Article

Evolutionary Optimisation for Reduction of the Low-Frequency Discrete-Spectrum Force of Marine Propeller Based on a Data-Driven Surrogate Model

Jing-Wei Jiang ^{1,2}, Yang Yang ³, Tong-Wei Ren ⁴, Fei Wang ² and Wei-Xi Huang ^{1,*} ¹ Applied Mechanics Laboratory, Department of Engineering Mechanics, Tsinghua University, Beijing 100084, China; jjw18@mails.tsinghua.edu.cn² Advanced Propulsion Technology Research Department, China Marine Development and Research Centre, Beijing 100161, China; 18811288585@163.com³ Office of Science and Technology, Nanjing University, Nanjing 210046, China; yy2012@nju.edu.cn⁴ State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210046, China; rentw@nju.edu.cn

* Correspondence: hwx@tsinghua.edu.cn; Tel.: +86-10-6279-7215

Abstract: For practical problems with non-convex, large-scale and highly constrained characteristics, evolutionary optimisation algorithms are widely used. However, advanced data-driven methods have yet to be comprehensively applied in related fields. In this study, a surrogate model combined with the Non-dominated Sorting Genetic Algorithm II-Differential Evolution (NSGA-II-DE) is applied to reduce the low-frequency Discrete-Spectrum (DS) force of propeller noise. Reduction of this force has drawn a lot of attention as it is the primary signal used in the sonar-based detection and identification of ships. In the present study, a surrogate model is proposed based on a trained Back-Propagation (BP) fully connected neural network, which improves the optimisation efficiency. The neural network is designed by analysing the depth and width of the hidden layers. The results indicate that a four-layer neural network with 64, 128, 256 and 64 nodes in each layer, respectively, exhibits the highest prediction accuracy. The prediction errors for the first order of DST, second order of DST and the thrust coefficient are only 0.21%, 5.71% and 0.01%, respectively. Data-Driven Evolutionary Optimisation (DDEO) is applied to a standard high-skew propeller to reduce DST. DDEO and a Traditional Evolutionary Optimisation Method (TEOM) obtain the same optimisation results, while the time cost of DDEO is only 0.68% that of the TEOM. Thus, the proposed DDEO is applicable to complex engineering problems in various fields.

Keywords: data-driven evolutionary optimisation; low-frequency discrete-spectrum force; propeller noise; neural network

Citation: Jiang, J.-W.; Yang, Y.; Ren, T.-W.; Wang, F.; Huang, W.-X. Evolutionary Optimisation for Reduction of the Low-Frequency Discrete-Spectrum Force of Marine Propeller Based on a Data-Driven Surrogate Model. *J. Mar. Sci. Eng.* **2021**, *9*, 18. <https://doi.org/doi:10.3390/jmse9010018>

Received: 20 November 2020

Accepted: 21 December 2020

Published: 25 December 2020

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Marine propeller noise is one of the most prominent noise components that is detectable outside a ship. The source of a marine propeller noise is an unsteady force that comprises a low-frequency Discrete-Spectrum (DS) force, caused by blade rotation in the incoming non-uniform flow, a low-frequency Broadband-Spectrum (BS) force, caused by the interaction between the blade and turbulent fluctuations, and a moderate-to-high-frequency BS force, caused by vortex shedding from the blade's trailing edge [1]. Compared to the moderate-to-high-frequency BS force, the low-frequency force exhibits high radiated energy [2], may cause considerable damage to the hull [2], has a long propagation distance [3] and can cause degradation of the marine biological environment [4]. The low-frequency BS force can be predicted and controlled by combining spectral methods with optimisation algorithms. In contrast, low-frequency DS force, which is the primary signal used in the sonar-based detection and identification of ships [2,5], has been extensively

researched to uncover its generation mechanism and develop control technologies, due to the considerable operational costs that arise from unsteady calculations.

1.1. Low-Frequency DS Force Optimisation

The potential flow (mainly the panel method [6–8]) and Computational Fluid Dynamics [9–12] (CFD) methods are the two widely used ways to predict the unsteady hydrodynamic performance of marine propellers. The DS force can be obtained with a Fourier transform. Both methods can obtain satisfactory prediction results.

An efficient approach for reducing a propeller's low-frequency DS force is to apply an appropriate skew. The optimisation of skew distribution has been extensively studied. Mautner [13] optimised the skew distribution, which is represented by either a cubic or a quadratic distribution, to reduce the blade-rate noise caused by the DS force. Subsequently, Mautner [14] developed an improved propeller-skew optimisation program by extending his previous study to include higher-order harmonics. Cai et al. [15] proposed an improved particle swarm optimisation algorithm, combined with an unsteady hydrodynamic performance prediction, to optimise a propeller skew distribution. Ebrahimi et al. [16] studied the effects of skew angles on noise, and recommended a skew angle of 45° to 60° . However, they did not study the optimisation of skew distribution, which is more important for practical applications. The optimisation mentioned above require hydrodynamic re-computation of the unsteady force at each iteration, which require large computing resources. Jiang et al. [17] decoupled the comprehensive optimisation of propeller hydrodynamic and noise performance into two separate optimisations by proposing the circumferential phase-shift effect of skew. Using the Non-dominated Sorting Genetic Algorithm II (NSGA-II), they optimised the skew distribution and reduced the first-order component of the DS force by about 10%.

1.2. Data-Driven Evolutionary Optimisation

Similar to many other engineering optimisations, the optimisation of propeller DS force involves the solution of non-convex, large-scale and highly constrained problems [18–20]. Evolutionary optimisation algorithms have been introduced as an effective means to address these optimisation problems [21,22]. By imitating the mechanism of natural selection and the laws of genetic information transmission in biological evolution, these algorithms continue iterating from the initial data to improve their current solutions, and gradually approach the optimal solution [23]. Evolutionary optimisation algorithms can effectively deal with various complex optimisation problems, but require considerable computational time. Therefore, they are limited to the situations where there is a small computational cost per validation [24].

However, in many optimisation problems, the recalculation of each iteration consumes considerable computation time and resources. To address the problem of excessive computational costs when evolutionary optimisation is applied, Data-Driven Evolutionary Optimisation (DDEO) has been proposed [25]. A common approach in DDEO is the use of a surrogate model [26,27]. The data collected from the verified simulation model are used to train the surrogate model, which exhibits a similar predictive accuracy to the simulation model but has a lower computational cost. The input data of a new optimisation problem are discriminated using the trained surrogate model in evolutionary optimisation, i.e., only the input data generating non-dominated surrogate results is used to generate the population of next iteration. The surrogate model preliminarily selects the data with the greatest likelihood of generating Pareto solutions [28], which contain all the non-dominated solutions in the entire feasible search space, and feeds these data to the simulation model to conduct simulation based on real data. Although the surrogate model is conducted on all input data, given that such a surrogate model is much more efficient than a simulation model, the overall computational cost of evolutionary optimisation is reduced.

1.3. Objective of the Present Study

The above literature review shows that all skew optimisations, except for that proposed in our previous study [17], require hydrodynamic re-computation of the unsteady force at each iteration. Moreover, mainly traditional optimisation algorithms—that is, genetic algorithms—have been used, and advanced data-driven methods have not yet been applied to this problem. Therefore, the primary objective of the present study is to apply the DDEO method to our previously proposed decoupled skew optimisation model [17]. We expect that this will reduce the scale of the calculation, thereby improving the calculation efficiency and the optimisation result.

The remainder of this paper is organised as follows. In Section 2, DDEO is introduced and its optimisation efficiency is mathematically compared with that of the NSGA-II. In Section 3, the depth and width of the hidden layer are analysed to build an efficient surrogate model for accurate prediction of the Discrete-Spectrum Thrust (DST). In Section 4, DDEO is applied to reduce the DS force of a widely used five-blade standard High-Skew Propeller (HSP) installed on the ship *Seiun-Mar* and the results are discussed in detail. Finally, conclusions are drawn in Section 5.

2. Methods

2.1. Optimisation Method

Simulation and surrogate models are two commonly applied concepts in DDEO. A simulation model is a numerical solver that can predict physical results that meet the accuracy requirements. In contrast, a surrogate model is a machine-learning model that is an approximation of the simulation model, such as a polynomial regression model, a support-vector machine or a neural network. Unlike the traditional optimisation method, which directly iterates the simulation model, a surrogate model is used in DDEO to substantially reduce the calculation required and the complexity of the problem.

According to the universal approximation theorem of neural networks [29], the DS force prediction can be theoretically fitted by a neural network-based surrogate model. The simulation model generates the datasets for training and testing the surrogate model, and the amount of training data required is determined by the complexity of the surrogate model. The more complex the surrogate model, the more training data that are required to prevent over-fitting. There are two approaches for training such a surrogate model: offline mode and online mode [30]. In offline mode, the surrogate model is not changed after the initial training, while in online mode, it is incrementally trained when obtaining more training data. Offline mode prevents upgrading of the surrogate model. Thus, online mode is used in this study as it can continuously improve the discrimination ability of the surrogate model.

Figure 1 shows the flowchart of DDEO, including the initial training process of the surrogate model (Figure 1a) and the DDEO in online mode (Figure 1b). In the surrogate model initialisation, the results calculated by the simulation model with different design parameters are used as the training data for the surrogate model. Using the trained surrogate model, we improve the Traditional Evolutionary Optimisation Method (TEOM), which is the NSGA-II-DE in the present study. The design parameters are firstly discriminated by the surrogate model and the surrogate results are preliminarily selected. The design parameters generating the non-dominated surrogate results, which are not worse than another surrogate result in all optimisation objectives, are used to generate the population of the next iteration. After some iterations, the non-dominated surrogate results generated by the final iteration input into the simulation model for further checking. Finally, the non-dominated simulation results are selected as the Pareto solutions.

Figure 1. Flowchart of data-driven evolutionary optimisation (DDEO). (a) Initial training process of surrogate model; (b) DDEO in online mode.

2.2. Optimisation Efficiency Comparison

In this subsection, the efficiency of the DDEO is compared with that of a TEOM. As the initial training process can be completed offline before solving the practical optimisation problem and the trained surrogate model can be used repeatedly in similar optimisation problems, the time required for the initial training of the surrogate model is not considered in the comparison. Moreover, to simplify the comparison process, the surrogate model in offline mode is used without considering the time consumed in the incremental training.

Let us assume that N is the number of input data combinations in the optimisation process, T_s and T_a are the times required for each iteration of the simulation model and the surrogate model, respectively, and K is the number of data combinations that must be recalculated by the simulation model after preliminary selection by the surrogate model. In addition, the time consumed by the TEOM is $T_1 = T_s \times N$ and that by the DDEO is $T_2 = T_a \times N + T_s \times K$. The condition under which the DDEO can achieve higher efficiency is $T_2 < T_1$, that is, $K < N \times (1 - T_a/T_s)$. As $T_a < T_s$ is evidently satisfied, when an appropriate value of K is considered, the DDEO can easily exhibit lower computational costs than the TEOM.

However, considering that insufficient training data lead to limited representation ability of the surrogate model, there are inevitably differences between the surrogate and simulation results. As the final Pareto solutions in the DDEO are based on surrogate results rather than simulation results, an excessive error in the surrogate results can mislead the evolutionary process. In addition, too small a value of K is likely to result in the optimal solution not being unidentified. Therefore, it is important to select a surrogate model that can achieve the desired discrimination precision by using sufficient training data. The computational cost of the DDEO is verified to be much less than that of a TEOM in the case study shown in Section 4.

2.3. Unsteady Thrust Prediction Method

To avoid performing unsteady calculations at each iteration in the simulation model, the strip method is adopted. Thus, the propeller is divided radially into several strips, as shown in Figure 2. As shown in Figure 3, two propellers are designed to evaluate the effect of skew on an unsteady thrust in the time domain: an HSP (Case 1) and a half-skew HSP (Case 2). The unsteady panel method is used to predict the unsteady thrust in these cases operating in the wake given by:

$$U(r/R, \theta) = A_0 \cdot \alpha \cdot \sin(Z \cdot \theta) + U_0, \quad (1)$$

where R is the blade radius, θ is the circumferential angle, A_0 is the pulsating velocity amplitude, Z is the blade number, U_0 is the mean velocity and α is the coefficient, defined as:

$$\alpha = \begin{cases} 1, & r \leq r_1 \\ 0, & r \geq r_2 \\ \frac{r_2 - r}{r_2 - r_1}, & r_1 \leq r \leq r_2 \end{cases}, \quad (2)$$

Figure 2. Schematic of the strip method.

Figure 3. Skew distributions in Case 1 and Case 2.

In this simulation, we set $r_1/R = 1.2$ and $r_2/R = 2.8$. A detailed explanation and validation of the unsteady panel method can be found in our previous paper [17].

Figure 4 plots the Unsteady Thrust Before Shift (UTBS) and Unsteady Thrust After Shift (UTAS) at two typical radii ($r/R = 0.7$ and 0.9). UTBS refers to the value of the unsteady thrust obtained in the original calculation, while UTAS refers to the value of the thrust obtained after applying a phase shift corresponding to the skew angle at each strip. Specifically, negative angles are shifted to the right, while positive angles are shifted to the left. The results indicate that the effect of skew is well-approximated by the phase shift. The same phenomenon is also observed in the other strips.

Thus, the unsteady thrust of the entire propeller can be obtained by the linear superposition of all strips after the phase shift according to the local skew, rather than through hydrodynamic simulation. First, the Unsteady Thrust of Each Strip (UTES) of the original propeller is calculated and recorded. Then, for a given skew, the unsteady thrust of the entire propeller is obtained by linear iteration, and the first and second orders of DST are obtained after Fourier transformation. The flowchart of the DST calculation is illustrated in Figure 5.

Figure 4. Unsteady thrust before shift (UTBS) and unsteady thrust after shift (UTAS) of Case 1 and Case 2 at (a) $r/R = 0.7$ and (b) $r/R = 0.9$.

Figure 5. Flowchart of discrete-spectrum thrust (DST) calculation.

3. Neural Network Design

Theoretically, a large hidden layer can represent any function; however, the structure of a hidden layer and the number of nodes need to be designed in real applications. In this section, a Back-Propagation (BP) fully connected neural network is designed for use as the surrogate model by analysing the effects of the number of layers and nodes on its predictive accuracy.

The DST is predicted as four inputs for the fitting function of three outputs. The four inputs are the coefficients (A_i ($i = 1, 2, 3, 4$)) of a fourth-order polynomial that is used to reconstruct the true skew distribution with high precision while ensuring smoothness, that is:

$$\text{skew}(r/R) = [A_4 \ A_3 \ A_2 \ A_1 \ A_0] \begin{bmatrix} (r/R)^4 & (r/R)^3 & (r/R)^2 & (r/R)^1 & (r/R)^0 \end{bmatrix}^T, \quad (3)$$

where A_0 is obtained by solving the equation to ensure that the initial starting point is consistent with that of the original propeller. In this study, an HSP, which is a five-blade propeller with a high skew installed on the ship *Seiun-Maru* [31], is used as the case study. Therefore, the equation of $\text{skew}(0.1972) = 0$ is used to obtain A_0 .

The three outputs are the DST at the first order (DST_1) of the blade-passing frequency, that at the second order (DST_2) and the thrust coefficient K_T , which is defined as:

$$K_T = \frac{T}{\rho n^2 D^4}, \quad (4)$$

The amplitude in the frequency domain can be obtained from the Fourier transform of the unsteady thrust of the propeller in the time domain.

The leaky Rectified Linear Unit (ReLU) [32], which is an improved version of ReLU, is used as an activation function to accelerate the convergence rate of the neural network.

Leaky ReLU overcomes the problem of a dead ReLU, which results in some neurons remaining inactivated and the corresponding parameters not being updated.

3.1. Dataset

First, a dataset containing 336,000 records is generated using the simulation model with 500 population size and 1000 times iteration. Each record contains seven elements: four inputs and three outputs. Specifically, the four inputs are the four coefficients of Equation (3), that is, A_1 to A_4 . The outputs are the corresponding first and second orders of DST, which are the optimisation goals, and the thrust coefficient K_T , which is a constraint value, as defined in Equation (4). Then, the dataset is divided into a training dataset, a validation dataset and a test dataset, comprising 202,000, 67,000 and 67,000 records, respectively. Here, the training dataset is used to train several candidate models with different structures, the validation dataset is used to select a candidate network with the best structure as the surrogate model, and the test dataset is used to test the performance of the surrogate model.

We also attempted to generate smaller datasets with the simulation model, i.e., with smaller population sizes and/or iteration times. However, these smaller datasets are not enough to train a surrogate model with satisfactory performance, because they may easily cause over-fitting when the training datasets are not large enough. In Section 4.3, we will see that the time cost using the datasets with the above sizes is much lower than that using the simulation model directly.

3.2. Neural Network Structure Analysis

We analyse the influence of different neural network structures on the performance of surrogate model. There are two key factors of neural network structure, hidden-layer depth and hidden-layer width. Though the grid search can make a comprehensive study of different neural network structures, it causes extremely high computational cost. Thus, neural networks' structures are usually designed empirically. It has been verified as a practical way to find adequate neural network structures, and the widely used neural networks, such as AlexNet [33] and VGG-16 [34], are designed empirically. Greater depth does seem to result in better generalisation for a wide variety of tasks when the training data is sufficient to avoid over-fitting [35], which suggests that using deep architectures indeed expresses a useful prior over the space of functions the model learns. Hence, we determine the depth of the neural network firstly, and then adapt the width of the neural network to further improve the prediction accuracy of the surrogate model.

Hidden-layer depth analysis. First, the depth of a network with the number of hidden layer nodes fixed at 1024 is analysed. Table 1 lists the prediction errors of hidden-layer numbers ranging from one to five after training. Here, (1024) indicates one layer with 1024 nodes, while (512-512) indicates two layers with 512 nodes each; the other expressions follow this logic. The parameters include the weights between adjacent layers (the input layer, the hidden layers and the output layer) and the biases of the hidden layers and the output layer. The errors on DST_1 , DST_2 and K_T are calculated as follows:

$$\eta = \frac{1}{k} \sum_{i=1}^k \frac{|r_i^{sur} - r_i^{sim}|}{r_i^{sim}}, \quad (5)$$

where k is the number of records in the validation dataset and r_i^{sur} and r_i^{sim} are the results of the surrogate model and the simulation model on the i th record, respectively.

The results in Table 1 show that the four layers with (128-256-512-128) exhibit the highest predictive accuracy. For DST_1 , the error is only 0.23%, and for DST_2 , it is 6.88%. Considering that the absolute value of DST_2 is considerably less than that of DST_1 , the 6.88% error of DST_2 is acceptable.

Table 1. Predictive error of different hidden-layer depths of the outputs.

Hidden Layer Number	Hidden Layer Structure	Parameters Number	DST ₁	DST ₂	K _T
1	(1024)	8195	0.70%	151.22%	0.45%
2	(512-512)	266,755	0.28%	13.01%	0.04%
3	(256-512-256)	264,963	0.40%	8.67%	0.01%
4	(128-256-512-128)	231,299	0.23%	6.88%	0.02%
5	(128-256-256-256-128)	198,531	0.26%	7.66%	0.01%

Hidden-layer width analysis. As shown above, the four hidden layers exhibit the best predictive performance. The widths of these layers are listed in Table 2. When the total number of nodes is 512, the errors for all three outputs are the smallest. The errors for DST₁, DST₂ and K_T are only 0.21%, 5.71% and 0.01%, respectively. Considering that the absolute value of DST₂ is very small ($\sim 0.1\%$ K_T), a relative error of $\sim 5\%$ indicates that the predictive accuracy meets the requirements. Therefore, we set the total number of nodes in the hidden layer as 512, and the nodes in each layer as 64, 128, 256 and 64, respectively. The structure of the neural network is shown in Figure 6.

Table 2. Predictive error of different hidden-layer widths of the outputs.

Total Nodes Number	Hidden Layer Structure	Parameters Number	DST ₁	DST ₂	K _T
256	(32-64-128-32)	14,819	0.25%	8.79%	0.01%
512	(64-128-256-64)	58,307	0.21%	5.71%	0.01%
1024	(128-256-512-128)	231,299	0.23%	6.88%	0.02%
2048	(256-512-1024-256)	921,347	0.23%	6.08%	0.01%

Figure 6. Structure of the neural network used in the surrogate model.

4. Results and Discussion

4.1. Optimisation Preparation

The optimisation presented in this study is aimed at reducing the DST of a propeller, which is the main source of radiated noise and structural vibration from a ship 2. Except for the skew, all other parameters have a limited effect on DST 17. Therefore, the skew distribution (represented by the fourth-order polynomial shown in Equation (1)) is optimised to minimise the DST. The objective is to minimise the DST₁ and DST₂ of the blade-passing frequency ($m \times nZ$, where $m = 1, 2$ and n is the blade number). To make the optimised skew distribution more practical, the following constraints are used: (i) the thrust coefficient K_T/K_{T-0} should be within $[0.99, 1.01]$, where K_{T-0} denotes the K_T of the propeller before optimisation; (ii) the maximum skew should be less than 50° ; (iii) the minimum skew

should be larger than -30° ; (iv) the radius at which the skew equals zero should be within $[0.4, 0.6]$; (v) the skew angle, expressed as

$$\theta_s = \max(\text{skew}(r/R)) - \min(\text{skew}(r/R)), \quad (6)$$

should lie between 45° and 50° ; (vi) the radius corresponding to the minimum skew should be within $[0.2, 0.6]$ and (vii) the monotonicity of the skew distribution should be specified as

$$\text{skew}(r/R) \begin{cases} \leq 0, & 0.2 \leq r/R \leq r/R_{\min} \\ \geq 0, & r/R_{\min} \leq r/R \leq 1 \end{cases} \quad (7)$$

4.2. DDEO Model

The BP fully connected neural network using the structure of (64-128-256-64) combined with the NSGA-II-DE [36] as the optimisation algorithm is proposed and built to reduce DST1 and DST2. We test the surrogate model on the test dataset, and the predictive errors on DST1, DST2 and K_T are 0.21%, 5.78% and 0.01%, respectively. The NSGA-II-DE is an improved version of the NSGA-II based on the concept of differential evolution. Note that the NSGA-II uses the binary crossover and the polynomial mutation algorithm, while the NSGA-II-DE uses the binomial distribution crossover and differential mutation algorithm in combination with differential evolution. As mentioned in Section 3, leaky ReLU 32 is used as an activation function to accelerate the convergence rate of the neural network.

The input layer contains four input values that match the first- to fourth-order coefficients (A_i , $i = 1, 2, 3, 4$) of the polynomial representation of the skew distribution defined in Equation (3). Moreover, the output layer is set with three parameters as optimisation objectives, namely DST₁, DST₂ and constraint K_T . The other constraints can be obtained through a simple computation without affecting the overall computational efficiency.

In multi-objective optimisation, the fitness of individuals should consider not only the target values but also the constraints of the penalty function. The penalty function is designed as the sum of the constraints not satisfied by the individuals. The fittings of the results that do not meet the constraints will be substantially reduced. The full setup details of the optimisation are listed in Table 3. Note that DST₁₋₀ and DST₂₋₀ denote the values of the DST at the first and second orders of the blade-passing frequency, respectively, before optimisation.

Table 3. Design parameters, constraints and objectives of optimisation.

Objectives							
Parameter Target	DST ₁ /DST ₁₋₀ minimum			DST ₂ /DST ₂₋₀ minimum			
Constraints							
Parameter Constraints	K_T/K_{T-0} [0.99, 1.01]	$\max(\text{skew}(r/R))$ ≤ 50	$\min(\text{skew}(r/R))$ ≥ -30	$r _{\text{skew} = 0}/R$ [0.4, 0.6]	θ_s [45.0, 50.0]	$skew'$ Equation (4)	$skew(0.1972)$ 0
Design Parameters							
Parameter Range	A_{Si} [−3000, 3000]						

4.3. Case Study

The proposed DDEO model is tested and compared with the TEOM, which is NSGA-II-DE in the present study, through a case study based on an HSP. The average inflow velocity (V_A) is 9 kn and the rotational speed (n) is 90.7 rpm. The detailed geometric parameters of HSP 31 are listed in Table 4.

Table 4. Geometric parameters of the high-skew propeller.

Blade No.	D/m	$P/D \downarrow_{r/R=0.7}$	Hub/D	Rake	Skew	Section Profile
5	3.6	0.944	0.197	−3.0	45.0	Modified SRI-B

In this optimisation, the population size and iteration number for both methods are 1000 and 1500, respectively. The optimised design parameters, that is, the polynomial coefficients, are listed in Table 5 and the distributions of the optimised skew are shown in Figure 7. Compared to the skew distribution of the HSP, the minimum skew of both optimisation methods decreases, while the maximum value increases. Furthermore, the radius at which $skew(r/R) = 0$ increases for both methods, with a slightly larger skew angle. The values of K_T for HSP and for these two optimisation methods over one rotation cycle are calculated using the unsteady panel method, and compared in Figure 8. A detailed introduction of these values and their comparison with the test data of the unsteady panel method have been presented by Jiang et al. [17]. The amplitude in Figure 8, as well as the values of DST_1 and DST_2 in Table 3, demonstrate that both optimisation methods substantially reduce the DST, by more than 25% for DST_1 and ~50% for DST_2 . In addition, the time cost of the DDEO is only 0.68% that of the TOEM, and the DDEO obtains a better solution for DST_1 , as shown in Table 5. The time cost of generating 336,000 records is 36.57 min, and the time cost of training the surrogate model is 24.19 min. The total time cost of DDEO, including generating records, training the surrogate model and carrying out DDEO, is 62.14 min, which is lower than that of TOEM. Considering that the trained surrogate model can be repeatedly used, DDEO is much more efficient than TOEM.

Table 5. Results of optimisation.

Objectives						
Parameter	DST ₁ /DST ₁₋₀	DST ₂ /DST ₁₋₀				
HSP	1.00	0.11				
DDEO	0.715	0.053				
TEOM	0.723	0.041				
Constraints						
Parameter	K _T /K _{T-0}	max(<i>skew</i> (<i>r</i> / <i>R</i>))	min(<i>skew</i> (<i>r</i> / <i>R</i>))	<i>skew</i> (0.1972)	<i>r</i> _{skew=0} / <i>R</i>	θ _s
HSP	1	42.8	−4.4	0	0.51	47.2
DDEO	1	43.8	−6.0	0	0.60	49.8
TEOM	1	44.1	−5.3	0	0.60	49.4
Optimisation details						
	design parameters/[<i>A</i> _{<i>si</i>}] (<i>i</i> = 0, 1, 2, 3, 4)	Call simulation model			Time cost	
HSP	[25.47, −204.45, 442.26, −327.23, 106.75]	—			—	
DDEO	[−24.69, 308.69, −1269.15, 1886.69, −857.57]	≤1000			1.38	
TEOM	[−21.35, 266.61, −1092.50, 1609.88, −718.00]	1,500,000			202.80

Figure 7. Comparison of skew distributions.

Figure 8. Variation in KT in one rotation cycle.

4.4. Comparison of Online Mode and Offline Mode

In training the surrogate model, DDEO uses online mode. It trains the surrogate model with the 202,000 records firstly (as shown in Figure 1a), and incrementally trains it with the simulation results generated with the design parameters preliminarily selected by the surrogate model (as shown in Figure 1b). In our experiments, we repeat the incremental training three times, and the population size and iteration times of each incremental training are 1000 and 1500, respectively.

To validate the effectiveness of online mode, we compare it with offline mode, i.e., the surrogate model is only trained with the 202,000 records without any incremental training, which costs 18.76 min. For each incremental training, the time costs of generating the simulation results and training the surrogate model are 1.38 min and 0.43 min, respectively. Thus, the total time costs of surrogate model training in offline mode and online mode are 18.76 min and 24.19 min, respectively.

Table 6 illustrates the predictive errors of the surrogate models in offline mode and online mode on the test dataset. It shows that the surrogate model in online mode outperforms that in offline mode on the predictive errors of both DST1 and DST2.

Table 6. Comparison of the predictive errors between offline mode and online mode.

	DST ₁	DST ₂
Offline	0.22%	6.01%
Online	0.21%	5.78%

Then, these two surrogate models trained in offline mode and online mode are used in the optimisation. Table 7 illustrates the optimisation details that shows that DST_1/DST_{1-0} in online mode is lower than that in offline mode. It means that the online mode obtains a better optimisation solution.

Table 7. Comparison of optimisation details between offline mode and online mode.

	Offline	Online
DST_1/DST_{1-0}	0.719	0.715
DST_2/DST_{2-0}	0.047	0.053
design parameters $[A_{si}] (i = 0, 1, 2, 3, 4)$	$[-23.37, 292.94, -1206.50,$ $1792.32, -811.23]$	$[-24.69, 308.69, -1269.15,$ $1886.69, -857.57]$

4.5. Pareto Solution Analysis

To further understand the DDEO process, its Pareto solutions are compared with those of the TEOM, as shown in Figure 9. As can be seen, the distributions of A_1 versus A_2 and A_3 versus A_4 for both methods are almost the same, and have the same slope and are concentrated at almost the same position. This reflects the effectiveness of the DDEO method.

Figure 9. Comparison of the Pareto solutions: (a) A_1 versus A_2 ; (b) A_3 versus A_4 .

5. Conclusions

In the present study, a DDEO method was proposed and successfully applied to a non-convex, large-scale and highly constrained problems, i.e., to reduce the low-frequency DS forces of a marine propeller. A neural network was carefully designed by analysing the depth and width of the hidden layers. The optimisation efficiency of the DDEO was greatly improved while maintaining similar optimisation results as compared to those of a TEOM, the NSGA-II-DE. The main findings of the present study are summarised as follows.

- (1) Both theoretical and actual optimisation problems show that the efficiency of the DDEO far exceeds that of the TEOM. In particular, in the optimisation of the skew distribution to reduce the DST, the time consumed by the DDEO is only 0.68% of that consumed by the TEOM.

- (2) The surrogate model comprising a four-layer BP fully connected neural network exhibits the highest predictive accuracy for DST, with errors for DST_1 , DST_2 and K_T of only 0.21%, 5.78% and 0.01%, respectively.
- (3) The optimisation results from the DDEO and the TEOM are similar, while the optimised DST_1 of the DDEO is smaller than that of the TEOM. In addition, the Pareto solutions for both methods have the same distributions.

In the future, we are going to conduct several model tests in the water tunnel to verify the optimisation results.

Author Contributions: Conceptualisation, J.-W.J., T.-W.R. and W.-X.H.; methodology, J.-W.J., T.-W.R. and W.-X.H.; software, J.-W.J. and T.-W.R.; validation, T.-W.R. and Y.Y.; formal analysis, J.-W.J. and W.-X.H.; investigation, F.W.; resources, Y.Y. and F.W.; data curation, T.-W.R.; writing—original draft preparation, J.-W.J.; writing—review and editing, W.-X.H.; visualisation, J.-W.J. and Y.Y.; supervision, W.-X.H.; project administration, W.-X.H.; funding acquisition, W.-X.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant No. 11772172.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Blake, W.K. *Mechanics of Flow-Induced Sound and Vibration*; Academic Press: Cambridge, MA, USA, 1986.
2. Cheng, M. A summary of acoustic characteristics and control techniques of underwater non-cavitation thruster. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2019**, *233*, 6339–6356. [\[CrossRef\]](#)
3. Gorji, M.; Ghassemi, H.; Mohamadi, J. Calculation of sound pressure level of marine propeller in low frequency. *J. Low Freq. Noise Vib. Act. Control* **2018**, *37*, 60–73. [\[CrossRef\]](#)
4. Mousavi, B.; Rahrovi, A.A.; Kheradmand, S. Numerical simulation of tonal and broadband hydrodynamic noises of non-cavitating underwater propeller. *Pol. Marit. Res.* **2014**, *21*, 46–53. [\[CrossRef\]](#)
5. Chevalier, F.; Audoly, C. Turbulent Flow-Induced Self Noise and Radiated Noise in Naval Systems—An Industry Point of View. In *Flinovia-Flow Induced Noise and Vibration Issues and Aspects*; Springer: Cham, Switzerland, 2015; pp. 211–225.
6. Lee, J. A Potential Based Panel Method for the Analysis of Marine Propellers in Steady Flow. Ph.D. Thesis, Department of Ocean Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA, 1987.
7. Hsin, C.Y. Development and Analysis of Panel Methods for Propellers in Unsteady Flow. Ph.D. Thesis, Department of Ocean Engineering, Massachusetts Institute of Technology, Cambridge, MA, USA, 1990.
8. Hoshino, T. Hydrodynamic Analysis of Propellers in Unsteady Flow Using a Surface Panel Method. *J. Soc. Nav. Archit. Jpn.* **1993**, *74*, 71–87. [\[CrossRef\]](#)
9. Cho, Y.; Moon, Y.J. Discrete noise prediction of variable pitch cross-flow fans by unsteady Navier-Stokes computations. *J. Fluids Eng.* **2003**, *125*, 543–550. [\[CrossRef\]](#)
10. Huyer, S.A.; Beal, D.; Macumber, D.; Annaswamy, A. Unsteady vortex flows produced by trailing edge articulation. *J. Fluids Eng.* **2008**, *130*, 1–7. [\[CrossRef\]](#)
11. Tian, J.; Zhang, Z.; Ni, Z.; Hua, H. Flow-induced vibration analysis of elastic propellers in a cyclic inflow: An experimental and numerical study. *Appl. Ocean Res.* **2017**, *65*, 47–59. [\[CrossRef\]](#)
12. Muscari, R.; Dubbioso, G.; Ortolani, F.; Di Mascio, A. CFD analysis of the sensitivity of propeller bearing loads to stern appendages and propulsive configurations. *Appl. Ocean Res.* **2017**, *69*, 205–219. [\[CrossRef\]](#)
13. Mautner, T.S. *A Propeller Skew Optimisation Method*; Report No. AD-A191751; Naval Ocean System Center: San Diego, CA, USA, 1987; p. 92152.
14. Mautner, T.S. An optimisation method for the reduction of propeller unsteady forces. In Proceedings of the AIAA 26th Aerospace Sciences Meeting, Reno, NV, USA, 11–14 January 1988.
15. Cai, H.P.; Ma, C.; Chen, K.; Qian, Z.F.; Zhang, H. A numerical optimisation method to determine propeller skew distribution. *J. Ship Mech.* **2014**, *18*, 771–777.
16. Ebrahimi, A.; Razaghian, S.; Nouri-Borujerdi, Z. A comprehensive study on noise reduction methods of marine propellers and design procedures. *Appl. Acoust.* **2019**, *150*, 55–69. [\[CrossRef\]](#)

17. Jiang, J.; Qi, J.; Cai, H.; Chen, K.; Huang, W. Prediction and optimisation of low-frequency discrete- and broadband-spectrum marine propeller forces. *Appl. Ocean Res.* **2020**, *98*, 102114. [[CrossRef](#)]
18. Jin, Y.; Sendhoff, B. A systems approach to evolutionary multiobjective structural optimisation and beyond. *IEEE Comput. Intell. Mag.* **2009**, *4*, 62–76. [[CrossRef](#)]
19. Chugh, T.; Sindhya, K.; Miettinen, K.; Jin, Y.; Kratky, T.; Makkonen, P. Surrogate-assisted evolutionary multiobjective shape optimisation of an air intake ventilation system. *IEEE Congr. Evol. Comput.* **2017**, *12*, 1541–1548.
20. Sindhya, K.; Ojalehto, V.; Savolainen, J.; Niemistö, H.; Hakanen, J.; Miettinen, K. Coupling dynamic simulation and interactive multiobjective optimisation for complex problems: An APROS-NIMBUS case study. *Expert Syst. Appl.* **2014**, *41*, 2546–2558. [[CrossRef](#)]
21. Dasgupta, D.; Michalewicz, Z. *Evolutionary Algorithms in Engineering Applications*; Springer Science & Business Media: Berlin, Germany, 2013.
22. Fleming, P.J.; Purshouse, R.C. Evolutionary algorithms in control systems engineering: A survey. *Control Eng. Pract.* **2002**, *10*, 1223–1241. [[CrossRef](#)]
23. Ong, Y.S.; Nair, P.B.; Keane, A.J. Evolutionary optimization of computationally expensive problems via surrogate modeling. *AIAA J.* **2003**, *41*, 687–696. [[CrossRef](#)]
24. Ong, Y.S.; Nair, P.B.; Keane, A.J.; Wong, K.W. Surrogate-assisted evolutionary optimization frameworks for high-fidelity engineering design problems. *Knowl. Inc. Evol. Comput.* **2005**, *41*, 307–331.
25. Jin, Y.; Wang, H.; Chugh, T.; Guo, D.; Miettinen, K. Data-Driven Evolutionary Optimization: An Overview and Case Studies. *IEEE Trans. Evol. Comput.* **2018**, *23*, 442–458. [[CrossRef](#)]
26. Tabatabaei, M.; Hakanen, J.; Hartikainen, M.; Miettinen, K.; Sindhya, K. A survey on handling computationally expensive multiobjective optimisation problems using surrogates: Non-nature inspired methods. *Struct. Multidiscip. Optim.* **2015**, *52*, 1–25. [[CrossRef](#)]
27. Jin, Y. A comprehensive survey of fitness approximation in evolutionary computation. *Soft Comput.* **2005**, *9*, 3–12. [[CrossRef](#)]
28. Li, H.; Zhang, Q. Multiobjective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II. *IEEE Trans. Evol. Comput.* **2009**, *13*, 284–302. [[CrossRef](#)]
29. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [[CrossRef](#)]
30. Wang, H.; Jin, Y.; Jansen, J.O. Data-Driven Surrogate-Assisted Multiobjective Evolutionary Optimisation of a Trauma System. *IEEE Trans. Evol. Comput.* **2016**, *20*, 939–952. [[CrossRef](#)]
31. Hoshino, T. Experimental data for unsteady panel calculations and comparisons (Seiun-Marun Hsp). In Proceedings of the 22nd ITTC Propulsion Committee Propeller RANS/Panel Method Workshop, Grenoble, France, 5–6 April 1998.
32. Maas, A.L.; Hannun, A.Y.; Ng, A.Y. Rectifier nonlinearities improve neural network acoustic models. *Proc. Int. Conf. Mach. Learn.* **2013**, *30*, 3.
33. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the International Conference on Neural Information Processing Systems, Lake Tahoe, CA, USA, 3–8 December 2012.
34. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
35. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; The MIT Press: Cambridge, MA, USA, 2016.
36. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 1–197. [[CrossRef](#)]