheme and the positive and negative ideal solutions.

(7) Determine the relative closeness of each mix proportion optimization scheme to the ideal solution

$$O_i = \frac{D_i^-}{D_i^- + D_i^+} \tag{32}$$

where O_i represents the relative closeness of the i-th optimized mix proportion scheme.

Implementation of NSDBO in three-objective HPC mixture proportion optimization

Three-objective optimization of concrete mixture proportions to identify optimal ingredient proportions is conducted through the integration of the CNN model and NSDBO algorithm, following the steps outlined below. The workflow of this process is depicted in Figure 3.

Step 1: The CS is represented using a CNN, while the DBO algorithm is employed to explore and determine optimal hyperparameter values.

Step 2: Determine the cost and carbon dioxide (CO_2) emission by applying Eqs. (20) and (21), respectively.

Step 3: Construct a three-objective optimization task by combining all objectives using Eqs. (33) and (34).

$$Objective\ function: \begin{cases} \max f(CS) \\ \min f(Cost) \\ \min f(CO_2) \end{cases} \tag{33}$$

$$Subjected\ to: \begin{cases} Range\ Constra\ \text{int}\ s \\ Volume\ Constra\ \text{int}\ s \\ Ratio\ Constra\ \text{int}\ s \end{cases} \tag{34}$$

Step 4: Utilize the NSDBO algorithm to search for optimal solutions to the formulated MOO problems.

Step 5: Obtain non-dominated PF solutions.

Step 6: Choose a solution from the PF based on Eq. (32).

Experimental results and analysis

All experiments were conducted using MATLAB R2022b on an Intel(R) Core(TM) i5-8400 CPU @ 2.80GHz processor with 8 GB RAM and running a 64-bit version of Microsoft Windows 11. Hence, different hardware environments will definitely result in different time-consuming outcomes.

High-strength concrete compressive strength prediction based on CNN

Data collection and processing

The experimental dataset on concrete CS is sourced from the UCI public databases, initially curated by Yeh (Yeh, 1998). It consists of 1133 samples. Initially, the data sample had a total of eight input variables, namely C, MP, FLA, W, SP, CA, FA, and Age. The output, representing concrete properties, is the uniaxial CS of the concrete. Furthermore, the statistical distributions of the relevant parameters are presented in Figure 4, enabling direct observation. The figure clearly shows the range and frequency of each parameter's distribution. In addition, five new features were introduced in the dataset (SCM/C (MP + FLA = SCM), W/C, C/CA, C/A, and CA/A). These

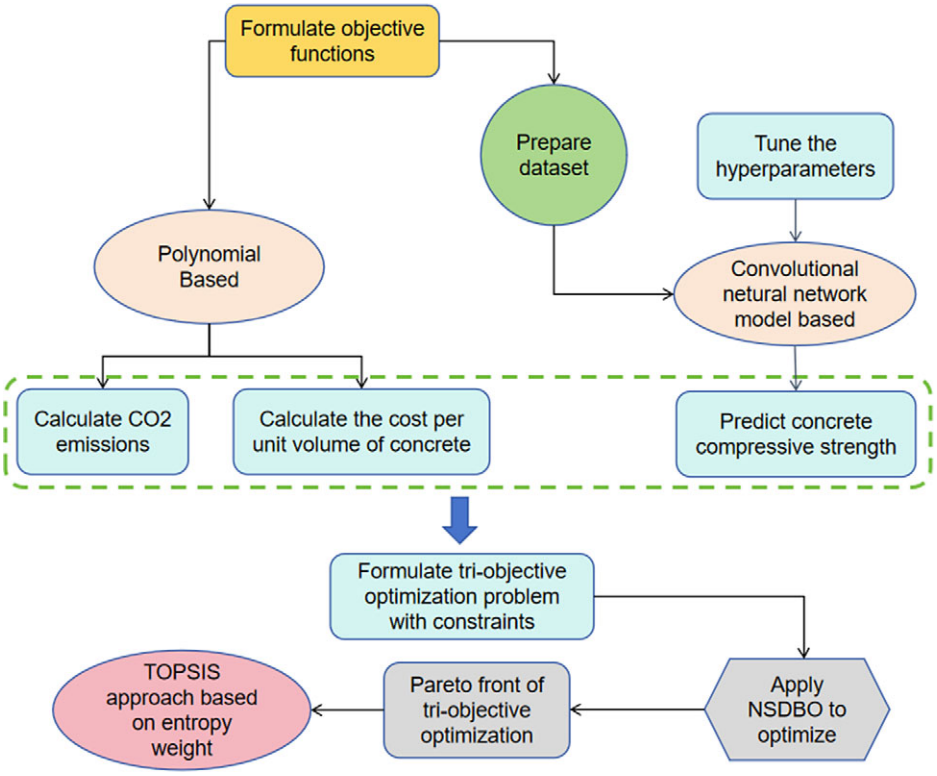


Figure 3. Flowchart of CNN–NSDBO–EWTOPSIS for high-strength concrete mix proportion optimization.

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

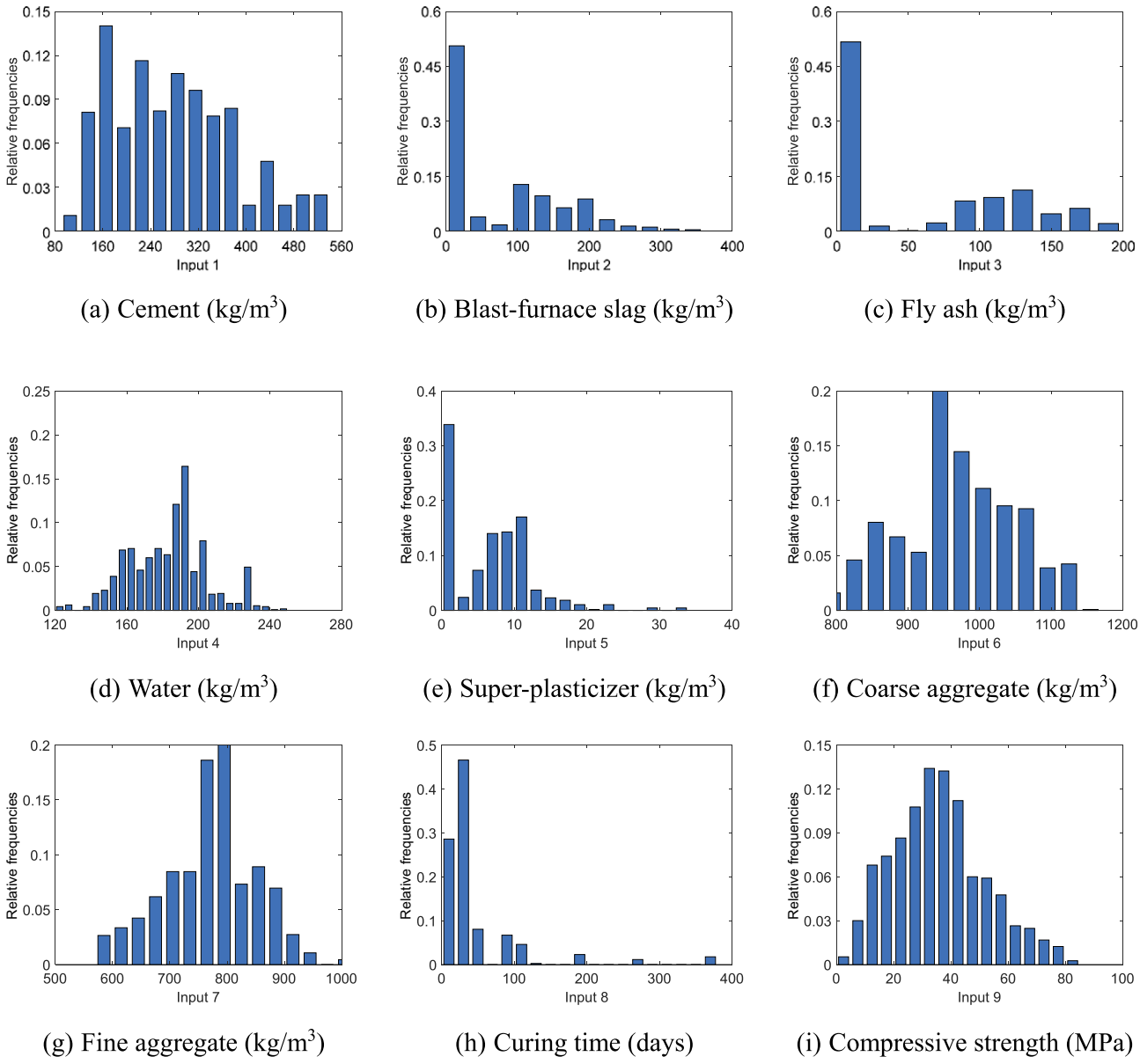


Figure 4. Statistical distributions of the input/output variables.

features exhibit a high correlation with CS, thereby enhancing the model's prediction accuracy. The sensitivity analysis is shown Figure 5. Table 6 presents detailed static data samples. The correlation coefficients between variables are illustrated using the Pearson correlation coefficient (Benesty et al., 2009) in Figure 6. The dataset contains no missing samples, and the variable ranges are evenly distributed.

CNN hyperparameter selection

In this section, the experiment involves optimizing the hyperparameters of CNN, using the DBO. Hyperparameter tuning aims to find the optimal hyperparameter combination for a CNN regression prediction model to achieve the best predictive performance. We choose to optimize the learning rate and the number of convolution kernels. This process can be viewed as an optimization problem. Specifically, the combination of hyperparameters is considered a vector of decision variables, analogous to dung beetles' position. The goal is to minimize the root mean square error, which is treated as the objective to be optimized. The optimal hyperparameters of the model are identified as the global best position found in the final iteration of the DBO algorithm. The detailed steps for using DBO in hyperparameter tuning are illustrated in Figure 7.

In the optimization process using the DBO, fitness serves as the primary metric for assessing individual performance, driving the optimization process, and guiding the selection of individuals. To enhance both the speed and accuracy of this optimization process, we employ the root mean square error between the concrete CS predicted by a trained CNN and the actual values as the fitness function for the DBO. This approach effectively captures the complex nonlinear relationship between the input variables and the output objectives. Thus, the fitness function based on the CNN can be expressed as follows:

$$RMSE = FitFun(X, \mathrm{TraIn}, \mathrm{TraOut}, \mathrm{TraNum}, \mathrm{DR}, \mathrm{NumF}, \mathrm{NumR}, \mathrm{FS}) \tag{35}$$

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

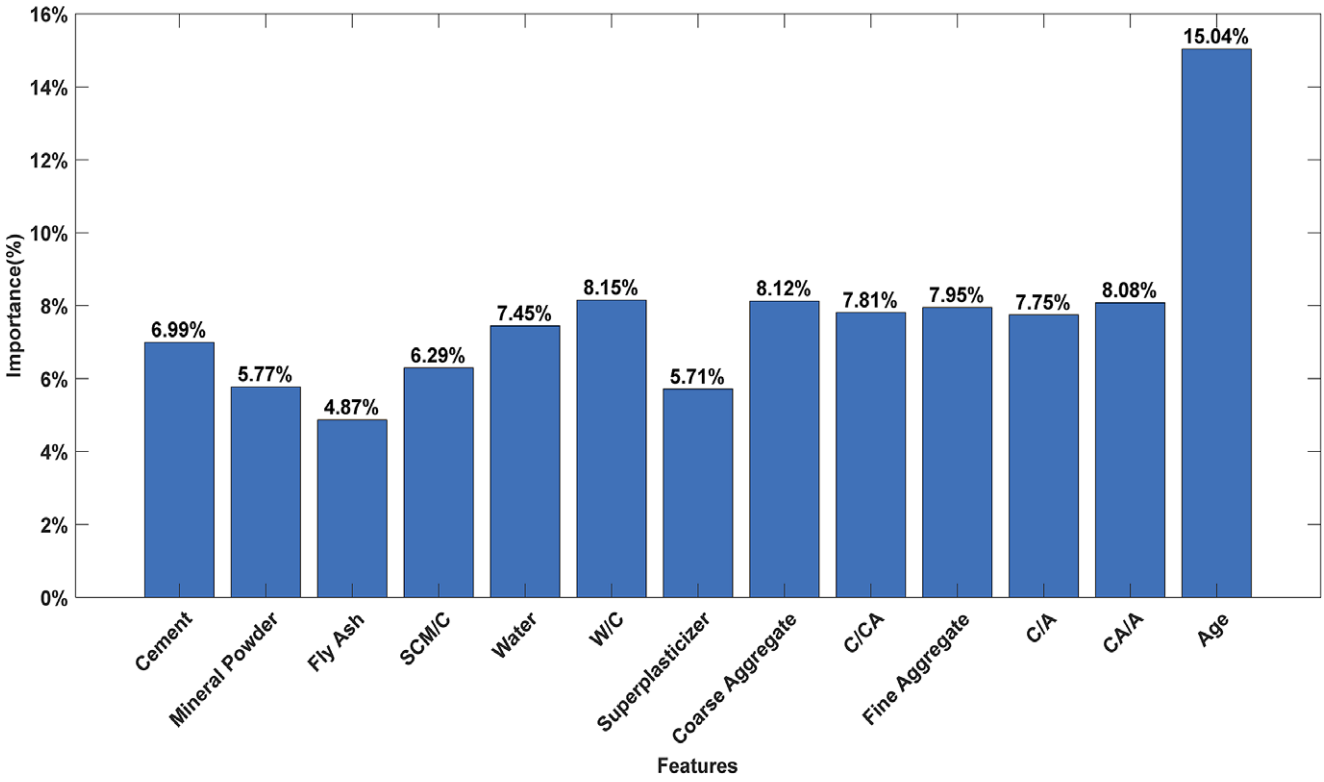


Figure 5. The importance of input variables in determining the CS of concrete.

Table 6. Statics of the dataset

	Mean	Std	Lower Limit	Upper Limit
Input				
C (kg)	276.5046	103.4699	102	540
MP (kg)	74.2662	84.2468	0	359.4000
FLA (kg)	62.8078	71.5832	0	260
SCM/C	0.6607	0.5955	0	2.7879
W (kg)	182.9847	21.7139	121.7500	247
W/C	0.7677	0.3214	0.2669	1.8824
SP (kg)	6.4155	5.7964	0	32.2000
CA (kg)	964.8331	82.7882	708	1145
C/CA	0.2895	0.1143	0.1150	0.6442
FA (kg)	770.4903	79.3739	594	992.6000
C/A	0.1607	0.0633	0.0558	0.3228
CA/A	0.5561	0.0358	0.4596	0.6522
Age (days)	44.0565	60.4413	1	365
Output				
CS (MPa)	35.8380	16.1005	2.3318	82.5992

where X represents the variables to be optimized, specifically the learning rate and the number of convolution kernels. *TraIn* denotes the input variables of the training set, while *TraOut* signifies the output of the training set. *TraNum* stands for the maximum number of training iterations, and *DR* represents the forgetting rate. Additionally, *NumF* and *NumR* indicate the number of input features and output features, respectively. Finally, *FS* denotes the size of the convolution kernel.

The parameter settings for DBO-optimized CNN are shown in Table 7. The hierarchical structure diagram of the constructed one-dimensional CNN is shown in Figure 8. The iterative convergence curve of the DBO-optimized CNN is shown in Figure 9.

Model validation

The training set is utilized to learn and establish a predictive model for concrete CS based on DBO–CNN. Subsequently, the established model is validated using the test set to assess its predictive performance. The prediction results and error diagrams for concrete CS obtained from the training set are depicted in Figure 10, with corresponding results and error diagrams for the test set shown in Figure 11. Additionally, Figure 12(a) illustrates the scatter plot of concrete CS obtained from the training set, while Figure 12(a) presents the corresponding scatter plot for the test set. According to the training outcomes of the DBO–CNN model, the following conclusions can be drawn: the DBO–CNN model exhibits outstanding regression learning capabilities and can serve as an effective alternative for capturing the relationship between concrete mix proportions and CS, thereby enhancing the efficiency of concrete design.

Furthermore, to demonstrate the effectiveness of DBO–CNN, we trained and tested CNN, BP, GA–BP, PSO–BP, ELM, RBF, and DBO–CNN separately on the training and testing sets. We evaluated our models using the four evaluation metrics mentioned earlier. The results are presented in Table 8, where it can be observed that DBO–CNN achieved the best performance. We also estimated the confidence interval based on the predicted values using statistical methods. Assuming the prediction errors follow a

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

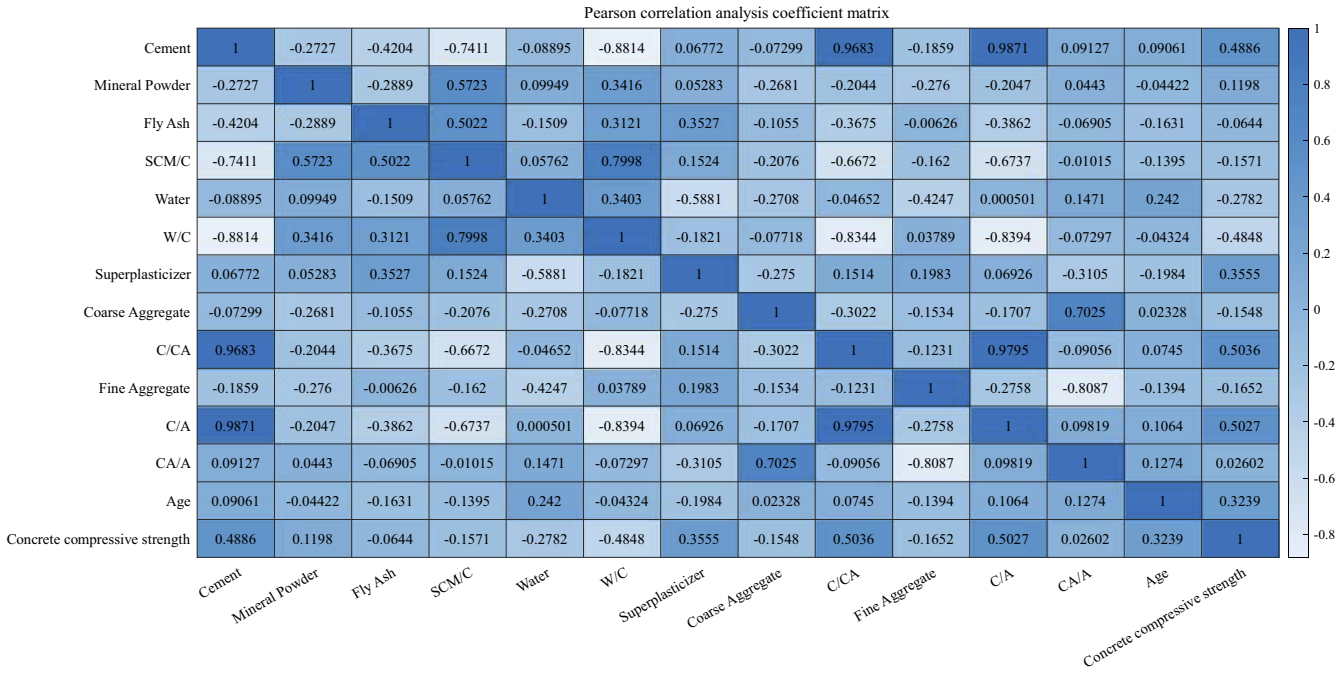


Figure 6. Pearson correlation coefficient between variables.

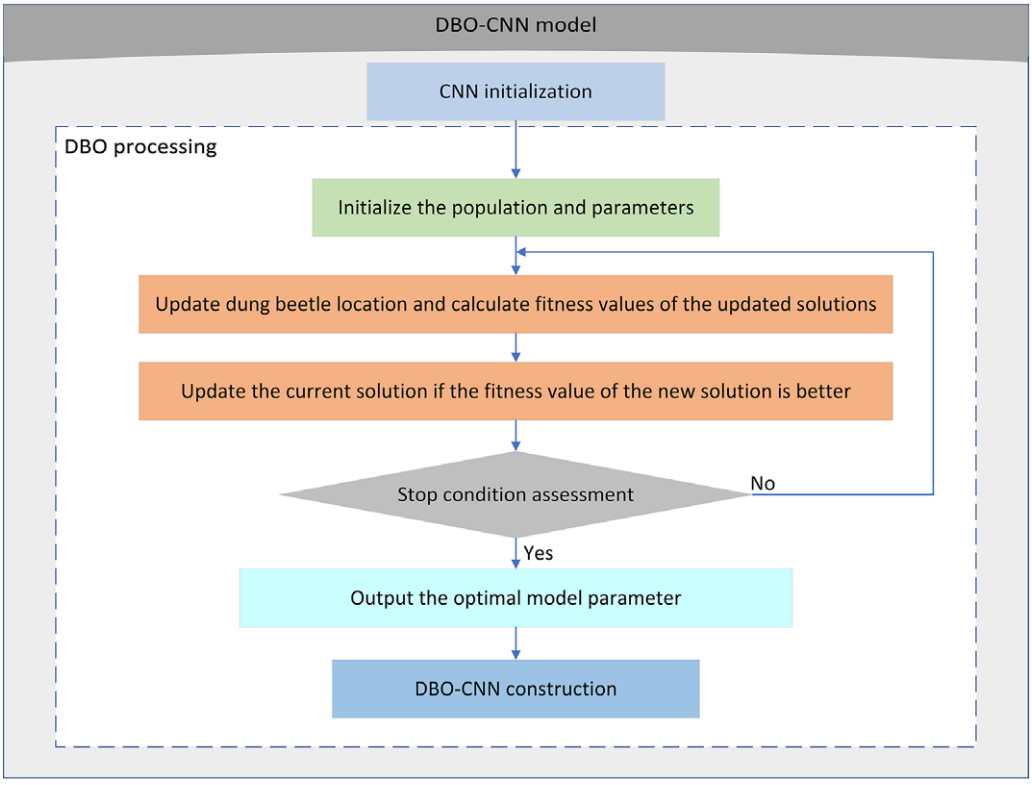


Figure 7. DBO–CNN model.

normal distribution, the confidence interval at the 95% confidence level was calculated using the mean and standard deviation. This calculation quantifies the uncertainty of the model's predictions, enhancing the reliability of decision-making. The histogram of error distribution and the fitting curve based on normal distribution are shown in Figure 13. Taking the first sample's predicted value as an example, its corresponding 95% confidence interval is [17.74, 34.83].

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

Table 7. Parameters setting of DBO–CNN

Parameters	Value
Population size	5
The maximum iteration number	20
D	2
The range of learning rate	[0.001, 0.01]
The number of convolution kernels	(Yeh, 1998; Mirjalili et al., 2014)
TraNum	150
DR	0.2
FS	6

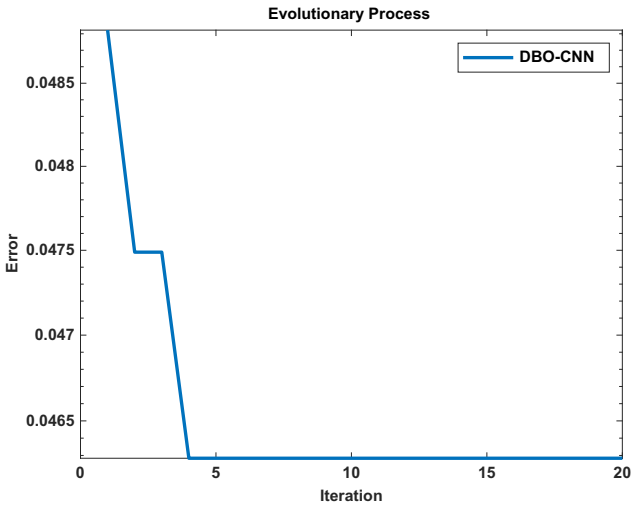


Figure 9. The iterative convergence curve of DBO-optimized CNN.

MOO of the HPC mixture proportion design based on CNN–NSDBO–EWTOPSIS

Definition of the objective function

(1) Objective function of concrete CS based on the DBO–CNN model

Previously, we optimized the hyperparameters of the CNN using DBO, resulting in the DBO–CNN regression prediction model. This model is used as the first objective function to be solved for the design problem of HPC mixture proportion. The objective function for maximizing the CS of concrete, $\max f_1$, can be formulated as follows:

$$f_1 = \max\{CNN\,\mathrm{Re}gression(C, MP, FLA, W, SP, CA, FA, Age)\} \tag{36}$$

(2) Concrete Cost Function

In this context, the real price of concrete primary materials depends on the unit prices of C, MP, FLA, W, SP, CA, and FA, which are equivalent to 0.5, 0.4, 0.28, 0.005, 0.16, 0.11, and 0.125 yuan/kg, respectively. The objective function for minimizing the concrete cost, $\min f_2$, can be formulated as follows:

$$\begin{aligned}\min f_2 = {} & 0.5C + 0.4MP + 0.28FLA + 0.005W + 0.16SP \\ & + 0.11CA + 0.125FA\end{aligned} \tag{37}$$

(3) Carbon dioxide emissions function of concrete

Typically, the amount of carbon dioxide produced by manufacturing 1 m^3 of concrete primarily depends on the quantity of cement used. When preparing concrete, using 1 kg of cement will produce approximately 0.9 kg of carbon dioxide emissions. The objective function for minimizing the carbon dioxide emissions of concrete, $\min f_3$, can be formulated as follows:

$$\min f_3 = 0.9C \tag{38}$$

Variable constraints

For the preparation of HPC, C and FLA are commonly used as binding materials, with super-plasticizes added to enhance workability. According to HPC design specifications and engineering practices, the dosage of each component in the HPC mixture should fall within an acceptable range and comply with the corresponding proportion constraints. The constraint conditions for optimizing the HPC mixture proportion are determined according to Eq. (39).

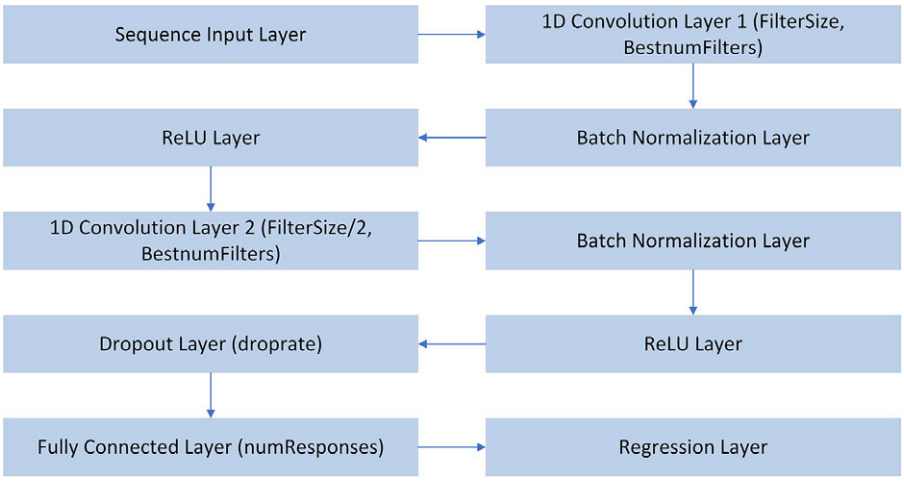


Figure 8. Hierarchical structure of one-dimensional CNN.

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

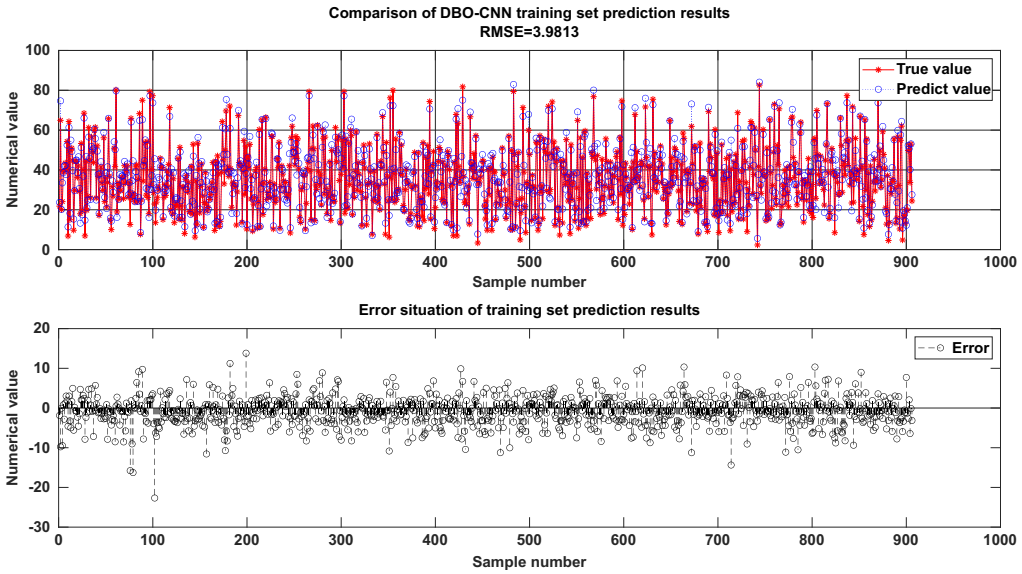


Figure 10. Train output results and error output results of prediction models based on DBO–CNN.

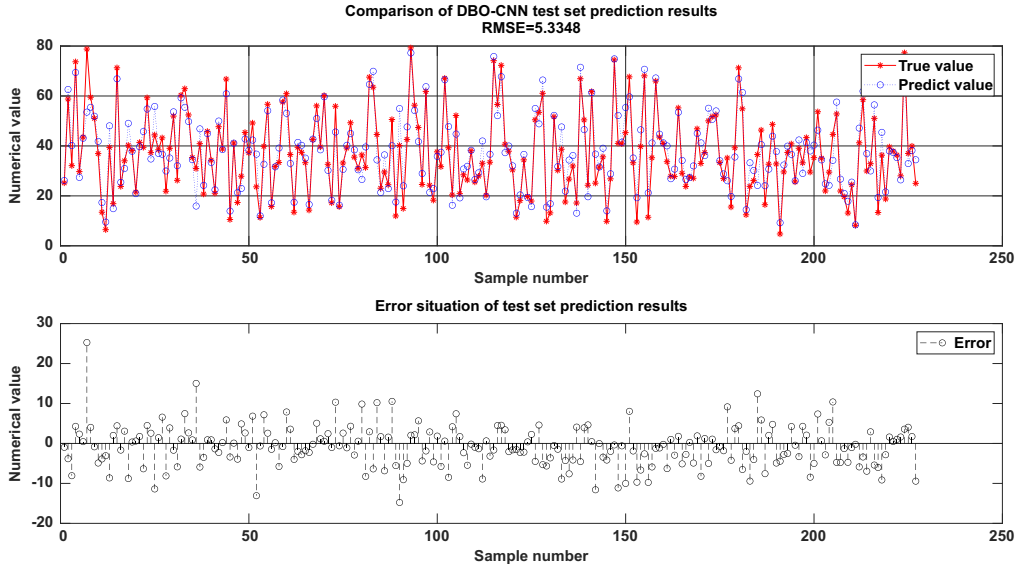


Figure 11. Test output results and error output results of prediction models based on DBO–CNN.

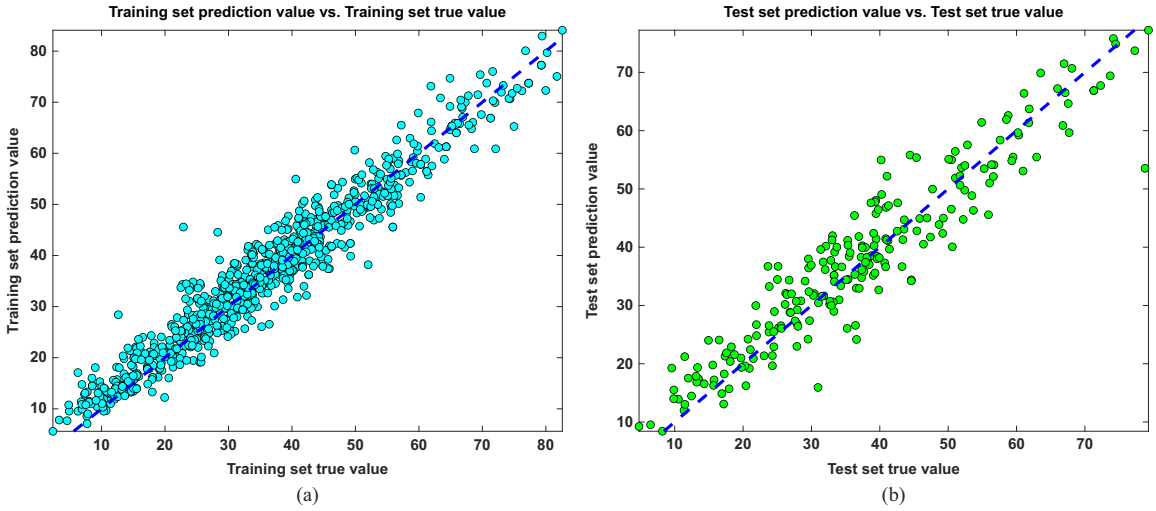


Figure 12. Prediction results for the concrete CS on training set and test set.

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

Table 8. Training and testing results of prediction models

	Train results				Test results			
	R^2	RMSE	MAE	MAPE	R^2	RMSE	MAE	MAPE
BP	0.88637	5.3551	4.0785	1.4173e-05	0.87398	5.977	4.6174	0.00011653
GA–BP	0.91376	4.9513	3.6402	1.0525e-05	0.83845	6.0561	4.6108	0.00018203
PSO–BP	0.86802	6.0948	4.7845	3.6057e-05	0.81579	6.8512	5.2287	0.00032205
ELM	0.76532	8.0105	6.2155	4.4319e-05	0.78497	8.3807	6.7827	0.00035562
RBF	0.83424	6.8077	5.3131	3.1934e-05	0.78154	7.7054	5.8852	0.0004256
CNN	0.93132	4.1779	3.2294	2.6001e-05	0.88434	5.6726	4.1719	0.00021343
DBO–CNN	**0.93853**	**3.9813**	**3.0456**	**8.6408e-06**	**0.89118**	**5.3348**	**4.1142**	**9.6936e-07**

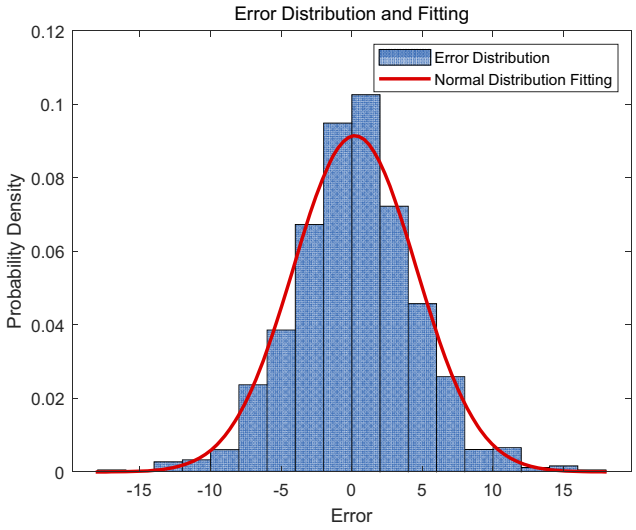


Figure 13. Error distribution histogram and fitting curve based on normal distribution.

$$
\begin{aligned}
&102 \le C \le 540\\
&0 \le MP \le 360\\
&0 \le FLA \le 260\\
&121 \le W \le 247\\
&0 \le SP \le 33\\
&708 \le CA \le 1145\\
&594 \le FA \le 993\\
&1 \le Age \le 365\\
&0 \le (MP + FLA)/C \le 0.35\\
&0.3 \le W/C \le 0.6\\
&0.2 \le C/CA \le 0.6\\
&0.05 \le C/(CA + FA) \le 0.33\\
&0.4 \le CA/(CA + FA) \le 0.65\\
&1 = \frac{C}{1440} + \frac{MP}{2800} + \frac{FLA}{2500} + \frac{W}{1000} + \frac{SP}{1200} + \frac{CA}{2700} + \frac{FA}{2600}
\end{aligned}
\tag{39}
$$

Three-objective optimization results and discussion

During the optimization process, NSDBO performs four different operations. Therefore, several parameters must be set before the iterative optimization process, including the population size, the maximum number of iterations, and the parameters k, b, and s. To balance optimization effectiveness and convergence speed, the population size is set to 60, and the maximum number of iterations (serving as the termination criterion) is set to 100. Additionally, k is set to 0.1, b to 0.3, and s to 0.5. To demonstrate the effectiveness of NSDBO, we selected NSGA-II and Multi-Objective Particle Swarm Optimization (MOPSO) as comparative algorithms, as they have been widely used in the literature to solve similar problems. The algorithm parameter settings are detailed in Table 9.

Table 9. Algorithm parameter settings

Algorithm	Parameters	Value
NSDBO	Population size	60
	Maximum number of iterations	100
	Repository size	60
	k	0.1
	b	0.3
	s	0.5
NSGA-II	Population size	60
	Maximum number of iterations	100
	Repository size	60
	Tournament size	2
	Distribution index of crossover operator	20
	Distribution index of mutation operator	20
MOPSO	Population size	60
	Maximum number of iterations	100
	Repository size	60
	Inertia weight	0.5
	Inertia weight damping rate	0.99
	Personal learning coefficient	1
	Global learning coefficient	2
	Number of grids per dimension	7
	Inflation rate	0.1
	Leader selection pressure	2
	Deletion selection pressure	2
	Mutation rate	0.1

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

With the objective function, variable constraints, and model parameters set as described, 60 Pareto-optimal concrete mixtures were generated using 100 iterations of NSDBO, as illustrated in Figure 14. In the figure, the proportions of the ingredients in the Pareto solution are shown, where the *y*-axis represents the CS, and the *x*-axis represents all the components of the concrete mix. Additionally, we successfully obtained the PF, and the EWTOPSIS method was used for decision-making. Figure 15 shows the EWTOPSIS scores corresponding to the optimal front of the three-objective optimization problem. The solution with the highest score was chosen for practical implementation. In the figure, the color of the solutions corresponds to their assigned EWTOPSIS score. The obtained Pareto solution set and PF are presented in Table 10. To demonstrate the superiority of NSDBO in solving this problem, we also employed NSGA-II and MOPSO, which have been used to solve similar problems. The chart comparing the experimental results is shown in Figure 16. When solving this problem, only the spacing metric was used for evaluation because the true PF was unknown. A smaller spacing value indicates a more evenly distributed solution set. The values of the spacing metric obtained from the experiment are presented in Table 11. From Table 11 and Figure 16, we can observe that the distribution of the Pareto solution set for NSDBO is more uniform.

Conclusion and future work

Under the pressures of global warming and environmental protection, reducing the carbon footprint of building materials has become an industry trend. Developing more intelligent HPC mixture proportion design methods can meet performance requirements, while minimizing environmental impact and cost, thus supporting sustainable construction practices. This article established a hybrid MOO method, the CNN–NSDBO–EWTOPSIS method, aimed at enhancing the CS of concrete, while reducing its cost and carbon dioxide emissions. The proposed method contains three main steps: (1) train a CNN model to establish a regression predictive model for concrete CS by capturing the nonlinear mapping between concrete mix proportions and CS indicators; (2) optimize the hyperparameters of CNN using DBO; and (3) perform MOO of concrete mixture proportion using NSDBO to obtain the optimal concrete mixture proportion. Eventually, the effectiveness of the proposed approach was validated using data from the UCI public dataset.

Based on specific case study data, several valuable findings emerge. First, the developed CNN regression prediction model exhibits promising predictive performance across the entire dataset. Second, the mapping function derived from CNN training can act as a proxy model for iterative optimization. It can rapidly and precisely forecast the CS of each newly generated concrete mixture proportion, thus decreasing the experimental effort in concrete mixture proportion design, enhancing the efficiency of the iterative computation process, and significantly improving the overall efficiency of concrete mixture proportion optimization. Lastly, NSDBO exhibits outstanding MOO capability. It effectively balances the maximization of concrete CS with the minimization of concrete cost and carbon dioxide emissions. Therefore, the proposed hybrid intelligent CNN–NSDBO–EWTOPSIS method can effectively predict concrete CS and optimize mixture proportion. Additionally, this method is simple to implement, is highly reliable, and holds significant practical value in engineering practice for concrete construction. It can also offer insights for comparable projects. Nonetheless, this study has certain potential limitations. First, the predictive accuracy of the CNN regression algorithm heavily relies on the quality and diversity of the training dataset. In this research, the 1133 data instances used for training the model were sourced from the UCI public dataset. To ensure the applicability of our proposed method across various real-world scenarios, the dataset will be expanded in future research to include more factors. In summary, future studies need to broaden the concrete

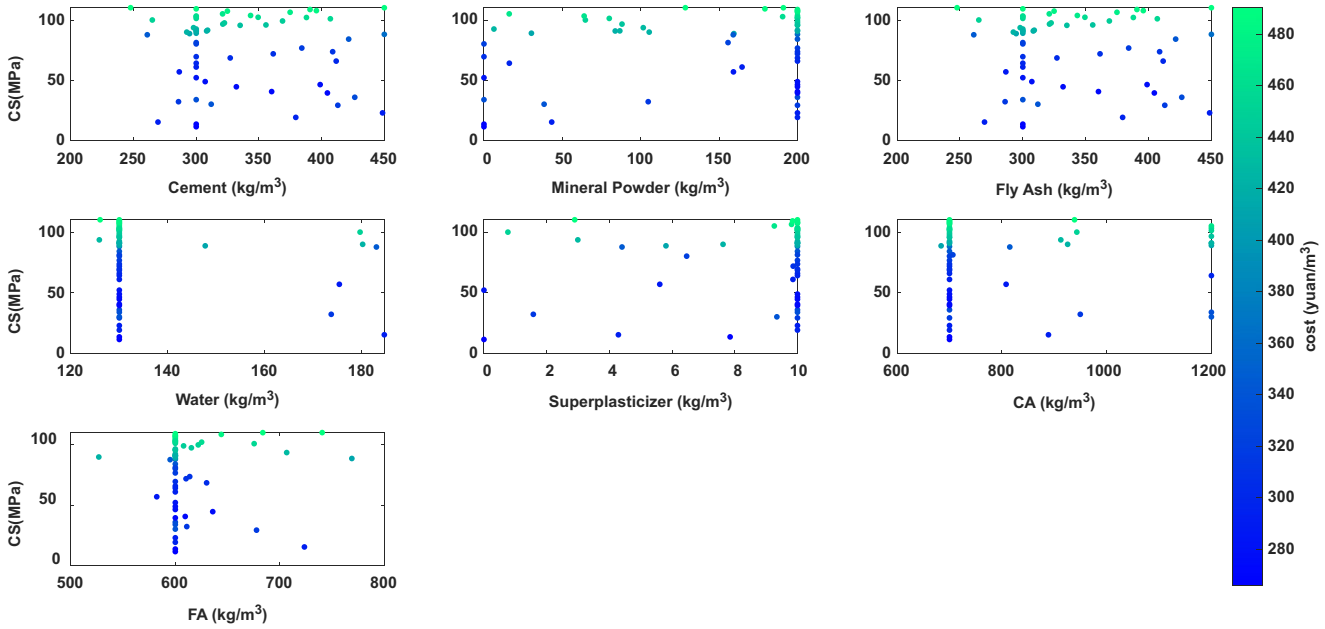


Figure 14. Effect of ingredient proportions on CS and cost in the obtained PF optimal solutions.

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

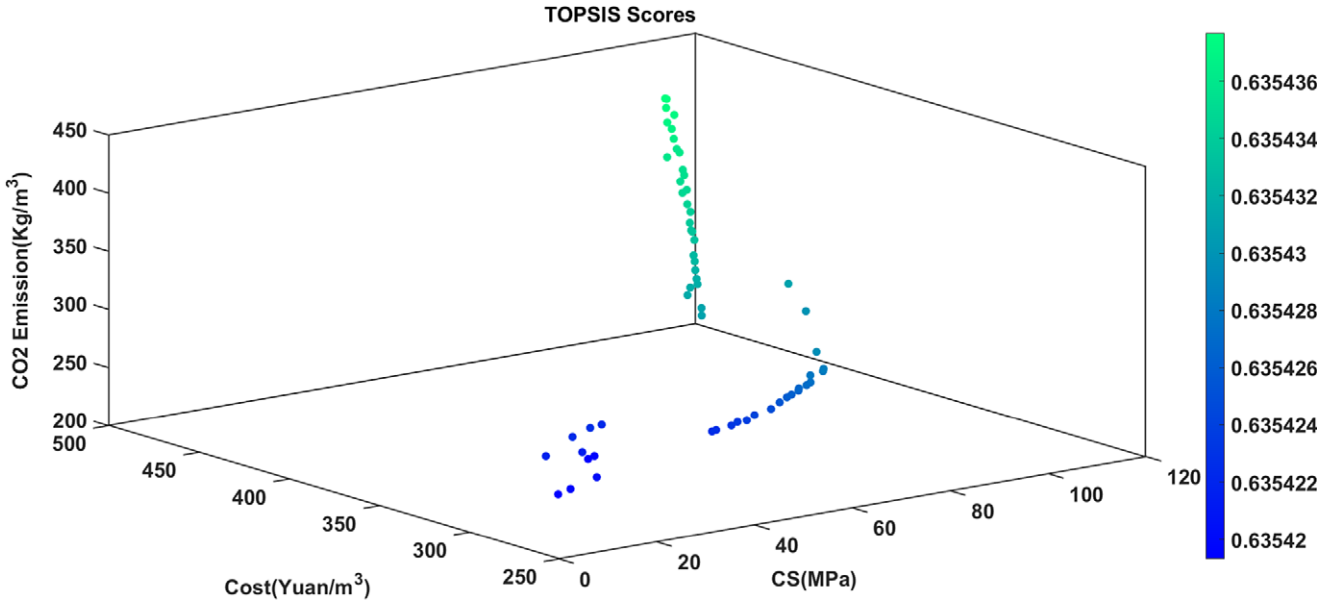


Figure 15. Three objective PF with EWTOPSIS evaluation.

Table 10. Obtained Pareto solution set and Pareto front

PS								PF		
C	MP	FLA	W	SP	CA	FA	Age	CS	Cost	CO_2
269.53	43.23	4.16	184.63	4.29	888.94	723.55	22.53	15.60	292.84	224.58
247.68	128.48	52.93	126.04	2.89	938.68	740.56	22.35	110.21	490.33	405.00
300.00	0.00	0.00	130.00	0.00	700.00	600.00	365.00	11.74	265.75	270.00
450.00	191.01	0.00	130.00	10.00	700.00	683.73	339.72	110.17	489.39	405.00
450.00	200.00	0.00	130.00	10.00	700.00	600.00	365.00	88.17	362.12	320.75
294.70	159.44	91.95	147.71	5.81	684.00	768.93	22.35	88.78	411.88	270.00
426.44	200.00	0.00	130.00	10.00	700.00	600.00	365.00	36.10	330.73	249.11
448.59	200.00	0.00	130.00	10.00	700.00	600.00	365.00	23.19	292.00	234.03
285.89	104.79	5.79	173.69	1.57	949.69	610.97	25.90	32.44	314.40	262.12
412.93	200.00	0.00	130.00	10.00	700.00	677.72	365.00	29.49	317.19	239.35
260.98	159.03	111.34	183.00	4.40	815.12	595.05	24.00	87.81	345.56	270.00
421.59	200.00	0.00	130.00	10.00	700.00	600.00	365.00	84.22	341.71	309.27
286.50	159.16	102.35	175.36	5.61	808.07	582.25	22.36	57.13	287.48	270.00
300.00	0.00	0.00	130.00	0.00	700.00	600.00	365.00	52.31	283.59	270.00
300.00	0.00	0.00	130.00	10.00	1200.00	600.00	365.00	34.13	335.16	240.78
321.04	16.11	0.00	130.00	9.26	1200.00	600.00	339.77	105.11	468.52	372.41
311.96	38.37	0.00	130.00	9.34	1200.00	600.00	335.57	30.37	339.73	224.99
297.89	101.63	46.79	125.88	2.99	912.53	706.59	25.31	93.68	429.07	305.20
300.00	30.27	3.52	130.00	10.00	1200.00	600.00	358.83	89.11	412.90	275.70
335.03	200.00	0.31	130.00	10.00	700.00	600.00	365.00	95.62	434.53	319.75

(*Continued*)

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

Table 10. (*Continued*)

PS								PF		
C	MP	FLA	W	SP	CA	FA	Age	CS	Cost	CO_2
265.00	64.69	60.26	179.63	0.76	943.08	621.94	24.67	100.04	450.45	339.87
379.40	200.00	1.60	130.00	10.00	700.00	600.00	365.00	19.47	296.48	224.63
374.89	200.00	0.04	130.00	9.81	700.00	600.00	358.11	106.43	475.29	380.09
395.84	200.00	0.00	130.00	10.00	700.00	600.00	365.00	107.79	482.58	389.79
343.36	200.00	0.00	130.00	10.00	700.00	600.00	365.00	103.78	471.77	367.94
324.87	200.00	0.00	130.00	10.00	700.00	600.00	365.00	107.37	478.98	386.24
300.00	179.23	0.00	130.00	9.84	700.00	600.00	362.49	109.28	487.28	398.98
300.00	6.36	0.00	130.00	10.00	700.00	600.00	365.00	92.56	426.68	312.26
300.00	63.91	0.00	130.00	10.00	1200.00	600.00	365.00	103.33	465.22	378.27
292.46	105.42	68.32	180.17	7.63	925.54	526.79	23.70	90.02	423.10	281.73
300.00	164.66	0.00	130.00	9.86	700.00	600.00	361.95	61.14	293.57	270.00
398.84	200.00	1.40	130.00	10.00	700.00	600.00	365.00	46.59	277.46	271.15
406.98	200.00	0.00	130.00	10.00	700.00	675.45	365.00	101.05	455.89	346.40
387.92	200.00	1.57	130.00	10.00	700.00	600.00	365.00	102.11	459.94	353.58
307.17	200.00	0.00	130.00	10.00	700.00	600.00	365.00	49.07	279.05	270.00
384.20	200.00	1.76	130.00	10.00	700.00	600.00	365.00	76.83	319.18	270.00
390.76	200.00	0.00	130.00	10.00	700.00	644.01	365.00	108.71	481.23	396.24
300.00	155.70	0.00	130.00	10.00	706.45	600.00	365.00	81.36	324.12	270.00
408.82	200.00	0.00	130.00	10.00	700.00	613.79	361.29	73.79	310.81	270.00
332.01	200.00	0.02	130.00	10.00	700.00	635.82	365.00	44.79	275.98	270.00
300.00	79.94	0.66	130.00	10.00	1200.00	600.00	365.00	101.36	454.34	349.41
349.39	200.00	0.00	130.00	10.00	700.00	625.18	365.00	102.30	459.10	364.03
321.21	88.07	1.42	130.00	10.00	1200.00	600.00	365.00	96.65	438.47	324.26
308.31	83.73	2.54	130.00	10.00	1200.00	600.00	365.00	90.96	420.65	296.05
300.00	190.54	0.00	130.00	10.00	700.00	600.00	358.58	102.81	459.69	358.79
322.51	200.00	0.00	130.00	10.00	700.00	615.38	358.41	97.63	442.51	329.36
300.00	86.68	3.30	130.00	10.00	1200.00	600.00	365.00	91.05	424.40	286.88
411.63	200.00	0.00	130.00	10.00	700.00	600.00	365.00	66.03	300.28	270.00
300.00	0.00	0.00	130.00	6.47	700.00	600.00	296.12	80.27	321.49	270.00
309.10	200.00	0.31	130.00	10.00	700.00	600.00	365.00	91.60	423.10	301.95
300.00	200.00	0.00	130.00	10.00	700.00	600.00	260.01	90.81	419.75	292.02
404.61	200.00	0.00	130.00	10.00	700.00	600.00	365.00	39.68	272.83	270.00
300.00	16.18	0.00	130.00	10.00	1200.00	600.00	365.00	64.28	298.12	270.00
360.16	200.00	0.00	130.00	10.00	700.00	609.48	365.00	40.81	273.64	270.00
368.84	200.00	0.00	130.00	10.00	700.00	607.93	365.00	99.22	446.46	335.80
327.13	200.00	0.02	130.00	10.00	700.00	629.96	362.00	68.65	303.56	270.00
361.42	200.00	0.01	130.00	9.86	700.00	610.27	358.14	72.04	308.27	270.00
355.62	200.00	0.01	130.00	10.00	700.00	600.00	365.00	95.95	437.17	326.54
300.00	0.00	0.00	130.00	10.00	700.00	600.00	365.00	69.69	306.13	270.00

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

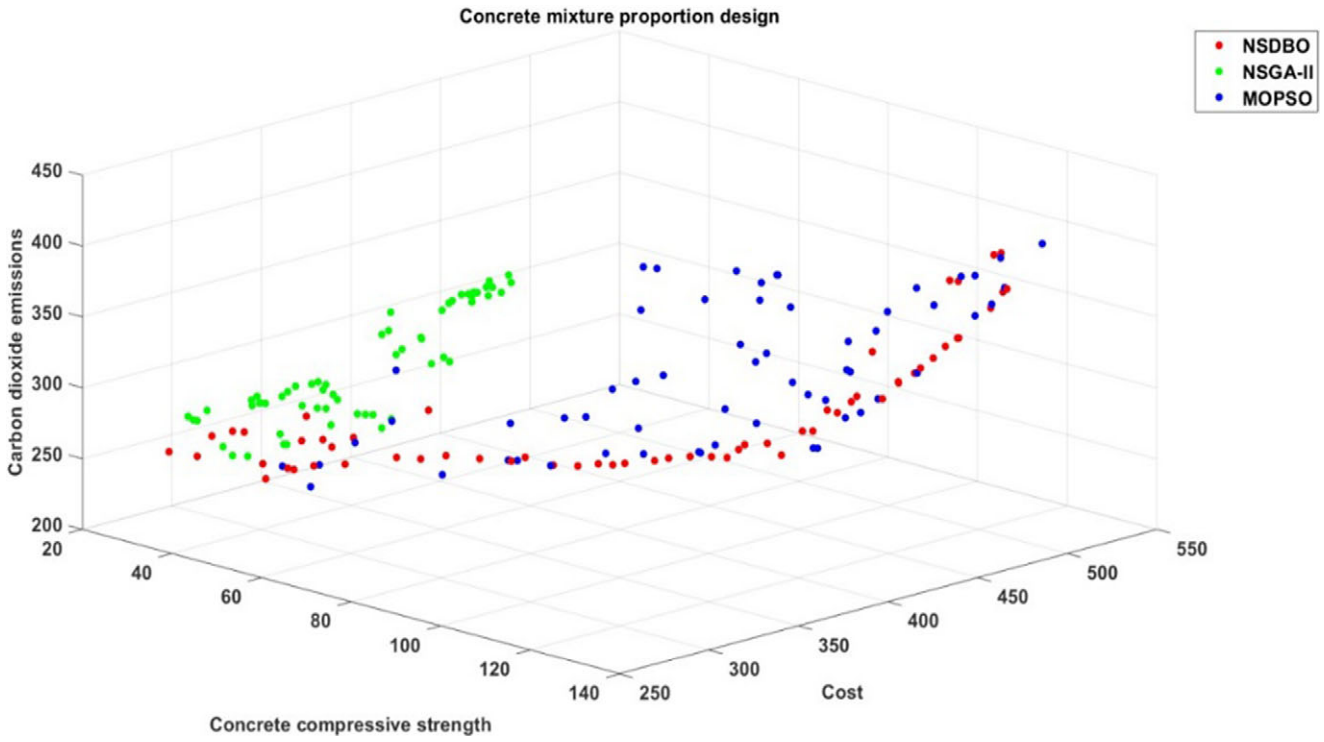


Figure 16. Algorithm comparison chart.

Table 11. Spacing metric

Algorithm	Spacing metric value
NSDBO	**8.5603**
NSGA-II	9.1125
MOPSO	10.6574

mixture proportion data to increase the generalization of the proposed approach and enhance the optimization algorithm for more efficient acquisition of the Pareto solution set.

Acknowledgements. The authors express great thanks to the financial support from the National Natural Science Foundation of China.

Author contribution. QL was involved in the supervision and carried out the review and editing process; JW drafted the manuscript; YZ was involved in the supervision and carried out the review; and YW carried out the experiments. All authors read and approved the final manuscript.

Competing interests. The authors declare none.

Funding statement. This research is funded by the National Natural Science Foundation of China, Grant number 62066005, U21A20464.

Ethical approval. This article does not contain any studies with human participants or animals performed by any of the authors.

References

Agrawal SP, Jangir P, Abualigah L, Pandya SB, Parmar A, Ezugwu AE, Arpita SA and Smerat A (2024) The quick crisscross sine cosine algorithm for optimal FACTS placement in uncertain wind integrated scenario based power systems. *Results in Engineering* **25**, 103703.

Asteris PG and Kolovos KG (2019) Self-compacting concrete strength prediction using surrogate models. *Neural Computing and Applications* **31**(Suppl 1), 409–424.

Bai X, Ma Z, Chen W, Wang S and Fu Y (2023) Fault diagnosis research of laser gyroscope based on optimized-kernel extreme learning machine. *Computers and Electrical Engineering* **111**, 108956.

Belgiu M and Drăguț L (2016) Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing* **114**, 24–31.

Benesty J, Chen J, Huang Y and Cohen I (2009) Pearson correlation coefficient. In *Noise Reduction in Speech Processing*, Springer, Heidelberg. 37–40.

Bertsimas D and Tsitsiklis J (1993) Simulated annealing. *Statistical Science* **8**(1), 10–15.

Bhadra S, Sagan V, Skobalski J, Grignola F, Sarkar S and Vilbig J (2024) End-to-end 3D CNN for plot-scale soybean yield prediction using multitemporal UAV-based RGB images. *Precision Agriculture* **25**(2), 834–864.

Buchanan S and Crawford C (2024) Probabilistic lithium-ion battery state-of-health prediction using convolutional neural networks and Gaussian process regression. *Journal of Energy Storage* **76**, 109799.

Bui DK, Nguyen T, Chou JS, Nguyen-Xuan H and Ngo TD (2018) A modified firefly algorithm-artificial neural network expert system for predicting compressive and tensile strength of high-performance concrete. *Construction and Building Materials* **180**, 320–333.

Cai J, Sun B, Wang H, Zheng Y, Zhou S, Li H, Huang Y and Zong P (2024) Application of the improved dung beetle optimizer, muti-head attention and hybrid deep learning algorithms to groundwater depth prediction in the Ningxia area, China. *Atmospheric and Oceanic Science Letters* **18**, 100497.

Chen B, Wang L, Feng Z, Liu Y, Wu X, Qin Y and Xia L (2023) Optimization of high-performance concrete mix ratio design using machine learning. *Engineering Applications of Artificial Intelligence* **122**, 106047.

Dantzig GB (2002) Linear programming. *Operations Research* **50**(1), 42–47.

Dorigo M and Di Caro G (1999) Ant colony optimization: A new meta-heuristic. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, Vol. **2**. IEEE, pp. 1470–1477.

Feng DC, Liu ZT, Wang XD, Chen Y, Chang JQ, Wei DF and Jiang ZM (2020) Machine learning-based compressive strength prediction for concrete: An adaptive boosting approach. *Construction and Building Materials* **230**, 117000.

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

Feng L, **Zhou Y**, **Luo Q and Wei Y** (2024) Complex-valued artificial hummingbird algorithm for global optimization and short-term wind speed prediction. *Expert Systems with Applications* **246**, 123160.

Fu J, **Wu C**, **Wang J**, **Haque MM**, **Geng L and Meng J** (2024) Lithium-ion battery SOH prediction based on VMD-PE and improved DBO optimized temporal convolutional network model. *Journal of Energy Storage* **87**, 111392.

Goh ATC (1995) Back-propagation neural networks for modeling complex systems. *Artificial Intelligence in Engineering* **9**(3), 143–151.

Hariri-Ardebili MA, **Mahdavi P and Pourkamali-Anaraki F** (2024) Benchmarking AutoML solutions for concrete strength prediction: Reliability, uncertainty, and dilemma. *Construction and Building Materials* **423**, 135782.

He J, **Guo W**, **Wang S**, **Chen H**, **Guo X and Li S** (2024) Application of multi-strategy based improved DBO algorithm in optimal scheduling of reservoir groups. *Water Resources Management* **38**, 1–19.

He Y, **Wang W**, **Li M and Wang Q** (2024) A short-term wind power prediction approach based on an improved dung beetle optimizer algorithm, variational modal decomposition, and deep learning. *Computers and Electrical Engineering* **116**, 109182.

Holland JH (1992) Genetic algorithms. *Scientific American* **267**(1), 66–73.

Huang H, **Yao Z**, **Wei X and Zhou Y** (2024) Twin support vector machines based on chaotic mapping dung beetle optimization algorithm. *Journal of Computational Design and Engineering* **11**(3), 101–110.

Jangir P, **Agrawal SP**, **Pandya SB**, **Parmar A**, **Kumar S**, **Tejani GG and Abualigah L** (2024) A cooperative strategy-based differential evolution algorithm for robust PEM fuel cell parameter estimation. *Ionics*, 1–39.

Jangir P, **Ezugwu AE**, **Arpita ASP**, **Pandya SB**, **Parmar A**, **Gulothungan G and Abualigah L** (2024) Precision parameter estimation in proton exchange membrane fuel cells using depth information enhanced differential evolution. *Scientific Reports* **14**(1), 29591.

Jangir P, **Ezugwu AE**, **Saleem K**, **Arpita ASP**, **Pandya SB**, **Parmar A**, **Gulothungan G and Abualigah L** (2024) A hybrid mutational northern goshawk and elite opposition learning artificial rabbits optimizer for PEMFC parameter estimation. *Scientific Reports* **14**(1), 28657.

Jangir P, **Ezugwu AE**, **Saleem K**, **Arpita ASP**, **Pandya SB**, **Parmar A**, **Gulothungan G and Abualigah L** (2024) A levy chaotic horizontal vertical crossover based artificial hummingbird algorithm for precise PEMFC parameter estimation. *Scientific Reports* **14**(1), 29597.

Jangir P and Jangir N (2017) Non-dominated sorting whale optimization algorithm (NSWOA): A multi-objective optimization algorithm for solving engineering design problems. *Global Journal of Researches in Engineering: F Electrical and Electronics Engineering* **17**, 15–42.

Jiachen H and Li-hui F (2024) Robot path planning based on improved dung beetle optimizer algorithm. *Journal of the Brazilian Society of Mechanical Sciences and Engineering* **46**(4), 1–20.

Jiang H, **Deng J and Chen Q** (2023) Olfactory sensor combined with chemometrics analysis to determine fatty acid in stored wheat. *Food Control* **153**, 109942.

Kennedy J and Eberhart R (1995) *Particle Swarm Optimization. Proceedings of ICNN'95-International Conference on Neural Networks*, Vol. **4**. IEEE, pp. 1942–1948.

Kumar P and Pratap B (2024) Feature engineering for predicting compressive strength of high-strength concrete with machine learning models. *Asian Journal of Civil Engineering* **25**(1), 723–736.

Li Y, **Sun K**, **Yao Q and Wang L** (2024) A dual-optimization wind speed forecasting model based on deep learning and improved dung beetle optimization algorithm. *Energy* **286**, 129604.

Li X, **Wang W**, **Ye L**, **Ren G**, **Fang F**, **Liu J**, **Chen Z and Zhou Q** (2024) Improving frequency regulation ability for a wind-thermal power system by multi-objective optimized sliding mode control design. *Energy* **300**, 131535.

Li P, **Zhao H**, **Gu J and Duan S** (2024) Dynamic constitutive identification of concrete based on improved dung beetle algorithm to optimize long short-term memory model. *Scientific Reports* **14**(1), 6334.

Liu Q and Jiang X (2024) Dynamic multi-objective optimization control for wastewater treatment process based on modal decomposition and hybrid neural network. *Journal of Water Process Engineering* **61**, 105274.

Liu K, **Zheng J**, **Dong S**, **Xie W and Zhang X** (2023) Mixture optimization of mechanical, economical, and environmental objectives for sustainable recycled aggregate concrete based on machine learning and metaheuristic algorithms. *Journal of Building Engineering* **63**, 105570.

Lv B, **Liu C**, **Li T**, **Meng F**, **Fu Q**, **Ji Y and Hou R** (2023) Evaluation of the water resource carrying capacity in Heilongjiang, eastern China, based on the improved TOPSIS model. *Ecological Indicators* **150**, 110208.

Mai C, **Zhang L**, **Chao X**, **Hu X**, **Wei X and Li J** (2024) A novel MPPT technology based on dung beetle optimization algorithm for PV systems under complex partial shade conditions. *Scientific Reports* **14**(1), 6471.

Mandal A and Rajput SPS (2025) Computational optimization of ceramic waste-based concrete mixtures: A comprehensive analysis of machine learning techniques. *Archives of Computational Methods in Engineering* **32**, 1–20.

Mirjalili S, **Jangir P and Saremi S** (2017) Multi-objective ant lion optimizer: A multi-objective optimization algorithm for solving engineering problems. *Applied Intelligence* **46**, 79–95.

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

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

Naderpour H, **Rafiean AH and Fakharian P** (2018) Compressive strength prediction of environmentally friendly concrete using artificial neural networks. *Journal of Building Engineering* **16**, 213–219.

Pant M, **Zaheer H**, **Garcia-Hernandez L and Abraham A** (2020) Differential evolution: A review of more than two decades of research. *Engineering Applications of Artificial Intelligence* **90**, 103479.

Pourdaryaei A, **Mohammadi M**, **Mubarak H**, **Abdellatif A**, **Karimi M**, **Gryazina E and Terzija V** (2024) A new framework for electricity price forecasting via multi-head self-attention and CNN-based techniques in the competitive electricity market. *Expert Systems with Applications* **235**, 121207.

Suthaharan S (2016) Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning. 207–235.

Tipu RK, **Panchal VR and Pandya KS** (2023) Multi-objective optimized high-strength concrete mix design using a hybrid machine learning and metaheuristic algorithm. *Asian Journal of Civil Engineering* **24**(3), 849–867.

Tu N, **Fan Z**, **Pang X**, **Yan X**, **Wang Y**, **Liu Y and Yang D** (2024) A multi-objective scheduling method for hybrid integrated energy systems via Q-learning-based multi-population dung beetle optimizers. *Computers and Electrical Engineering* **117**, 109223.

Vinod HD (1969) Integer programming and the theory of grouping. *Journal of the American Statistical Association* **64**(326), 506–519.

Wan G, **Dong Q**, **Sun X**, **Zheng H**, **Cheng M**, **Qiao W and Jia Y** (2024) Utilizing CNN to predict homogeneous thermo-mechanical properties of conductive layers for reliability numerical analysis in electronics. *Microelectronics Reliability* **157**, 115400.

Wang Z, **Huang L**, **Yang S**, **Li D**, **He D and Chan S** (2023) A quasi-oppositional learning of updating quantum state and Q-learning based on the dung beetle algorithm for global optimization. *Alexandria Engineering Journal* **81**, 469–488.

Wang D, **Ji Y**, **Xu W**, **Lu J and Dong Q** (2025) Multi-objective optimization design of recycled concrete based on the physical characteristics of aggregate. *Construction and Building Materials* **458**, 139623.

Wei X, **Bai W**, **Liu L**, **Li Y and Wang Z** (2023) A hybrid dung beetle optimization algorithm with simulated annealing for the numerical Modeling of asymmetric wave equations. *Applied Geophysics* **21**, 1–15.

Wen-Chao B, **Liang-Duo S**, **Liang C and Chu-Tian X** (2023) Monthly runoff prediction based on variational modal decomposition combined with the dung beetle optimization algorithm for gated recurrent unit model. *Environmental Monitoring and Assessment* **195**(12), 1538.

Xu D, **Li Z and Wang W** (2024) An ensemble model for monthly runoff prediction using least squares support vector machine based on variational modal decomposition with dung beetle optimization algorithm and error correction strategy. *Journal of Hydrology* **629**, 130558.

Xue H, **Li T**, **Li J**, **Zhang Y**, **Huang S**, **Li Y**, **Yang C and Zhang W** (2023) Multi-objective optimization for turning process of 304 stainless steel based on dung beetle optimizer-Back propagation neural network and improved particle swarm optimization. *Journal of Materials Engineering and Performance* **33**, 1–14.

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press

Xue J and Shen B (2020) A novel swarm intelligence optimization approach: Sparrow search algorithm. *Systems Science & Control Engineering* **8**(1), 22–34.

Xue J and Shen B (2023) Dung beetle optimizer: A new metaheuristic algorithm for global optimization. *The Journal of Supercomputing* **79**(7), 7305–7336.

Yeh IC (1998) Modeling of strength of high-performance concrete using artificial neural networks. *Cement and Concrete Research* **28**(12), 1797–1808.

Yuan Q, **Wu L**, **Huang Y**, **Guo Z and Li N** (2023) Water-body detection from Spaceborne SAR images with DBO-CNN. *IEEE Geoscience and Remote Sensing Letters*.

Zhang Y, **An M**, **Han B**, **Wang J and Wang Y** (2025) Multi-parameter mix proportion design and optimization of manufactured sand concrete based on RSM. *Structure* **73**, 108321.

Zhang J, **Huang Y**, **Wang Y and Ma G** (2020) Multi-objective optimization of concrete mixture proportions using machine learning and metaheuristic algorithms. *Construction and Building Materials* **253**, 119208.

Zhang B, **Pan L**, **Chang X**, **Wang Y**, **Liu Y**, **Jie Z**, **Ma H**, **Shi C**, **Guo X**, **Xue S and Wang L** (2025) Sustainable mix design and carbon emission analysis of recycled aggregate concrete based on machine learning and big data methods. *Journal of Cleaner Production* **489**, 144734.

Zhang Z, **Zhang L**, **Liu D**, **Sun N**, **Li M**, **Faiz MA**, **Li T**, **Cui S and Khan MI** (2024) Measurement and analysis of regional water-energy-food nexus resilience with an improved hybrid kernel extreme learning machine model based on a dung beetle optimization algorithm. *Agricultural Systems* **218**, 103966.

Zhang D, **Zhang Z**, **Zhang J**, **Zhang T**, **Zhang L and Chen H** (2024) UAV-assisted task offloading system using dung beetle optimization algorithm & deep reinforcement learning. *Ad Hoc Networks* **156**, 103434.

Zhang Y, **Zhou Y**, **Zhou G and Luo Q** (2023) An effective multi-objective bald eagle search algorithm for solving engineering design problems. *Applied Soft Computing* **145**, 110585.

Zhao M, **Jiang H and Chen Q** (2023) Identification of procymidone in rapeseed oils based on olfactory visualization technology. *Microchemical Journal* **193**, 109055.

Zhu F, **Li G**, **Tang H**, **Li Y**, **Lv X and Wang X** (2024) Dung beetle optimization algorithm based on quantum computing and multi-strategy fusion for solving engineering problems. *Expert Systems with Applications* **236**, 121219.

Zhu X, **Ni C**, **Chen G and Guo J** (2023) Optimization of tungsten heavy alloy cutting parameters based on RSM and reinforcement dung beetle algorithm. *Sensors* **23**(12), 5616.

https://doi.org/10.1017/S0890060425100115 Published online by Cambridge University Press