



Zhang, Shenglong and Zhang, Baoji and Tezdogan, Tahsin and Xu, Leping and Lai, Yuyang (2017) Computational fluid dynamics-based hull form optimization using approximation method. Engineering Applications of Computational Fluid Mechanics. pp. 1-15. ISSN 1994-2060 , <http://dx.doi.org/10.1080/19942060.2017.1343751>

This version is available at <https://strathprints.strath.ac.uk/61227/>

Strathprints is designed to allow users to access the research output of the University of Strathclyde. Unless otherwise explicitly stated on the manuscript, Copyright © and Moral Rights for the papers on this site are retained by the individual authors and/or other copyright owners. Please check the manuscript for details of any other licences that may have been applied. You may not engage in further distribution of the material for any profitmaking activities or any commercial gain. You may freely distribute both the url (<https://strathprints.strath.ac.uk/>) and the content of this paper for research or private study, educational, or not-for-profit purposes without prior permission or charge.

Any correspondence concerning this service should be sent to the Strathprints administrator: strathprints@strath.ac.uk

Computational fluid dynamics-based hull form optimization using approximation method

Shenglong Zhang ^a, Baoji Zhang^b, Tahsin Tezdogan ^c, Leping Xu^a and Yuyang Lai^d

^aMerchant Marine College, Shanghai Maritime University, Shanghai, China; ^bCollege of Ocean Environment and Engineering, Shanghai Maritime University, Shanghai, China; ^cDepartment of Naval Architecture & Ocean and Marine Engineering, University of Strathclyde, Glasgow, UK; ^dTechnique Department, Beijing Soyotec Co., Ltd., Beijing, China

ABSTRACT

With the rapid development of the computational technology, computational fluid dynamics (CFD) tools have been widely used to evaluate the ship hydrodynamic performances in the hull forms optimization. However, it is very time consuming since a great number of the CFD simulations need to be performed for one single optimization. It is of great importance to find a high-effective method to replace the calculation of the CFD tools. In this study, a CFD-based hull form optimization loop has been developed by integrating an approximate method to optimize hull form for reducing the total resistance in calm water. In order to improve the optimization accuracy of particle swarm optimization (PSO) algorithm, an improved PSO (IPSO) algorithm is presented where the inertia weight coefficient and search method are designed based on random inertia weight and convergence evaluation, respectively. To improve the prediction accuracy of total resistance, a data prediction method based on IPSO-Elman neural network (NN) is proposed. Herein, IPSO algorithm is used to train the weight coefficients and self-feedback gain coefficient of ElmanNN. In order to build IPSO-ElmanNN model, optimal Latin hypercube design (Opt LHD) is used to design the sampling hull forms, and the total resistance (objective function) of these hull forms are calculated by Reynolds averaged Navier–Stokes (RANS) method. For the purpose of this article, this optimization framework has been employed to optimize two ships, namely, the DTMB5512 and WIGLEY III, and these hull forms are changed by arbitrary shape deformation (ASD) technique. The results show that the optimization framework developed in this study can be used to optimize hull forms with significantly reduced computational effort.

ARTICLE HISTORY

Received 27 November 2016
Accepted 14 June 2017

KEYWORDS

Hull forms optimization; approximate method; IPSO-Elman neural network; optimal Latin hypercube design; arbitrary shape deformation

1. Introduction

In recent years, hull form optimization has gained great interest for the purpose of minimizing the total resistance which results in minimizing the running cost. At the end of the 1990s, the optimization theory was introduced into the field of hull form design by potential flow theory. Dawson method was used to calculate the sum of flat friction resistance and wave-making resistance which was as the objective function to design the ship-type (Kim, 1995). Wave-making resistance was calculated by potential flow method, and optimal design of hull form was carried out with adjoint optimization method (Huan & Huang, 1998). Rankine source was used to calculate the wave resistance and viscous resistance of Series60, and the hull form with the lowest resistance was obtained by nonlinear programming method (Zhang & Zhang, 2015).

With the rapid development of computational fluid dynamics (CFD) technique and optimization technique, hull forms design based on simulation-based design

(SBD) technique has become the main trend in the twenty-first century (Kim & Yang, 2010; Tahara, Peri, Campana, & Stern, 2008). The CFD tools have become the main method for calculating the hull resistance and simulating the flow field, but its calculation time is rather long. In order to promote the application of SBD technique to the practical engineering, and to reduce the computational time of a typical CFD work, the application of approximate technique has become the key to the development in ship-type optimization. By processing of approximate technique, the original complex problem is turned into a relatively small approximate sub-problem, and the optimal solution of the original problem is obtained by successive approximation. The issue of predictions using the approximate model has been highlighted by some designers not only in the engineering (Chang, Feng, Liu, Zhan, & Cheng, 2012; Kamiński, 2015; Qian, Mao, Wang, & Yun, 2012) but also in the real-life (Chau & Wu, 2010; Wang, Kwokwing, Xu, & Chen,

CONTACT Shenglong Zhang  suckersands88@163.com

2015; Wu, Chau, & Li, 2009). In recent years, one of the approximate techniques widely used is the neural network, which is a kind of information processing method developed according to the enlightenment of biological nervous system. On account of the computer simulation, it can predict and analysis the data by learning, control and identification. Now, it has been applied into different kinds of optimization problems (Hu & Balakrishnan, 2005; Liu & Luo, 2005; Liu, Tian, Liang, & Li, 2015; Puig, Witczak, Nejari, Quevedo, & Korbicz, 2007; Zhang, 2016). The ElmanNN is a typical multi-layer dynamic recurrent neural network, and it has a stronger global stability and a characteristic of time-varying since adding the contest nodes to the hidden nodes of feed-forward network as time delay operator (Ding, Jia, Su, Xu, & Zhang, 2008). Besides, ElmanNN has been compared with BPNN which is derived from data prediction (Ding et al., 2008; Zhang, Hao, Kong, & Li, 2013; Zhou, Yang, & Yang, 2011), and they all found that the predicting precision and convergence rate of ElmanNN are much higher than BPNN. Li and Liang (2007) carried out a comparison of ElmanNN and RBFNN for flood freeway speed limitation and assumed that the ElmanNN has stronger adaptation and better generalization ability and can build the approximate model more accurately. On the basis of the local feedback network function, the ElmanNN can process the data more precision for nonlinear problem (Liu et al., 2015), which is of great important for the hull resistance prediction. Up to now, hull resistance has been predicted by using radial basis function (RBF) (Huang, Wang, & Yang, 2015; Huang & Yang, 2016), artificial neural networks (Couser, Mason, Mason, Smith, & Konsky, 2004), Holtrop and Mennen's method (Ortigosa, López, & García, 2009), genetic neural network (Chen & Ye, 2009), and BP neural network (Hou, Liu, & Liang, 2016). However, there are few literatures have been published for predicting hull resistance using ElmanNN. In the light of these considerations, an ElmanNN has been used into the hull form optimization in this study to approximately calculate the total resistance values.

Particle swarm optimization (PSO), a kind of intelligent optimization algorithm, was developed by Kennedy and Eberhart in 1995 (Kennedy & Eberhart, 1995). It has obtained a lot of interest in diverse optimization problems because of the advantages of fast convergence, easy implementation and simple calculation rules. Although PSO algorithm has become a relative mature method, it is easy to fall into the local optimal solution. Therefore, many researchers have put forward a variety of improvement measures to prove that the new methods are superior to traditional PSO algorithm. The hybridized technique named PSO-GA algorithm was presented by taking

the advantages of both PSO and GA algorithm for solving the nonlinear design optimization problems (Garg, 2016; Nik, Nejad, & Zakeri, 2016). The weighted particle for incorporation into the particle swarm optimization, and the enhanced particle swarm optimizer incorporating a weighted particle (EPSOWP) was developed to improve the evolutionary performance for a set of benchmark function (Li, Wang, Hsu, & Chen, 2014). Based on the random linear combination between the local best position and the global best position, the particle swarm without velocity equation (PSWV) algorithm was studied by Tungadio, Jordaan, and Siti (2016). A novel multi-sub-swarm particles warm optimization (MSSPSO) was used to find multi-solutions for multilayer ensemble pruning model by Zhang and Chau (2009). In this model, the MSSPSO algorithm was used to generate a different pruning based on previous oracle output for each layer. A more fast and accurate input variable selection (IVS) algorithm has been developed by Taormina and Chau (2015) integrating binary-coded discrete fully informed particle swarm optimization (BFIPS) and extreme learning machines (ELM). As we all know, the evaluation of the convergence is one of the most important factors for the PSO algorithm. If the convergence is decreased, the convergence rate of this algorithm will be affected heavily (Han, Li, & Wei, 2006). However, the article above did not mention how to evaluate the premature convergence. In this paper, an improved PSO (IPSO) algorithm has been developed by integrating the randomly distributed inertia weight coefficient and the evaluation of premature convergence. The performance of this new method has been evaluated by employing them in the optimization of the four mathematical functions.

Although ElmanNN has some advantages, there also exist some problems in its application. When dealing with high dimensional data, such as ship resistance values, useful information can be overwhelmed by large quantities of redundant data and relevant redundant information may take up much storage space and may consume a lot of calculation time (Ding et al., 2008). It is also easy to fall into local minimum because the gradient descent method is used to train the network parameters. By developing the improved ElmanNN (Shen et al., 2015; Song & Zhao, 2016; Zhou, Ding, & He, 2013), these problems can be overcome effectively. In this article, to improve the accuracy of the resistance prediction using ElmanNN, an effective IPSO algorithm developed is used to optimize the inertia weight coefficients and self-feedback gain coefficient of ElmanNN. Then the performance of the IPSO-ElmanNN method is compared in the prediction of the ship resistance with the original ElmanNN.

The aim of present work is to describe a practical ship hull form optimization loop using the IPSO-ElmanNN method. An arbitrary shape deformation (ASD) technique is used to alter the geometry of hull. Total resistance in calm water is selected as the objective function, and optimization is carried out at design speed. The IPSO-Elman algorithm is proposed to approximately calculate the total resistance, and the hull form is optimized using an IPSO algorithm. Finally, two ships are presented and discussed: namely, the David Taylor Model Basin (DTMB) model5512 (a ship model of the US Navy Combatant) and the WIGLEY III (a mathematical ship form widely used on the international) ships.

2. Optimizers

2.1. Particle swarm optimization (PSO) algorithm

On the D-dimensional space, the i -th particle velocity and position can be written as $V^i = (v_{i,1} v_{i,2} v_{i,3} \dots v_{i,d})$ and $X^i = (x_{i,1} x_{i,2} x_{i,3} \dots x_{i,d})$, respectively. $pbest$ is used to denote the local best solution and $gbest$ is used to denote the global optimal solution. In each of iteration, the particle updates itself by tracking $pbest$ and $gbest$. After finding these two optimal values, the velocity and position of each particle is updated using the following formulas:

$$v_{i,j}(t+1) = \omega v_{i,j}(t) + c_1 r_1 [p_{i,j} - x_{i,j}(t)] + c_2 r_2 [p_{g,j} - x_{i,j}(t)] \quad (1)$$

$$x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1) \quad j = 1, 2, \dots, d \quad (2)$$

where ω is the inertia weight coefficient, c_1 and c_2 are the acceleration coefficients with positive values called cognitive and social parameters, respectively. r_1 and r_2 are the random variables between 0 and 1. Generally, the range of the total number of particles N , c_1 and c_2 are reported as follows: $20 \leq N \leq 40$; $c_1 + c_2 \leq 4$ (Azimifar & Payan, 2016).

2.2. Improved particle swarm optimization (IPSO) algorithm

The inertia weight coefficient ω is a parameter to influence the trade-off between global and local searches within the solution domain. In order to improve the precision of the PSO algorithm, the adjustment of ω will become very important. Large ω can avoid getting into local optimal solution, and small one can effectively accelerate convergence speed of iteration. It can be calculated by constant method, linear decrement method,

self-adaptation method and so on. In this paper, ω is subject to a random distribution, so that it can overcome the instability caused by the linear decline of the two aspects.

The formula of ω can be written as:

$$\omega = \psi + \sigma * N(0,1) \quad (3)$$

$$\psi = \psi_{\min} + (\psi_{\max} - \psi_{\min}) * rand(0,1) \quad (4)$$

where ψ is the mean value of random weight, σ is the variance of random weight. $N(0,1)$ is the random variable of standard normal distribution. ψ_{\min} is the minimum value of random weight. ψ_{\max} is the maximum value of random weight. $rand(0,1)$ is the random variable between 0 and 1.

The convergent degree of PSO algorithm can be obtained from (Wang, Ma, Wang, & Wang, 2011):

$$\Delta = |f_g - f'_{avg}| \quad (5)$$

where Δ is the evaluation index of convergent degree, the smaller the Δ is, the better convergent it will be. f_g is the fitness of the optimal particle, f'_{avg} is the fitness average which is calculated by the particles with better fitness than f_{avg} (fitness average of the whole particles) which can be got from $f_{avg} = 1/n \sum_{i=1}^n f_i$, f_i is the i -th particle fitness, n is the size of the particle swarm.

The particle tends to premature convergence if Δ is less than threshold Δ_d and the optimal solution and the expected optimal solution f_d is not obtained. Consider the minimization problem:

$$\Delta < \Delta_d \quad (6)$$

$$f_g > f_d \quad (7)$$

At this time, partially inactive particles are modified by Gaussian mutation to early jump out of local optimum and quickly find the global optimal solution. The formal can be written as:

$$\frac{f_g - f_i}{f_g - f'_{avg}} \leq \theta \quad (8)$$

where θ is the threshold, for i -th particle satisfying the inequality (8), the mutation is carried out with random one-dimensional location:

$$x_{id}^{(k+1)} = x_{id}^{(k)} + \eta \xi \quad (9)$$

where η is the coefficient of variation, ξ is the random variable in accordance with $N(0,1)$ distribution.

The general flow chart of IPSO algorithm is as follows:

- (1) Initialize the velocities and positions of a set of particles randomly.

- (2) Update the velocities and positions of particles according to Equations (1) and (2).
- (3) Update the inertia weight coefficient ω using Equations (3) and (4).
- (4) Calculate the fitness f_i .
- (5) Update the individual extremum $pbest$ and the group extremum $gbest$.
- (6) Determine whether the premature convergence is produced by Equations (6) and (7). If the answer is yes, the population needs to mutate by Equations (8) and (9). If not, repeat the steps 2–6 until the stopping criterion is met.
- (7) Output the global optimal result.

2.3. Verification and validation for the IPSO algorithm

To verify the applicability of the IPSO algorithm, four functions are studied, as shown in Equations (10–13). PSO and IPSO algorithm are used to find the minimum value of four functions respectively. The parameters of two algorithms are shown in Table 1.

$$f_1(x) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i)^2 + (x_i - 1)^2) \quad (10)$$

$$f_2(x) = 0.5 + \frac{\sin^2 \left(\sqrt{\sum_{i=1}^D x_i^2} \right) - 0.5}{\left[1.0 + 0.001 \left(\sum_{i=1}^D x_i^2 \right) \right]^2} \quad (11)$$

$$f_3(x) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad (12)$$

$$f_4(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \left(\frac{x_i}{\sqrt{i}} \right) + 1 \quad (13)$$

After the completion of optimization, the results are tabulated in Table 2. The optimization results of the IPSO algorithm can get the global optimal solution for four functions, while PSO algorithm and IPSO algorithm with self-adaptive weight and experimental design by Li (2012) were trapped in a local optimum. It can be seen that the IPSO algorithm developed in this paper has very high-precision results in the optimization.

Figure 1 shows a comparison of the iterative processes of the optimization using the PSO and IPSO algorithms

Table 1. Parameters.

Method	N	c_1	c_2	ω	ψ_{\max}	ψ_{\min}	Δ_d	θ	η
PSO	40	2	2	0.5	–	–	–	–	–
IPSO	40	2	2	–	0.8	0.5	$1e^{-4}$	0.2	0.1

Table 2. Optimization results of different algorithms.

	D	Minimum value	PSO	IPSO	IPSO (Li, 2012)
$f_1(x)$ (Rosenbrock)	10	0	5.3067	0	4.855
$f_2(x)$ (Schaffer)	10	0	0.009716	0	0.0097
$f_3(x)$ (Rastrigrin)	10	0	16.9250	0	3.669
$f_4(x)$ (Griewank)	10	0	1.0277	0	0.0694

individually. As can be seen from the figure, in 1000 iterations, the IPSO algorithm can give a better fitness value which is near to the global optimal solution at the initial stage of optimization compared to the PSO algorithm. Although the mutation operation is added in the IPSO algorithm, the convergence speed of this algorithm is still not affected. The improving of the convergence speed is mainly because the weight coefficient is not a fixed value but a random distribution value. At the early stage of optimization, if the particle is near the global optimum, it can automatically produce a relatively small value to accelerate the convergence speed. If the optimization has got into local extremum at the early stage of optimization, the constantly change of the weight coefficient and the convergence evaluation algorithm can help to overstep the local extremum.

3. Geometry reconstruction

Based on B-spline technique, arbitrary shape deformation (ASD) technique is a method to change the shape of different models by Sculptor tool (Sun, Song, & An, 2010). It requires that the ASD volume is set up outside the body with many control points and connections. When the control points are moved, the shape of the relevant areas can be changed. The new surface can be achieved by the movements of control points, in which the continuity of 3-order can be maintained. This can insure the smoothness of the new model even under the conditions of large deformations. This direct deformation method can make it easy for the change of the complex geometric. Taking a cylinder as an example, the deformation process can be written as follows:

- (1) Build an ASD volume around the cylinder with many control points and connections between them, as shown in Figure 2. The deformation volume can be finer or coarser, depending on the shape change control desired.
- (2) Define the control parameters including the position of point and the direction of movement. As can be seen from Figure 2, point 1 and point 2 are taken as control parameters in order to change the geometry shape.

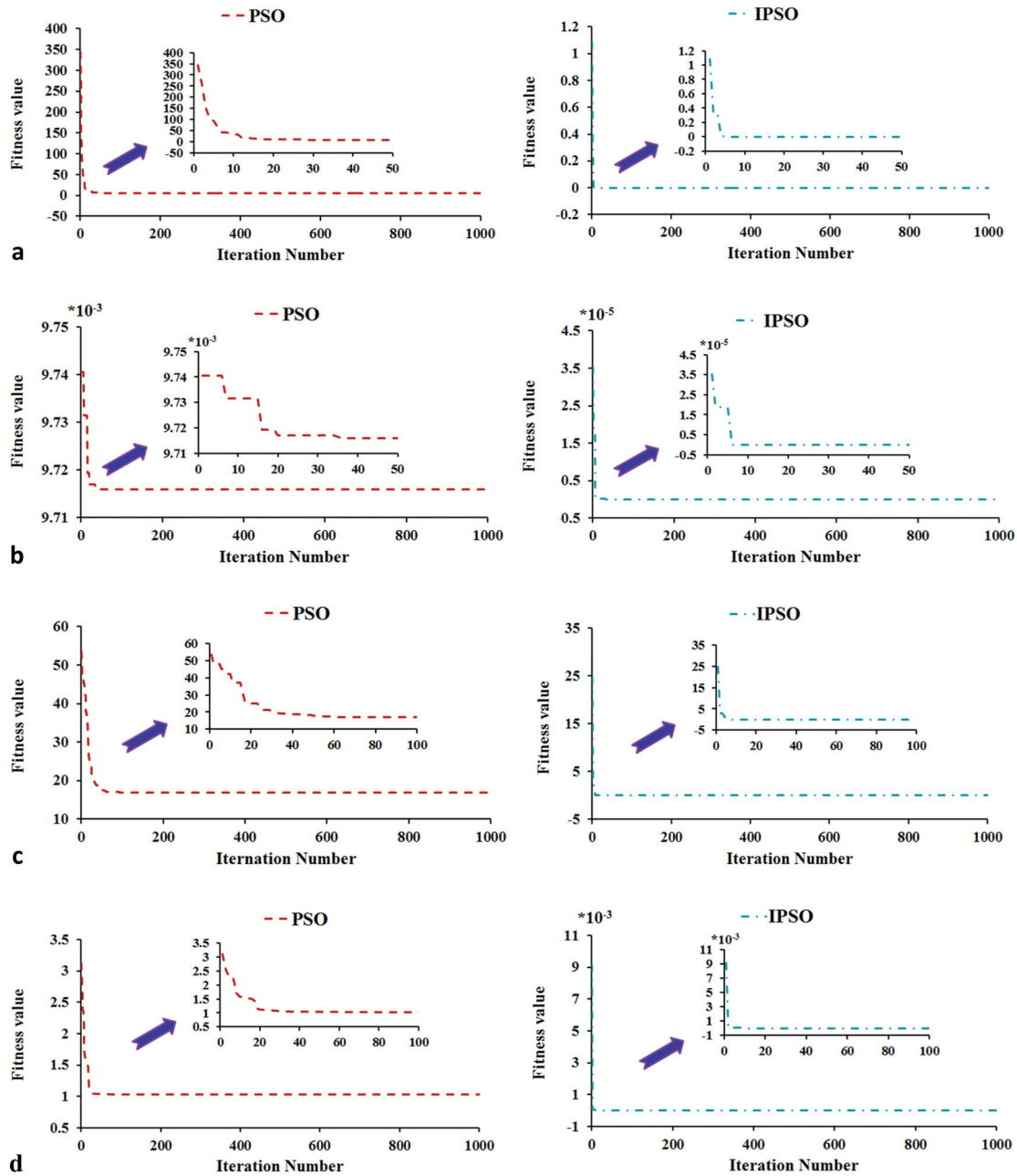


Figure 1. Convergence history of the different algorithms (a) $f_1(x)$, (b) $f_2(x)$, (c) $f_3(x)$, (d) $f_4(x)$.

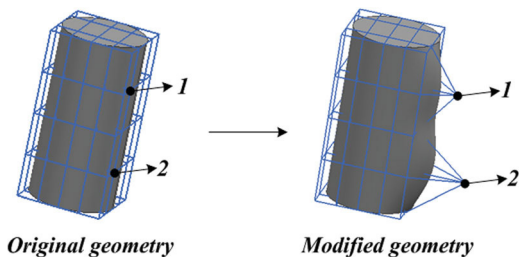


Figure 2. Geometry deformation using two parameters.

- (3) Freeze the ASD volume, and change control parameters by the movements of the points according to the requirements of the designers.

- (4) Get new geometry in term of geometry generation algorithm with modified control points.

4. Calculation of total resistance based on CFD

4.1. Mesh generation and boundary conditions

Overset mesh is a method (CD-Adapco, 2014) in which grids are divided for two parts (background grids and overset grids) of the model and then nested into the background grids. The background grids are static and the overset grids can be moved. The flow field information is exchanged at the interface according to the interpolation between each other. Then, the whole flow field

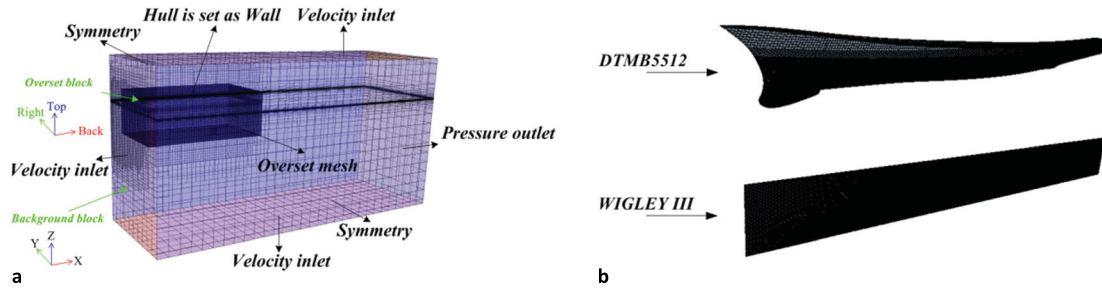


Figure 3. Grid division of computational domain and hull (a) Mesh of the computational domain, (b) Computational mesh on the ship geometries.

can be calculated (Zhao, Gao, & Xia, 2011). In this article, an overset mesh is used to divide the computational domain with linear interpolation, as shown in Figure 3. Figure 3(a) shows the mesh of the whole physical model, and the refined mesh is adopted near the free surface. Figure 3(b) shows the mesh of hull surface.

Figure 3(a) also shows the boundary conditions of the computational domain. In the light of the requirement of the overset mesh, the whole model needs two individual blocks which are named as background block and overset block. Background block is only a cuboid, and overset block is a model with Boolean subtraction between the cuboid and the hull. For the background block, the length in front, back, top, bottom and left of the hull are taken as $1.5 L_{pp}$ (L_{pp} is the length between perpendiculars of hull), $6 L_{pp}$, $1 L_{pp}$, $4 L_{pp}$ and $3 L_{pp}$, respectively. For the overset block, the length in front, back and left of the hull are taken as $1 L_{pp}$, $1.5 L_{pp}$ and $2 L_{pp}$, respectively.

- (1) Background block: front, top and bottom boundaries are selected as velocity inlet. Back boundary is selected as pressure outlet and both sides of tank are selected as symmetry plane.
- (2) Overset block: the right side of the cuboid is set as the symmetry plane and the rest of surface is set as overset mesh. A no-slip boundary condition is set for the hull geometry.

4.2. Calculation methods

In this study, Star-CCM+ is used as a RANS solver. Based on the user manual of the software (CD-Adapco, 2014), the hull is set as the fixed model and floating on the water under design draft. The total resistance in calm water is calculated by using RANS equations (Ferziger & Peric, 2002) based on the SIMPLE (Patankar & Spalding, 1972) algorithm. The RNG $k-\varepsilon$ model (Yakhot & Orszag, 1986) is selected as turbulence model. A Volume of Fluid (VOF) model (Hirt & Nichols, 1981) is selected to model and position the free surface. The whole computational domain is discretized by finite volume method.

The second-order upwind difference scheme is adopted for the convection term and the centric difference scheme for the dissipation term. The ship speed is taken as the initial velocity to start the iterative calculation.

5. The establishment of approximation method for CFD data

5.1. Optimal Latin Hypercube Design (Opt LHD)

There are a variety of experimental designs, such as central composite, orthogonal design, full factorial design and so on. Among them, the random Latin hypercube design algorithm is improved to obtain better uniform, space-filling and equilibrium, which is known as optimal Latin hypercube design (Opt LHD) algorithm (Morris & Mitchell, 1995). Matrix generating step is as follows:

m test points, n factors constitute the n^*m matrix:

$$x = [x_1, x_2, x_3, \dots, x_m] \quad (14)$$

Analyze the i -th test point:

$$x_i^T = [x_{i1}, x_{i2}, x_{i3}, \dots, x_{in}] \quad (15)$$

The Latin hypercube design algorithm is used to generate an initial design matrix according to the formula (15), and then update the design matrix through element exchange. Finally, optimal space filling is obtained by principle of max-min distance:

$$d(x_i, x_j) = d_{ij} = \left[\sum_{k=1}^n |x_{ik} - x_{jk}|^t \right]^{\frac{1}{t}} \quad (16)$$

where $t = 1$ or 2 , $1 \leq i, j \leq m$, $i \neq j$, the sampling point $d(x_i, x_j)$ is the minimum distance between x_i and x_j .

Now take 3 levels of 3 factors as example. The samples distributions are obtained by random Latin hypercube design and optimal Latin hypercube design for nine experiments, as shown in Figure 4. As can be seen from the figure, the samples are more uniform and accurate to express the characteristics of spatial distribution by the optimal Latin hypercube design.

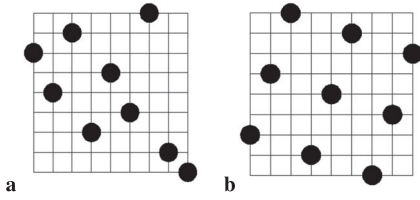


Figure 4. Comparison of samples (a) Random latin hypercube design, (b) Optimal latin hypercube design (above: Present study, below: Wang & Lu, 2015).

Based on the Opt LHD algorithm, 200 schemes are designed to calculate the total resistance, as shown in Table 3. The space distributions of samples are shown in Figure 5. The total resistance R_T is calculated by RANS-CFD method in Section 4. The total resistance coefficient C_T can be obtained using the following equation:

$$C_T = \frac{R_T}{0.5\rho v_{\text{hull}}^2 S} \quad (17)$$

where ρ is the fluid density, v_{hull} is the speed of hull, S is the wetted surface area.

In the Table 3: a_{11} , a_{12} , a_{21} , and a_{22} are the design variables of the DTMB5512 ship, which have been listed in Section 6. C_{T1} is the total resistance coefficient of the DTMB5512 ship. b_{11} , b_{12} , b_{21} , and b_{22} are the design variables of the WIGLEY III ship, which have been listed in Section 6. C_{T2} is the total resistance coefficient of WIGLEY III ship.

5.2. Elman neural network

The ElmanNN was proposed by Elman in 1990 (Elman, 1990). Its main structure is feed forward connection, which includes the input nodes, hidden nodes, context nodes and output nodes, as shown in Figure 6. The input nodes, hidden nodes and output nodes are similar to the

feed forward neural network. The input nodes unit only plays the role of signal transmission, and output nodes unit plays the role of the linear weighting. The context nodes are used to memory the previous output value of the hidden nodes and then return to the input.

Nonlinear state space equations of ElmanNN can be written as:

$$y(k) = g(w^3 x(k)) \quad (18)$$

$$x(k) = f(w^1 x_c(k) + w^2 u(k-1)) \quad (19)$$

$$x_c(k) = \beta x_c(k-1) + x(k-1) \quad (20)$$

where w^3 represents the connection weight from the hidden nodes to the output nodes, $x(k)$ means the output of the hidden nodes. w^1 represents the connection weight from the context nodes to the hidden nodes, w^2 represents the connection weight from the input nodes to the hidden nodes. $x_c(k)$ means the output of the context nodes. β is the self-feedback gain coefficient. $g(\cdot)$ represents the transfer function of the output nodes, $f(\cdot)$ represents the transfer function of the hidden nodes.

The error function can be written as:

$$E = \frac{1}{2} (y_d(k) - y(k))^T (y_d(k) - y(k)) \quad (21)$$

The dynamic learning algorithm can be summarized as follows:

$$\Delta w_{ij}^3 = \vartheta_3 \delta_i^0 x_j(k), \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n \quad (22)$$

$$\Delta w_{jq}^2 = \vartheta_2 \delta_j^h u_q(k-1), \quad j = 1, 2, \dots, n; \quad q = 1, 2, \dots, r \quad (23)$$

$$\Delta w_{jl}^1 = \vartheta_1 \sum_{i=1}^m \left(\delta_i^0 w_{ij}^3 \right) \frac{\partial x_j(k)}{\partial w_{jl}^1}, \quad j = 1, 2, \dots, n; \quad l = 1, 2, \dots, n \quad (24)$$

Table 3. Experiment samples.

No.	DTMB5512					WIGLEY III				
	a_{11}	a_{12}	a_{21}	a_{22}	$C_{T1} * 10^{-3}$	b_{11}	b_{12}	b_{21}	b_{22}	$C_{T2} * 10^{-3}$
1	-0.0201	-0.02442	0.0623	-0.0479	4.341	-0.0369	-0.5494	-0.061	-0.00275	5.536
2	-0.2533	-0.03809	0.0573	-0.0468	4.373	-0.3791	-0.5237	-0.6747	0.04178	5.206
3	-0.1065	-0.09362	-0.1148	0.0884	4.343	-0.0932	-0.7357	-0.0161	0.00361	5.577
4	-0.201	-0.02528	-0.0143	-0.0214	4.342	-0.3068	-0.2731	-0.2924	0.03976	5.369
5	-0.3075	-0.08337	-0.0721	-0.1115	4.472	-0.249	-0.0867	-0.7229	0.04294	5.297
6	-0.2432	-0.01503	-0.145	-0.019	4.341	-0.008	-0.3373	-0.1221	0.036	5.426
7	-0.3176	-0.0808	-0.0759	-0.0329	4.387	-0.3936	-0.3502	-0.1896	0.02993	5.311
8	-0.0422	-0.00392	-0.1136	0.0769	4.352	-0.1044	-0.5783	-0.1542	0.01287	5.502
9	-0.0683	-0.10131	-0.0683	-0.0075	4.497	-0.2506	-0.1382	-0.4305	-0.00969	5.402
10	-0.203	-0.13975	0.0824	0.0561	4.383	-0.3438	-0.045	-0.3181	0.04612	5.536
...
...
197	-0.2915	-0.12779	-0.057	-0.0098	4.493	-0.355	-0.4016	-0.7454	0.02212	5.211
198	-0.3437	-0.06286	0.0673	0.0792	4.361	-0.3213	-0.0257	-0.2024	0.01402	5.291
199	-0.3276	-0.07739	0.0523	0.0098	4.362	-0.1141	-0.3181	-0.6008	0.05913	5.358
200	-0.2854	-0.10387	-0.0043	0.033	4.451	-0.0273	-0.498	-0.3566	0.00448	5.502

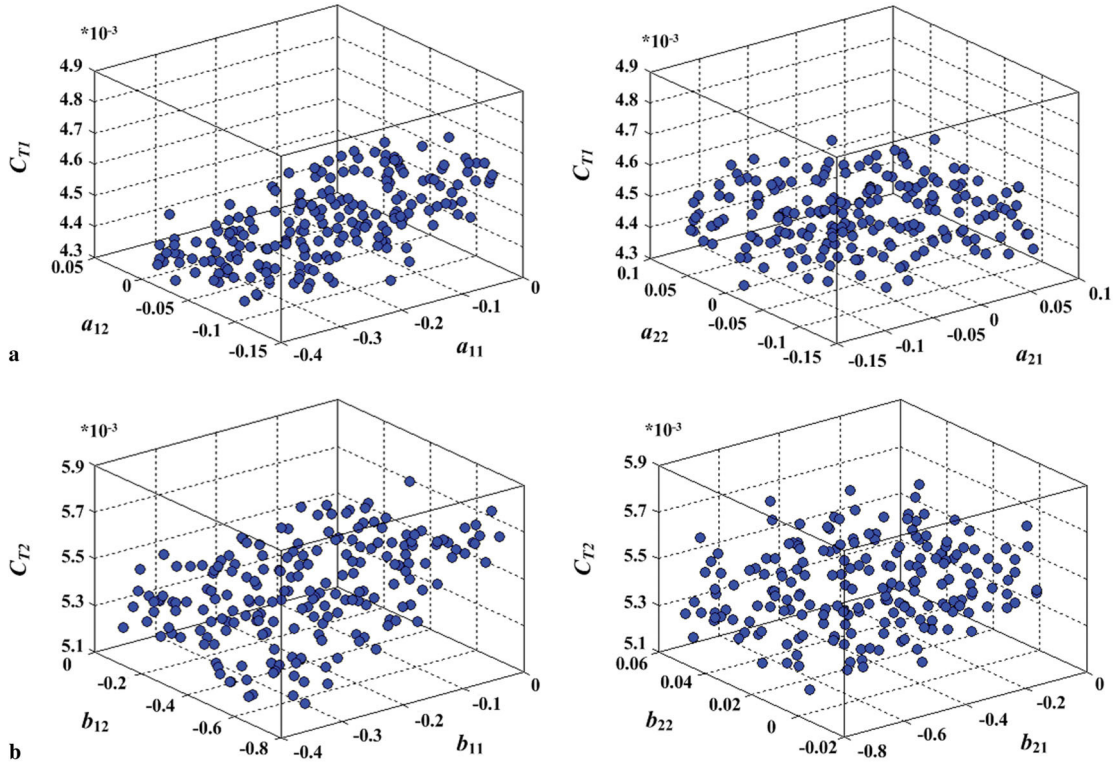


Figure 5. Space distributions of samples (a) DTMB5512, (b) WIGLEYIII.

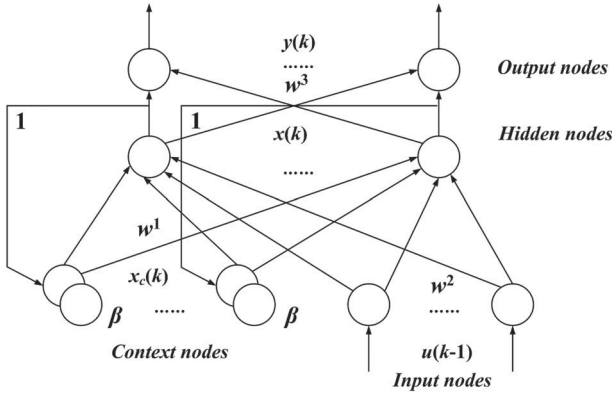


Figure 6. The framework of ElmanNN (above: Present study, below: Wang et al., 2011).

where ϑ_3 , ϑ_2 and ϑ_1 are the learning step.

$$\delta_i^0 = (y_{d,i}(k) - y(k))g_i'(\cdot) \quad (25)$$

$$\delta_j^h = \sum_{i=1}^m (\delta_i^0 w_{ij}^3) f_j'(\cdot) \quad (26)$$

$$\frac{\partial x_j(k)}{\partial w_{jl}^1} = f_j'(\cdot)x_l(k-1) + \beta \frac{\partial x_j(k-1)}{\partial w_{jl}^1} \quad (27)$$

5.3. IPSO-Elman neural network

The IPSO-ElmanNN can be expressed as follows:

- (1) First, the samples are trained according to the input and output. Secondly, the node numbers of the input nodes, hidden nodes and output nodes are designed. Finally, the topology structure of ElmanNN is defined.
- (2) Define the parameters of IPSO algorithm, which includes the population size, number of iterations, inertia factor and acceleration coefficients.
- (3) Define the evaluation function of particles. The mean square error function G is used as the fitness evaluation function of the particles. The position of the particle with the lowest fitness is the optimal solution of the model when the iteration is stopped. The fitness function can be defined by the following equation:

$$G = \frac{1}{N} \sum_{i=1}^N (y_{i(\text{Elman})} - y_{i(\text{CFD})})^2 \quad (28)$$

where $y_{i(\text{Elman})}$ is the i -th predicted value by ElmanNN, $y_{i(\text{CFD})}$ is the i -th expected output by CFD method.

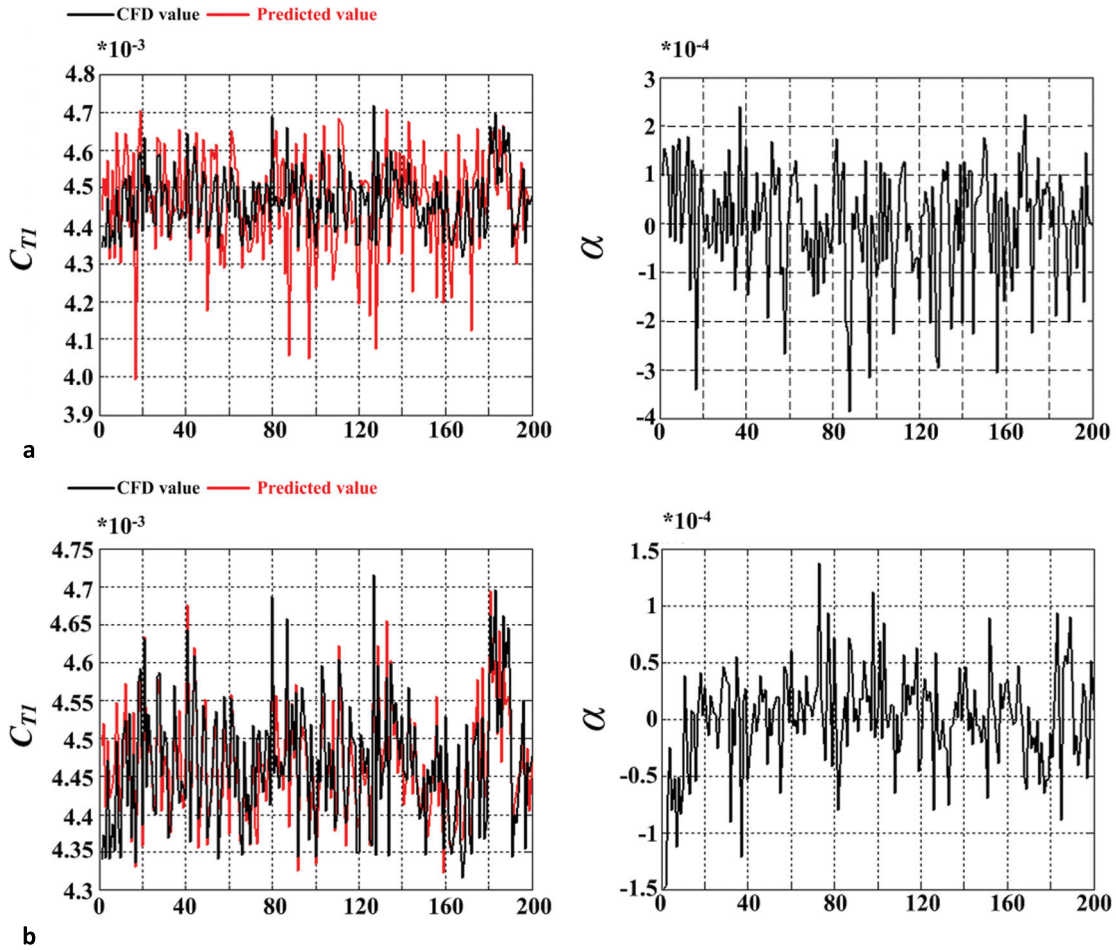


Figure 7. Prediction results of C_{T1} for DTMB5512ship (a) The prediction of the C_{T1} using ElmanNN, (b) The prediction of the C_{T1} using IPSO-ElmanNN.

- (4) The velocities and positions of a set of particles are initialized randomly.
- (5) Update the velocities and positions of particles and inertia weight coefficient ω .
- (6) Calculate the fitness using the formula (28).
- (7) Update the individual extremum $pbest$ and the group extremum $gbest$ according to the fitness of the particles.
- (8) Update the solution, and adjust the weights w^1 , w^2 , w^3 , and self-feedback gain coefficient β .
- (9) Evaluate whether the premature convergence is produced. If the answer is yes, mutate the population. If not, repeat the steps 5–9 until the final end is met.
- (10) The optimal result is used to train the weights and self-feedback gain coefficient, and then output prediction results of hull resistance.

5.4. Verification and validation for IPSO-ElmanNN

In order to test the effect of the IPSO-ElmanNN, ElmanNN and IPSO-ElmanNN prediction models are

implemented in MATLAB (R2016a) with the samples from Table 3. Two algorithms are then used to predict the total resistance subsequently. The cell numbers of input nodes, hidden nodes and output nodes are 4, 12 and 1, respectively. Figures 7 and 8 show the predictions results of the total resistance coefficients for two ships in question. α is the deviation between the ElmanNN//IPSO-ElmanNN and the CFD methods. Table 4 shows the average error results of these predictions.

When comparing the IPSO-ElmanNN with the ElmanNN for predicting the total resistance coefficients, the former has improved the prediction accuracy, not only for the DTMB5512 (with the average error about $3.2 \times 10^{-5}\%$), but also for the WIGLEY III (with the average error about $4.7 \times 10^{-5}\%$). The reason of this improved performance is that the IPSO algorithm has found a set of more suitable coefficients to train the ElmanNN in order to avoid the difficulty in choosing the coefficients through experience. Although the forecasting precision of the IPSO-Elman algorithm is preferable to the Elman algorithm, there are some errors between the CFD data and prediction data. The main reason for such errors is

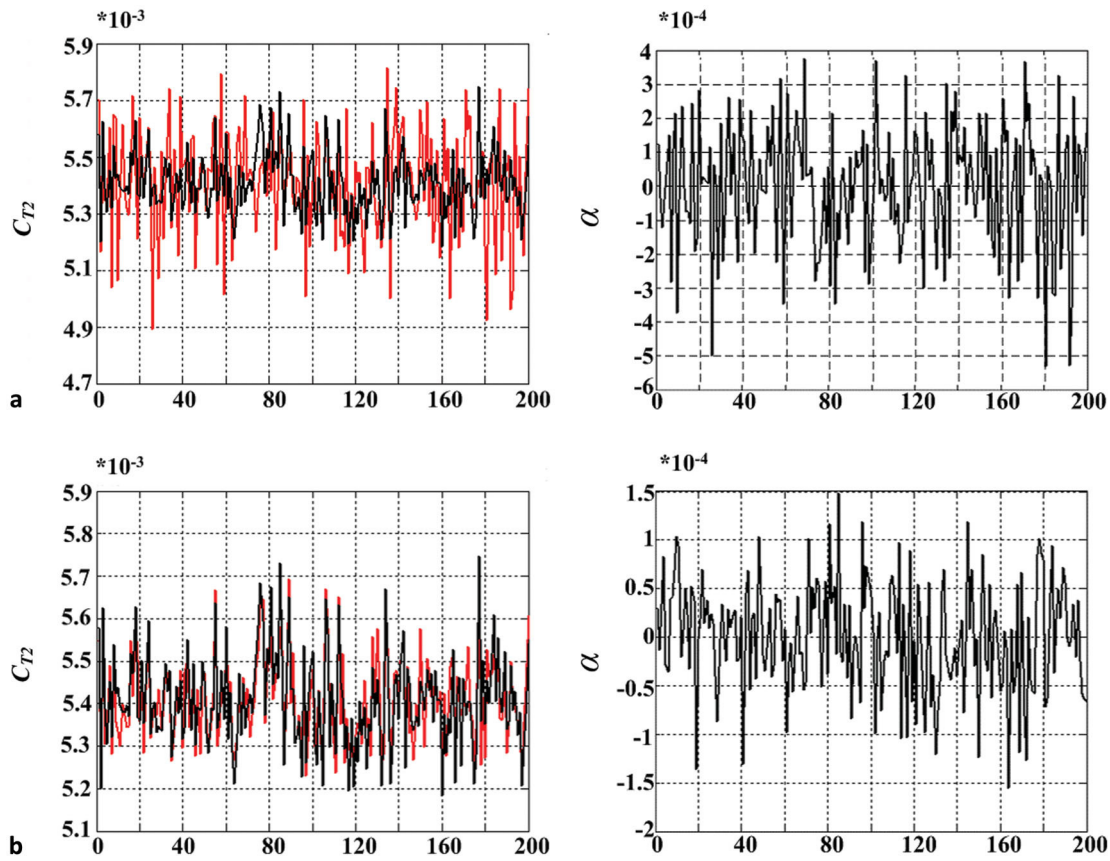


Figure 8. Prediction results of C_{T2} for WIGLEYIIIship (a) The prediction of the C_{T2} using ElmanNN, (b) The prediction of the C_{T2} using IPSO-ElmanNN.

Table 4. Result of the total resistance coefficient prediction based on the different algorithms.

Training algorithms	Average Error (%) for 200 sampling hull forms	
	DTMB5512	WIGLEY III
ElmanNN	8.89×10^{-5}	1.41×10^{-4}
IPSO-ElmanNN	3.2×10^{-5}	4.7×10^{-5}

that the number of samples is not adequate. The network training results can be improved effectively by increasing the number of training samples.

6. Hull form optimization problems

6.1. Optimize processes

In this paper, optimization process is shown in Figure 9, including the following five steps:

- (1) Change design variables using the IPSO algorithm.
- (2) On the basis of a set of design variables, change hull form shape using the ASD technique.
- (3) Calculate the displacement of each new hull and compare with the constraint. If the condition is met,

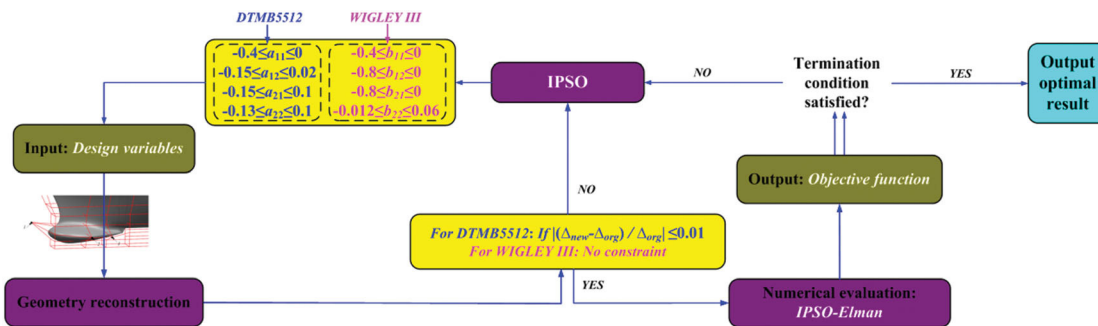


Figure 9. Flow chart of the optimization loop.



Figure 10. Modified region of hull forms.

continue to calculate, otherwise return to step 1 to recalculate the new hull.

- (4) Calculate the total resistance in calm water using IPSO-ElmanNN, and then save the result.
- (5) Repeat steps 1–4 until the stopping criterion is met. Then the new hull can be output. The whole work is completed on the ISIGHT platform (Wang, Mo, & Zhang, 2011).

6.2. Geometry

In this paper, the DTMB5512 and WIGLEY III ships are selected as optimization models. The modified area have been shown in Figure 10. The major characteristics are shown in Table 5.

6.3. Optimization strategy

The objective function is the total resistance coefficients C_{T1} (DTMB5512) and C_{T2} (WIGLEY III) in calm water at design speed. The displacement constraint has been

Table 5. Major characteristics of two ships.

Hull forms	DTMB5512	WIGLEY III
Design speed Fr	0.28	0.3
Length between perpendiculars L_{pp} (m)	3.048	3
Draft d (m)	0.132	0.1875
Breath B (m)	0.409	0.3
Displacement Δ (kg)	86.4	78
Block Coefficient C_B	0.507	0.4622

defined in formula (29) (only for DTMB5512). The design variables, a_{11} , a_{12} , a_{21} and a_{22} (DTMB5512) and b_{11} , b_{12} , b_{21} and b_{22} (WIGLEY III) have been shown in Table 6.

$$\left| \frac{\Delta_{\text{new}} - \Delta_{\text{org}}}{\Delta_{\text{org}}} \right| \leq 0.01 \quad (29)$$

where Δ_{new} represents the variable of new hull, and Δ_{org} represents the variable of parent hull.

In the Table 6: No.1 to No.6 are the positions of design variables, as shown in Figure 11.

6.4. Results and discussion

Since the calculation of the total resistance costs less than two minute under the help of the IPSO-ElmanNN, the optimization efficiency has been greatly improved compared with CFD runs optimization loop. After the completion of the optimization, the excellent hull forms

Table 6. Control parameters of two ships.

Hull forms	Design variables	No.	Moving directions	Constraints
DTMB5512	a_{11}	1	X	$-0.4 \leq a_{11} \leq 0$
	a_{12}	1	Z	$-0.15 \leq a_{12} \leq 0.02$
	a_{21}	2	Y	$-0.15 \leq a_{21} \leq 0.1$
	a_{22}	3	Y	$-0.13 \leq a_{22} \leq 0.1$
WIGLEY III	b_{11}	4	X	$-0.4 \leq b_{11} \leq 0$
	b_{12}	5	Y	$-0.8 \leq b_{12} \leq 0$
	b_{21}	6	Y	$-0.8 \leq b_{21} \leq 0$
	b_{22}	4	Z	$-0.012 \leq b_{22} \leq 0.06$

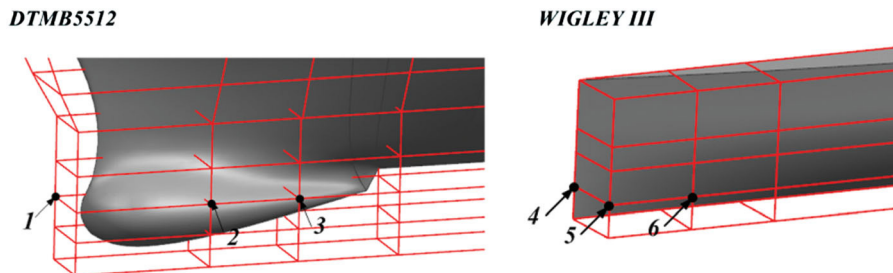


Figure 11. Control positions of two ships.

Table 7. Resistance results of the optimized hulls.

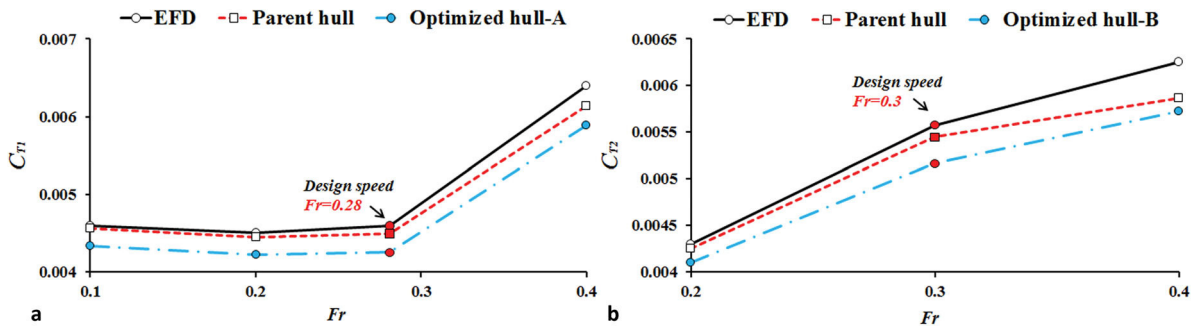
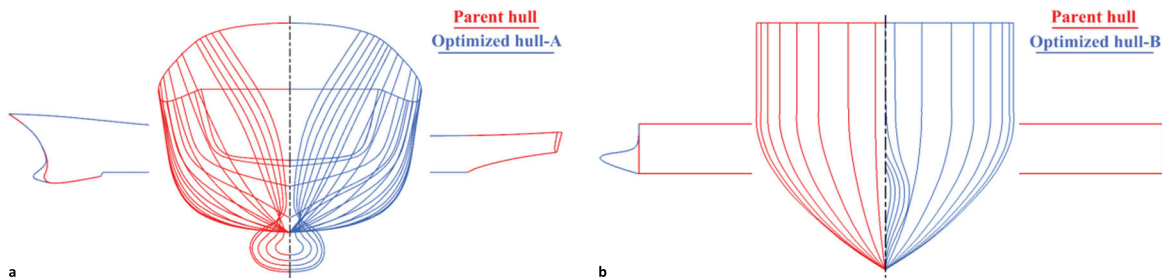
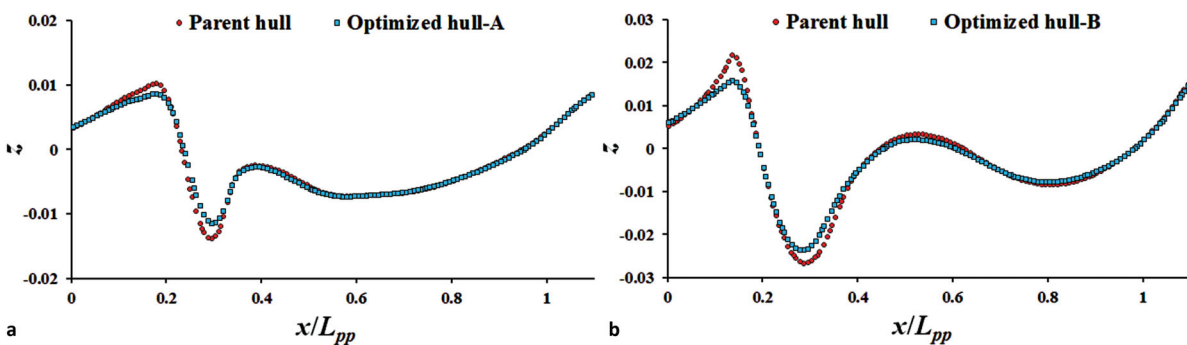
Method	Parent hulls	Optimized hulls	Fr	C_{Torg}/C_{Topt}
IPSO-ElmanNN + IPSO	DTMB5512	Optimized hull-A	0.28	1.0591
	WIGLEY III	Optimized hull-B	0.3	1.05468

with lower total resistance are obtained. The work is carried out on the Intel Corei5-5200U CPU @2.2 GHz, and the CFD runs in this paper are using the ARCHIE-WeSt Height Performance Computer (<http://www.archie-west.ac.uk>). Table 7 shows the comparison of the obtained optimization results. The results clearly state that the total resistance of the optimized hull-A is decreased by 5.58%, and the optimized hull-B is decreased by 5.19%. Following this, further calculations are carried out for other speeds and comparison is made against the parent hull by RANS-CFD results and the EFD data obtained from Iowa Institute of Hydraulic Research (Gui, Longo, & Stern, 2001a,2001b; Longo & Stern, 2005) and Delft University

of Technology (1992), as shown in Figure 12. For the optimized hull-A, C_{T1} decreases at all speeds especially in design speed and high speed, and for the optimized hull-B, C_{T2} decreases at all speeds especially in the design speed.

Figure 13 shows the comparison of the hull lines for parent hull and optimized hulls. Figure 14 shows the comparison of longitudinal wave cut for parent hull and optimized hulls along the $y/L_{pp} = 0.082$ plan (z represents the height of free surface). It can be found that the amplitude of waves has been reduced which indicates the reduction in total resistance for the optimized hulls.

Figure 15 presents the wave patterns for the parent hull and the optimized hulls. As the change of the bow shape for both of the ships, the wave patterns in the forward shoulder have been reduced significant, while the change of the shoulder waves and the stern waves are not very

**Figure 12.** The total resistance coefficients for a range of Fr numbers (a) DTMB5512, (b) WIGLEYIII.**Figure 13.** Comparison of hull lines (a) DTMB5512, (b) WIGLEYIII.**Figure 14.** Comparison of wave profile at $y/L_{pp} = 0.082$ (a) DTMB5512, (b) WIGLEYIII.

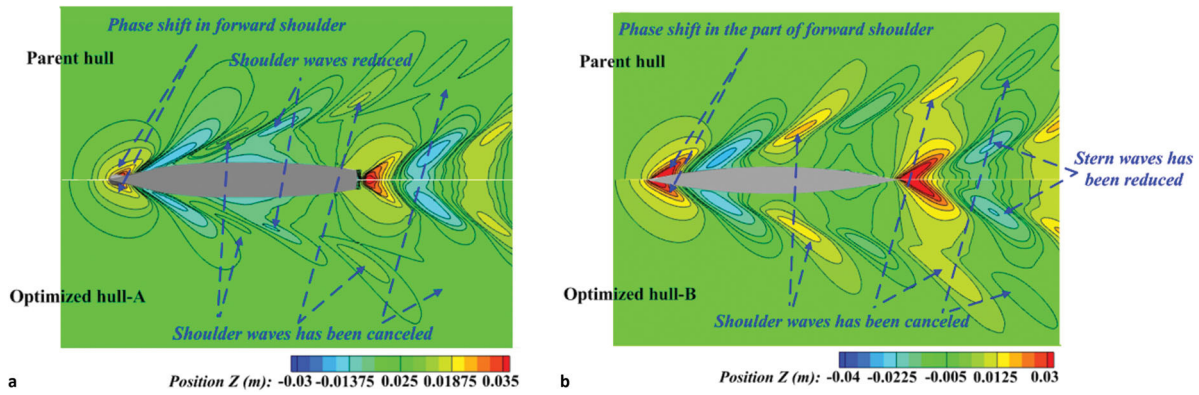


Figure 15. Comparison of the wave patterns around the vessels (a) DTMB5512, (b) WIGLEYIII.

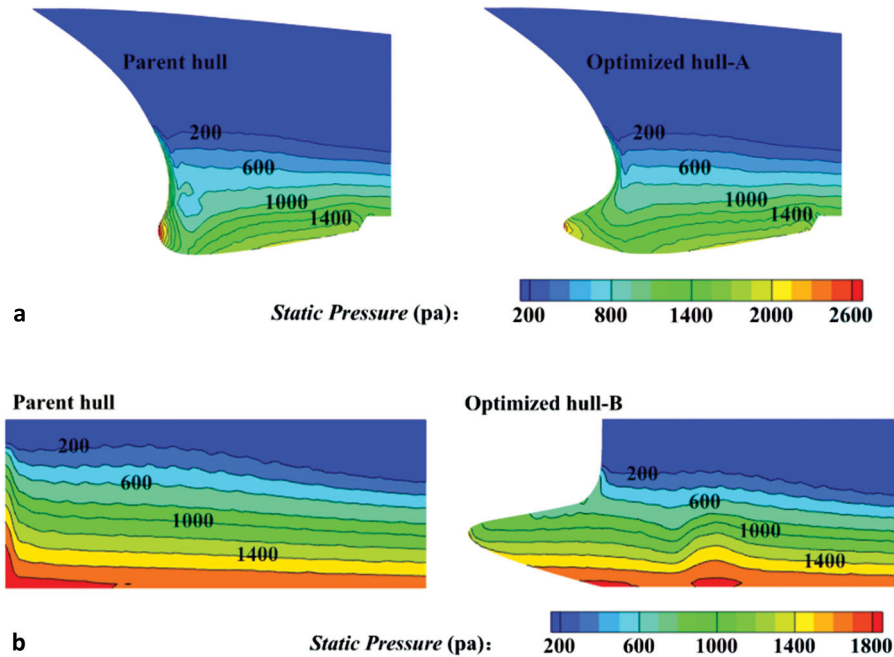


Figure 16. Comparison of the static pressure on the ship surfaces (a) DTMB5512, (b) WIGLEYIII.

significant. Figure 16 is the comparison of the static pressure on hull surface. The new hull forms have changed the pressure distribution near the bow, and wave-resistance has been decreased which ends the decrease of the total resistance.

7. Conclusion

- (1) In order to overcome the shortcomings of the PSO algorithm, an IPSO algorithm has been proposed with random weighting method and the evaluation of convergence. Through the test of four mathematical functions, it can be concluded that IPSO algorithm developed in this paper can effectively improve the optimization accuracy.
- (2) ElmanNN parameters are trained by IPSO algorithm, and IPSO-ElmanNN prediction model is proposed.

Then ElmanNN and IPSO-ElmanNN are carried out on the prediction of the hull resistance. The results indicate that the IPSO-ElmanNN has the advantages of precision and stability.

- (3) Aiming to deal with the challenge of larger surface variation, the ASD technique is used to change the shape of geometry. The new model can be quickly obtained while ensuring the smoothness of surfaces. Because of less design variables, it also can improve the efficiency of ship-type optimization.
- (4) In terms of hull form optimization problem, IPSO-ElmanNN-based approximate optimization approach is developed and then two ships have been applied in this loop. It is concluded that the approximate optimization system is feasible and rational for optimization of hull forms which can provide relevant technical support in ship-type design.

- (5) Since the optimization loop in this paper is developed for single objective function (minimum total resistance in calm water), and the seakeeping performances are not taken into account. To obtain a hull form with a better performance, future work will force on the multi-objective optimization including rapidity, seakeeping and manipulation.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

The study in this paper was supported by the grants of the National Science Foundation of China, (No. 51009087) and the grants from the National Science Foundation of Shanghai (No.14ZR1419500). Results were obtained using the EPSRC funded ARCHIE-WeSt High Performance Computer (www.archie-west.ac.uk). EPSRC grant no. EP/K000586/1.

ORCID

Shenglong Zhang  <http://orcid.org/0000-0002-9348-4395>

Tahsin Tezdogan  <http://orcid.org/0000-0002-7032-3038>

References

- Azimifar, A., & Payan, S. (2016). Enhancement of heat transfer of confined enclosures with free convection using blocks with PSO algorithm. *Applied Thermal Engineering*, 101, 79–91. doi:10.1016/j.applthermaleng.2015.11.122
- CD-Adapco. (2014). User guide STAR-CCM+ Version 9.0.2.
- Chang, H. C., Feng, B. W., Liu, Z. Y., Zhan, C. S., & Cheng, X. D. (2012). Research on application of approximate model in hull form optimization. *Shipbuilding of China*, 53(1), 88–98. (In Chinese).
- Chau, K. W., & Wu, C. L. (2010). A hybrid model coupled with singular spectrum analysis for daily rainfall prediction. *Journal of Hydroinformatics*, 12(4), 458–473. doi:10.2166/hydro.2010.032
- Chen, A. G., & Ye, J. W. (2009). *Research on the genetic neural network for the computation of ship resistance*. International conference on computational intelligence and natural computing, Japan.
- Couser, P., Mason, A., Mason, G., Smith, C. R., & Kinsky, B. R. V. (2004). *Artificial neural networks for hull resistance prediction*. At Siguenza, Spain: Compit.
- Ding, S. F., Jia, W. K., Su, C. Y., Xu, X. Z., & Zhang, L. W. (2008). PCA-Based elman neural network algorithm. *Tien Tzu Hsueh Pao/Acta Electronica Sinica*, 5370(2), 315–321. doi:10.1007/978-3-540-92137-0_35
- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14(2), 179–211.
- Ferziger, J. H., & Peric, M. (2002). *Computational methods for fluid dynamics* (3rd ed.). Berlin, Gsssermany: Springer.
- Garg, H. (2016). A hybrid PSO-GA algorithm for constrained optimization problems. *Applied Mathematics and Computation*, 274(11), 292–305. doi:10.1016/j.amc.2015.11.001
- Gui, L., Longo, J., & Stern, F. (2001a). Biases of PIV measurement of turbulent flow and the masked correlation-based interrogation algorithm. *Experiments in Fluids*, 30(1), 27–35. doi:10.1007/s003480000131
- Gui, L., Longo, J., & Stern, F. (2001b). Towing tank PIV measurement system, data and uncertainty assessment for DTMB model 5512. *Experiments in Fluids*, 31(3), 336–346. doi:10.1007/s003480100293
- Han, J. H., Li, Z. R., & Wei, Z. C. (2006). Adaptive particle swarm optimization algorithm and simulation. *Journal of System Simulation*, 18(10), 2969–2971. (In Chinese).
- Hirt, C. W., & Nichols, B. D. (1981). Volume of fluid method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1), 20–225. doi:10.1016/0021-9991(81)90145-5
- Hou, Y. H., Liu, F., & Liang, X. (2016). Minimum resistance hull form optimization design based on IPSO-BP neural network. *Journal of Shanghai Jiaotong University*, 50(8), 1193–1199. (In Chinese).
- Hu, Z., & Balakrishnan, S. N. (2005). Parameter estimation in nonlinear systems using hopfield neural networks. *Journal of Aircraft*, 42(1), 41–53. doi:10.2514/1.3210
- Huan, J., & Huang, T. T. (1998). *Sensitivity analysis methods for shape optimization in nonlinear free surface flow*. Proceedings 3rd osaka colloquium on advanced CFD applications to ship flow and hull form design, Japan.
- Huang, F., Wang, L., & Yang, C. (2015). *Hull form optimization for reduced drag and improved seakeeping using a surrogate-based method*. International Society of Offshore and Polar Engineers.
- Huang, F. X., & Yang, C. (2016). Hull form optimization of a cargo ship for reduced drag. *Journal of Hydrodynamics Series B*, 28(2), 173–183. doi:10.1016/S1001-6058(16)60619-4
- Journée, J. M. J. (1992). *Experiments and calculations on 4 wigley hull forms in head waves*. Holand: Delft University of Technology.
- Kamiński, B. (2015). A method for the updating of stochastic kriging metamodels. *European Journal of Operational Research*, 247(3), 859–866. doi:10.1016/j.ejor.2015.06.070
- Kennedy, J., & Eberhart, R. C. (1995). *Particle swarm optimization*. Proceedings of the 1995 IEEE international conference on neural networks, Australia.
- Kim, H., & Yang, C. (2010). A new surface modification approach for CFD-based hull form optimization. *Journal of Hydrodynamics, Series B*, 22(5), 520–525. doi:10.1016/S1001-6058(09)60246-8
- Kim, K. J. (1995). *Ship flow calculations and resistance minimization*. Swedish: Chalmers University of Technology.
- Li, N. J., Wang, W. J., Hsu, J., & Chen, C. (2014). Enhanced particle swarm optimizer incorporating a weighted particle. *Neurocomputing*, 124(2014), 218–227. doi:10.1016/j.neucom.2013.07.005
- Li, S. Z. (2012). *Research on hull form design optimization based on SBD technique*. Wuhan: China Ship Research and Development Academy. (In Chinese).
- Li, Z., & Liang, X. R. (2007). Control for freeway speed limitation based on elman neural network. *Microcomputer Information*, 23(28), 27–29. (In Chinese).
- Liu, H., Tian, H. Q., Liang, X. F., & Li, F. Y. (2015). Wind speed forecasting approach using secondary decomposition algorithm and elman neural networks. *Applied Energy*, 157, 183–194. doi:10.1016/j.apenergy.2015.08.014
- Liu, Z., & Luo, Z. (2005). Alertness states classification by SOM and LVQ neural networks. *Proceedings of World Academy of Science Engineering & Technique*, 1(4), 228–231. Retrieved

- from https://hal.inria.fr/inria-00000662/file/ljJT2004_khaled.pdf
- Longo, J., & Stern, F. (2005). Uncertainty assessment for towing tank tests with example for surface combatant DTMB model 5415. *Journal of Ship Research*, 49(1), 55–68. Retrieved from http://www.ivt.ntnu.no/imt/courses/tmr7/resources/Stern_uncertainty_anal_JSR.pdf
- Morris, M. D., & Mitchell, T. J. (1995). Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43, 381–402. doi:10.1016/0378-3758(94)00035-T
- Nik, A. A., Nejad, F. M., & Zakeri, H. (2016). Hybrid PSO and GA approach for optimizing surveyed asphalt pavement inspection units in massive network. *Automation in Construction*, 71, 325–345. doi:10.1016/j.autcon.2016.08.004
- Ortigosa, I., López, R., & García, J. (2009). Prediction of total resistance coefficients using neural networks. *Journal of Maritime Research*, 6, 15–26. Retrieved from <https://www.researchgate.net/publication/228658884>
- Patankar, S. V., & Spalding, D. B. (1972). A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer*, 15(10), 1787–1806. doi:10.1016/0017-9310(72)90054-3
- Puig, V., Witzczak, M., Nejjari, F., Quevedo, J., & Korbicz, J. (2007). A GMDH neural network-based approach to passive robust fault detection using a constraint satisfaction backward test. *Engineering Applications of Artificial Intelligence*, 20(7), 886–897. doi:10.1016/j.engappai.2006.12.005
- Qian, J. K., Mao, X. F., Wang, X. Y., & Yun, Q. Q. (2012). Ship hull automated optimization of minimum resistance via CFD and RSM technique. *Journal of Ship Mechanics*, 16(1), 36–43. (In Chinese).
- Shen, C., Song, R., Li, J., Zhang, X. M., Tang, J., Shi, Y. B., ... Cao, H. L. (2015). Temperature drift modeling of MEMS gyroscope based on genetic-elman neural network. *Mechanical Systems and Signal Processing*, 72, 897–905. doi:10.1016/j.ymssp.2015.11.004
- Song, Z. Z., & Zhao, Y. Q. (2016). Suspension test system based on modified elman neural network. *China Mechanical Engineering*, 27(1), 1–6. (In Chinese).
- Sun, Z. X., Song, J. J., & An, Y. R. (2010). Optimization of the head shape of the CRH3 high speed train. *Science China Technological Sciences*, 53(12), 3356–3364. doi:10.1007/s11431-010-4163-5
- Tahara, Y., Peri, D., Campana, E. F., & Stern, F. (2008). Computational fluid dynamics-based multi-objective optimization of a surface combatant using a global optimization method. *Journal of Marine Science and Technology*, 13(2), 95–116. doi:10.1007/s00773-007-0264-7
- Taormina, R., & Chau, K. W. (2015). Data-driven input variable selection for rainfall-runoff modeling using binary-coded particle swarm optimization and extreme learning machines. *Journal of Hydrology*, 529(3), 1617–1632. doi:10.1016/j.jhydrol.2015.08.022
- Tungadio, D. H., Jordaan, J. A., & Siti, M. W. (2016). Power system state estimation solution using modified models of PSO algorithm: Comparative study. *Measurement*, 92, 508–523. doi:10.1016/j.measurement.2016.06.052
- Wang, D. F., & Lu, F. (2015). Body-in-white lightweight based on multidisciplinary design optimization. *Journal of Jilin University (Engineering and Technology Edition)*, 45(1), 29–37. (In Chinese).
- Wang, W., Mo, R., & Zhang, Y. (2011). Multi-objective aerodynamic optimization design method of compressor rotor based on isight. *Procedia Engineering*, 15, 3699–3703. doi:10.1016/j.proeng.2011.08.693
- Wang, W. C., Kwokwing, C., Xu, D. M., & Chen, X. Y. (2015). Improving forecasting accuracy of annual runoff time series using ARIMA based on EEMD decomposition. *Water Resources Management*, 29(8), 2655–2675. doi:10.1007/s11269-015-0962-6
- Wang, X., Ma, L., Wang, B., & Wang, T. (2011). Application of evolutionary elman neural network in real-time data forecasting. *Electric Power Automation Equipment*, 31(12), 77–81. (In Chinese).
- Wu, C. L., Chau, K. W., & Li, Y. S. (2009). Methods to improve neural network performance in daily flows prediction. *Journal of Hydrology*, 372(1-4), 80–93. doi:10.1016/j.jhydrol.2009.03.038
- Yakhot, V., & Orszag, S. A. (1986). Renormalization group analysis of turbulence. I. Basic theory. *Journal of Scientific Computing*, 1(1), 3–51. doi:10.1007/BF01061452
- Zhang, B. J., & Zhang, Z. X. (2015). Research on theoretical optimization and experimental verification of minimum resistance hull form based on rankine source method. *International Journal of Naval Architecture and Ocean Engineering*, 7(5), 785–794. doi:10.1515/ijnaoe-2015-0055
- Zhang, H. L., Hao, Z. D., Kong, X. M., & Li, W. (2013). Estimation of calorific value of coals using BP and elman neural networks. *Advanced Materials Research*, 781-784, 39–44. doi:10.4028/www.scientific.net/AMR.781-784.39
- Zhang, J., & Chau, K. W. (2009). Multilayer ensemble pruning via novel multi-sub-swarm particle swarm optimization. *Journal of Universal Computer Science*, 15(4), 840–858. Retrieved from <https://www.researchgate.net/publication/220349014>
- Zhang, Y. H. (2016). Application of BP neural network in structural damage diagnosis of bridge behind abutment. *Applied Mechanics and Materials*, 847, 440–444. doi:10.4028/www.scientific.net/AMM.847.440
- Zhao, F. M., Gao, C. J., & Xia, Q. (2011). Overlap grid research on the application of ship CFD. *Journal of Ship Mechanics*, 15(4), 332–341. (In Chinese).
- Zhou, C., Ding, L. Y., & He, R. (2013). PSO-based Elman neural network model for predictive control of air chamber pressure in slurry shield tunneling under Yangtze river. *Automation in Construction*, 36(15), 208–217. doi:10.1016/j.autcon.2013.03.001
- Zhou, J. P., Yang, P., & Yang, N. (2011). Comparison and application of BP and elman network in fault diagnosis of condenser. *East China Electric Power*, 39(5), 822–824. (In Chinese).