# Model predictive ship collision avoidance based on Q-learning beetle swarm antenna search and neural networks

Xie, Shuo; Garofano, Vittorio; Chu, Xiumin; Negenborn, Rudy R.

**DOI**
10.1016/j.oceaneng.2019.106609

**Publication date**
2019

**Document Version**
Accepted author manuscript

**Published in**
Ocean Engineering

**Citation (APA)**
Xie, S., Garofano, V., Chu, X., & Negenborn, R. R. (2019). Model predictive ship collision avoidance based on Q-learning beetle swarm antenna search and neural networks. *Ocean Engineering*, *193*, Article 106609. https://doi.org/10.1016/j.oceaneng.2019.106609

# Model predictive ship collision avoidance based on Q-learning beetle swarm antenna search and neural networks

Shuo Xie[a,b], Vittorio Garofano[b], Xiumin Chu[a], Rudy R. Negenborn[b,a]

[a]*Intelligent Transportation Systems Research Center, Wuhan University of Technology, 430063, Wuhan, Hubei Province, P.R. China.*
[b]*Department of Maritime and Transport Technology, Delft University of Technology, 2628 CD Delft, The Netherlands*

**Abstract**

Real-time collision avoidance with full consideration of ship maneuverability, collision risks and International Regulations for Preventing Collisions at Sea (COLREGs) is difficult in multi-ship encounters. To deal with this problem, a novel method is proposed based on model predictive control (MPC), an improved Q-learning beetle swarm antenna search (I-Q-BSAS) algorithm and neural networks. The main idea of this method is to use a neural network to approximate an inverse model based on decisions made with MPC for collision avoidance. Firstly, the predictive collision avoidance strategy is established following the MPC concept incorporating an I-Q-BSAS algorithm to solve the optimization problem. Meanwhile, the relative collision motion states in typical encounters are collected for training an inverse neural network model, which is used as an approximated optimal policy of MPC. Moreover, to deal with uncertain dynamics, the obtained policy is reinforced by long-term retraining based on an aggregation of on-policy and off-policy data. Ship collision avoidance in multi-ship encounters can be achieved by weighting the outputs of the neural network model with respect to different target ships. Simulation experiments under several typical and multi-ship encounters are carried out using the KVLCC2 ship model to verify the effectiveness of the proposed method.

*Keywords:* collision avoidance, multi-ship encounters, predictive control, beetle swarm antennas search, neural networks

## 1. Introduction

Ship collision avoidance is an important research topic for navigation safety. Research methods on collision avoidance have been developed from traditional path planning methods (i.e., A* and artificial potential field) to intelligent optimization methods. In this section, a survey of the existing ship collision avoidance methods and the related works of the algorithms adopted in this study (i.e., model predictive control and beetle swarm antenna search) are introduced, the contributions of this paper are also expounded.

### *1.1. A survey of ship collision avoidance methods*

Existing ship collision avoidance methods can be divided into two main categories: path generation methods and intelligent optimization methods.

Traditional path generation methods, such as the A* algorithm (Chen et al., 2016) and artificial potential field (Xue et al., 2012), have been applied in collision avoidance for decades and showed good results. A* is a global heuristic search algorithm with both considerations of the start position and the destination. The main drawback of A* is the relatively low search efficiency in a large grid map. Hierarchical Planning (Wang et al., 2014; Cheng et al., 2014) is an effective approach to improve the efficiency by pre-processing a higher-level map before path planning. Different form the grid map in A*, APF uses artificial gravitational and repulsive fields to model the environment with small computation (Kim et al., 2011). The generated paths of APF are smoother than those of A*, which are more suitable for ships. Related works of APF (Lyu and Yin, 2018; Lazarowska, 2018) for ship path planning have been carried out in past years.

In recent years, with the introduction of COLREGs and the

*Email address:* xieshuo@whut.edu.cn (Shuo Xie)

development of collision risk model theories, intelligent optimization methods have been studied to achieve real-time and reliable ship collision avoidance (Li et al., 2019), such as neural networks (Simsir et al., 2014), fuzzy mathematics (Perera et al., 2012), swarm itelligence (Lazarowska, 2015) and reinforcement learning (Yin and Zhang, 2018).

Neural network is commonly used to model the uncertain factors in calculation of collision risks (Inaishi et al., 1992). Combinations with expert systems (Simsir et al., 2014) and fuzzy mathematics (Ahn et al., 2012) are the common approaches to overcome the defects of neural networks, i.e., local extremum and low precision with insufficient samples. Besides, fuzzy mathematics can be also applied in the fuzzy classification of collision risk (Hara and Hammer, 1993) and fuzzy reasoning (Perera et al., 2015). The performance of fuzzy mathematics mainly depends on the membership functions set in advance, which needs more prior knowledge.

With the development of computer science, swarm intelligence and model-free reinforcement learning methods attract more attention in ship collision avoidance. The common used swarm intelligence algorithms in ship area are ant colony optimization (Lazarowska, 2015; Tsou and Hsueh, 2010) and particle swarm optimization (Ma et al., 2018; Chen and Huang, 2012a), which can obtain good results with an appropriate fitness function (Liu et al., 2017). Reinforcement learning is a classical machine learning method, which has been widely used in artificial intelligence field. With the development of deep learning, deep reinforcement learning has been proposed to solve the continuous state decision problem through end-to-end learning (Mnih et al., 2015), which makes the application of reinforcement learning methods in ship collision avoidance become possible (Yin and Zhang, 2018). However, the relative low learning efficiency of RL has become the biggest obstacle in its practical application in ship collision avoidance.

In summary, traditional path generation methods and intelligent optimization methods have both been successfully studied for ship collision avoidance. SI algorithms are commonly used with ship collision risk model to achieve the collision avoidance. Model-free methods can obtain a general optimal policy through interactions with the environment, but have the disadvantage of low learning efficiency.

### *1.2. Related works on model predictive control for ship collision avoidance*

Ship motion has large inertia and hysteresis characteristics, which brings challenges for collision avoidance. Approaches in the control area for systems with time delay (Zhang and Zhang, 2017; Zhang et al., 2016; Ou et al., 2009; Zhang et al., 2004; Zheng et al., 2016, 2017a,b) provide solutions for ship collision avoidance. Among them, a typical model-based control method, i.e., model predictive control (MPC) (Zheng et al., 2017b, 2016, 2017a), has attracted much attention in ship collision avoidance due to the ability of fully considering the ship maneuverability model and the constraints. MPC is an effective approach with advantages of rolling optimization and state prediction, which can be easily combined with other algorithms.

Distributed MPC (DMPC) has been applied in the field of multi-agent collaborative collision avoidance to solve the problem of motion conflict between multiple agents (Negenborn and Maestre, 2014; Zheng et al., 2017b). An effective approach to realize the collision avoidance is to treat the collision risk index between each agent (e.g., relative distance) as constraint conditions in MPC. Zheng et al. (2017b) propose a novel cost effective robust DMPC for waterborne automated guided vehicles, which can model the price of robustness by explicitly considering uncertainty and system characteristics in a tube-based robust control framework. Li et al. (2017) focus on the collision avoidance problem in unmanned aerial vehicles (UAVs) formation and proposes a real-time cooperative path planning scheme based on DMPC. Dai et al. (2017) also use DMPC to plan the motion of multiple UAVs. Meanwhile, virtual state trajectories with compatibility constraints are used to guide the system instead of the real trajectory, which can guarantee the collision avoidance stability of the whole system. Perizzato et al. (2015) use DMPC to solve the collision avoidance problem of multiple autonomous ground robots, and realizes the autonomous path

planning of multiple robots with obstacles and internal collision avoidance constraints. In addition, uncertain disturbances also have great impact on multi-agents collision avoidance. Chen et al. (2018b) use DMPC to solve the vessel train formation control problem of cooperative multi-vessel systems and maintains the distance between multiple ships. Besides, the ship domain model is also used to set the constraint conditions in MPC. Abdelaal et al. (2016) propose a trajectory tracking and collision avoidance method based on nonlinear MPC in elliptical ship fields, and apply it to a 3-DOF ship model. Furthermore, a disturbance observer is introduced in (Abdelaal et al., 2018) in the designed controller, and effective collision avoidance can be realized under the condition of uncertain disturbance.

Generally, to achieve reliable collision avoidance, all states of the so-called own ship and target ships still need to be predicted for optimization, which has a certain influence on the real-time performance of MPC. In order to increase the solving speed of MPC, one approach is to simplify the multi-step prediction in MPC to a one-step final cost calculation by combination with a reinforcement learning method. In (Negenborn et al., 2005), a value-function MPC approach is proposed to reduce the computation by using the output of the designed value-function instead of the cost functions of the original MPC, which can be regarded as a model-based reinforcement learning (MBRL) method with a deterministic model and known objective function. Then, the value-function MPC approach can provide supervised experiences for model-free RL algorithms, i.e., TD learning. This idea can be seen as a prototype of a model-based method for model-free (MB-MF) learning.

### *1.3. Related works on beetle swarm antenna search*

Stochastic optimization algorithms, such as ant colony optimization (ACO) (Bououden et al., 2015), particle swarm optimization (PSO) (Chen et al., 2018a; Wang et al., 2018b), etc., are commonly used for optimization tasks, e.g., the optimization in MPC. Among them, the PSO algorithm has been most widely used to solve various optimization problems because of its simple structure and fast optimization speed.

As a bionic search algorithm similar to PSO, beetle antennas search (BAS) algorithm (Jiang and Li, 2018; Zhu et al., 2018) and beetle antennas swarm search (BSAS) (Wang and Chen, 2018; Chen et al., 2018c) have been proposed recently, which have more concise search rules. The optimality of BSAS has been validated in several optimization problems (Wang et al., 2018a). For general optimization issues, e.g., the optimization of model parameters (Chen et al., 2018c), the path planning problem (Mu et al., 2019) etc., the BSAS algorithm has proofed to be an effective optimization approach. Regarding the fine-tuning problem, the BSAS is also capable of adjusting the hyper-parameters, e.g., the PID parameters (Lin et al., 2018) and the neural network parameters (Sun et al., 2019). Due to the concise search strategy, the BSAS algorithm is considered to have great potential in solving optimization problems.

### *1.4. Contributions*

In this study, real-time collision avoidance considering ship maneuverability, collision risk and COLREGs is realized for multi-ship encounters. The main contributions of this paper are: 1) An improved Q-BSAS is proposed to realize ship collision avoidance with model predictive control; 2) A neural network-based inverse model is used for the optimal policy approximation to reduce the time cost and reinforced by long-term retraining for robust collision avoidance.

We compare existing methods used for ship collision avoidance from the aspects of on-line ability, consideration of the ship maneuverability, consideration of collision risk model, ability to deal with uncertain dynamics, optimization or learning time cost and consideration of COLREGs, as shown in Table 1, and detailed as follows:

(1) On-line ability: The proposed method solves on-line collision avoidance problem based on MPC and neural networks. Benefiting from rolling optimization of MPC, the proposed MPC method and approximated neural networks have better on-line ability to deal with dynamic ships compared with existing off-line path planning methods (e.g.., A* (Ma et al., 2014)).

(2) Consideration of the ship maneuverability and collision

risk model: Compared with existing optimization methods (e.g., neutral networks (Simsir et al., 2014), fuzzy mathematics (Perera et al., 2012) and swarm intelligence (Liu et al., 2017)), the proposed method considers ship maneuverability and collision risk model comprehensively and reduces the feedback delay by collision risk prediction.

(3) Ability of dealing with uncertain dynamics: In the proposed neural network-based method, the inverse model is retrained by using on-policy data on the long-term, which takes the advantage of reinforcement learning method (Yin and Zhang, 2018) and has the ability to deal with uncertain ship dynamics.

(4) Time cost: The proposed neural network-based method reduces the time cost of optimization by using an approximated inverse model as the optimal policy, which has the advantage of function approximation in neutral networks (Simsir et al., 2014) and reinforcement learning (Yin and Zhang, 2018). Besides, compared with the pure model-free method (Yin and Zhang, 2018), the optimal policy is initialized by MPC, which makes use of more prior model information and requires less time for learning.

(5) Consideration of COLREGs: COLREGs are used to set the control constraints in the proposed method, which can generate more reliable collision avoidance results (Abdelaal et al., 2018) than the methods without considering COLREGs.

In addition, compared with existing BAS-based optimization algorithms (Jiang and Li, 2018; Wang and Chen, 2018), the proposed I-Q-BSAS algorithm achieves better optimization performance by the exploitation of the historical optimums and Q-learning-based behavior decision.

### *1.5. Outlines*

The remainder of this article is organized as follows. In Section 2, Preliminaries including the ship hydrodynamic model, collision risk model and COLREGs are described. In Section 3, the predictive collision avoidance strategy based on MPC is proposed. In Section 4, the reinforced inverse method for predictive collision avoidance in multi-ship encounters is proposed. In Section 5, simulation experiments under multi-ship encoun-

Table 1: Comparison of different methods and the proposed method.

| Ship collision avoidance methods | | On-line ability | Consideration of maneuverability | Collision risk model | Ability to deal with uncertain dynamics | Small time cost of optimization or learning | Consideration of COLREGs |
|---|---|---|---|---|---|---|---|
| A* | (Ma et al., 2014; Chen et al., 2016) | | | | | ✓ | |
| Artificial patential field | (Naeem et al., 2016) | ✓ | | ✓ | | ✓ | ✓ |
| | (Xue et al., 2011) | ✓ | | ✓ | | ✓ | ✓ |
| | (Xue et al., 2012) | ✓ | | | | ✓ | |
| Neutral networks | (Simsir et al., 2014) | ✓ | | | | ✓ | |
| Fuzzy mathematics | (Perera et al., 2012, 2011) | ✓ | ✓ | | | ✓ | ✓ |
| Ant colony optimization | (Lazarowska, 2015) | ✓ | ✓ | | | ✓ | ✓ |
| Swarm intelligence | (Ma et al., 2018) | | | ✓ | | ✓ | |
| Particle swarm optimization | (Liu et al., 2017) | ✓ | | ✓ | ✓ | ✓ | ✓ |
| | (Chen and Huang, 2012b) | ✓ | ✓ | ✓ | ✓ | ✓ | |
| Model predictive control | (Abdelaal et al., 2018) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| | (Zheng et al., 2017b) | ✓ | ✓ | ✓ | | ✓ | |
| | (Abdelaal et al., 2016) | ✓ | ✓ | ✓ | | ✓ | |
| | (Abdelaal et al., 2018) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |
| Reinforcement learning | (Yin and Zhang, 2018) | ✓ | | ✓ | ✓ | | |
| The proposed method | | ✓ | ✓ | ✓ | ✓ | ✓ (smaller trained inverse model) | ✓ |

ters are carried out to assess the effectiveness of the proposed methods. In Section 6, conclusions and further research are presented.

## 2. Preliminaries

In this section, the ship hydrodynamic model, collision risk model and the COLREGs are introduced.

### 2.1. Ship hydrodynamic model

Generally, the 3-DOF motion coordinates of an underactuated surface vessel are shown in Fig. 1.

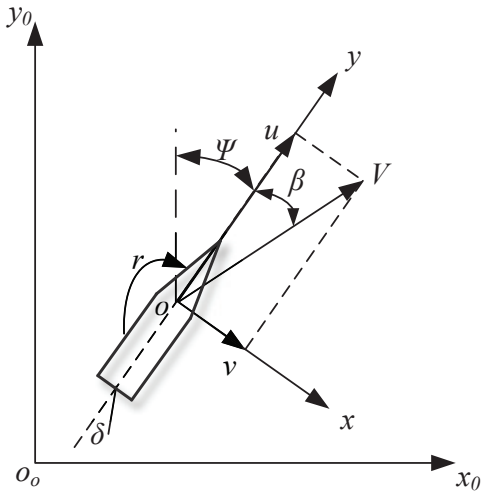


Figure 1: 3-DOF motion coordinate system of underactuated surface ship.

In Fig. 1, $O_o - x_o y_o$ is the inertial coordinate system of the vessel; $O - xy$ is the co-rotational coordinate system of the vessel; $u$, $v$ and $r$ are the velocities in surge (body-fixed $x$), sway (body-fixed $y$) and yaw directions, respectively; $\delta$ and $\psi$ are the rudder and heading angle of the vessel, respectively; $\beta$ is the drift angle. The kinematic model expressing the relationship between $[x, y, \psi]$ and $[u, v, r]$ is:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \sin(\psi) & \cos(\psi) & 0 \\ -\cos(\psi) & \sin(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix}, \tag{1}$$

Furthermore, a 3-DOF dynamics model of the vessel can be denoted as (Yasukawa and Yoshimura, 2015; Luo and Li, 2017):

$$\begin{aligned} &(m - X_{\dot{u}})\dot{u} = f_1(u, v, r, \delta), \\ &(m - Y_{\dot{v}})\dot{v} + (mx_G - Y_{\dot{r}})\dot{r} = f_2(u, v, r, \delta), \\ &(mx_G - N_{\dot{v}})\dot{v} + (I_z - N_{\dot{r}})\dot{r} = f_3(u, v, r, \delta), \end{aligned} \tag{2}$$

where $m$ is the total mass of the vessel; $x_G$ is the longitudinal coordinate of the gravity center of the vessel in surge direction; $I_Z$ is the moment of the inertia; $X_{\dot{u}}$, $Y_{\dot{v}}$, $Y_{\dot{r}}$, $N_{\dot{v}}$ and $N_{\dot{r}}$ are the inertia coefficients; $f_1$, $f_2$ and $f_3$ are the lumped forces and moments in 3-DOF, defined as:

$$\begin{aligned} f_1 =& X_u u + X_{uu}u^2 + X_{uuu}u^3 + X_{vv}v^2 + X_{rr}r^2 + X_{\delta\delta}\delta^2 \\ &+ X_{\delta\delta u}\delta^2 u + X_{vr}vr + X_{v\delta}v\delta + X_{v\delta u}v\delta u \\ &+ X_{uvv}uv^2 + X_{urr}ur^2 + X_{uvr}uvr + X_{r\delta}r\delta \\ &+ X_{ur\delta}ur\delta + X_0, \\ f_2 =& Y_{0u}u + Y_{0uu}u^2 + Y_r r + Y_\delta\delta + Y_{vvv}v^3 + Y_{\delta\delta\delta}\delta^3 \\ &+ Y_{vvr}v^2 r + Y_{vv\delta}v^2\delta + Y_{v\delta\delta}v\delta^2 + Y_{\delta u}\delta u + Y_{vu}vu \\ &+ Y_{ru}ru + Y_{\delta uu}\delta u^2 + Y_{rrr}r^3 + Y_{vrr}vr^2 + Y_{vuu}vu^2 \\ &+ Y_{ruu}ru^2 + Y_{r\delta\delta}r\delta^2 + Y_{rr\delta}r^2\delta + Y_{rv\delta}rv\delta + Y_0, \\ f_3 =& N_{0u}u + N_{0uu}u^2 + N_r r + N_\delta\delta + N_{vvv}v^3 + N_{\delta\delta\delta}\delta^3 \\ &+ N_{vvr}v^2 r + N_{vv\delta}v^2\delta + N_{v\delta\delta}v\delta^2 + N_{\delta u}\delta u + N_{vu}vu \\ &+ N_{ru}ru + N_{\delta uu}\delta u^2 + N_{rrr}r^3 + N_{vrr}vr^2 + N_{vuu}vu^2 \\ &+ N_{ruu}ru^2 + N_{r\delta\delta}r\delta^2 + N_{rr\delta}r^2\delta + N_{rv\delta}rv\delta + N_0, \end{aligned} \tag{3}$$

where $X_*$, $Y_*$ and $N_*$ are the hydrodynamic coefficients with respect to the motion state $*$, e.g., $X_u$ is the hydrodynamic coefficient with respect to $u$ in the surge direction. Without lose of generality, (1)~(3) can be denoted by the following non-linear state-space model:

$$\begin{aligned} &\dot{\boldsymbol{X}} = \boldsymbol{f_A}(\boldsymbol{X}, \delta), \\ &\boldsymbol{X} = \begin{bmatrix} x & y & \psi & u & v & r \end{bmatrix}^{\mathrm{T}}. \end{aligned} \tag{4}$$

where $\boldsymbol{f_A}$ represents the state transition function between the lumped state $\boldsymbol{X}$, the control rudder $\delta$ and the differential $\dot{\boldsymbol{X}}$.

### 2.2. Collision risk model

In typical encounters, the collision risk index (CRI) with respect to two ships can be evaluated using several relative motion parameters (Szlapczynski and Szlapczynska, 2017), i.e., distance of the closest point of approach (DCPA), time to the closest point of approach (TCPA), the relative distance and position direction and the relative speed, as shown in Fig. 2

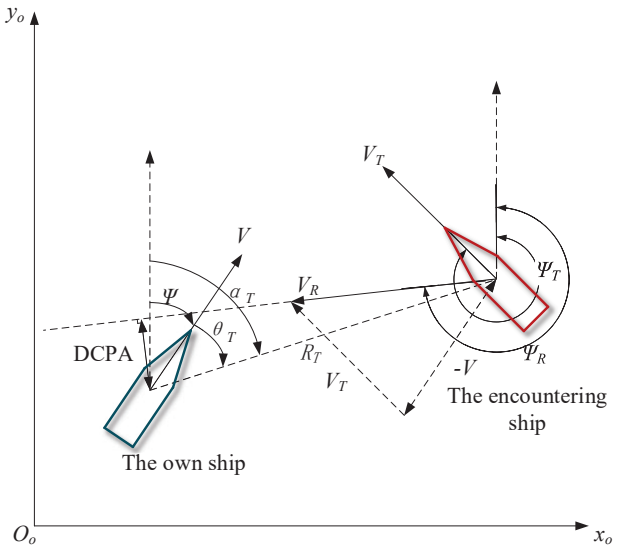


Figure 2: Motion parameters of two ships in typical encounter.

For convenience of expression, the ship we consider for collision avoidance is defined as the own ship and the obstacle ship which causes the encounter situation with the own ship is defined as the target ship. Conventionally, the relative motion parameters are calculated by:

$$\begin{aligned} DCPA &= R_T \sin\left(\psi_R - \alpha_T - \pi\right), \\ TCPA &= R_T \cos\left(\psi_R - \alpha_T - \pi\right) / V_R, \\ R_T &= \sqrt{(x_T - x)^2 + (y_T - y)^2}, \\ \theta_T &= \alpha_T - \psi \pm 2\pi, \end{aligned} \tag{5}$$

where $R_T$ is the relative distance between two ships; $\psi_R$ is the relative course direction of the target ship; $\alpha_T$ is the true relative position direction of the target ship; $\theta_T$ is the angle converted from $\alpha_T$ in the body-fixed coordinate system of the own ship to the inertial coordinate system; $V_R$ is the relative speed of the target ship. The intermediate parameters in (5) can be calculated by:

$$\begin{aligned} v_{x_R} =& u_T \sin(\psi_T) + v_T \cos(\psi_T) - (u \sin(\psi) + v \cos(\psi)), \\ v_{y_R} =& u_T \cos(\psi_T) - v_T \sin(\psi_T) - (u \cos(\psi) - v \sin(\psi)), \\ V_R =& \sqrt{{v_{x_R}}^2 + {v_{y_R}}^2}, \\ \psi_R =& \arctan \frac{v_{x_R}}{v_{y_R}}, \\ &+ \begin{cases} 0 & v_{x_R} \geq 0 \cap v_{y_R} \geq 0 \\ \pi & (v_{x_R} < 0 \cap v_{y_R} < 0) \cup (v_{x_R} \geq 0 \cap v_{y_R} < 0) \\ 2\pi & (v_{x_R} < 0 \cap v_{y_R} \geq 0) \end{cases}, \\ \alpha_T =& \arctan \frac{x_R}{y_R} \\ &+ \begin{cases} 0 & x_R \geq 0 \cap y_R \geq 0 \\ \pi & (x_R < 0 \cap y_R < 0) \cup (x_R \geq 0 \cap y_R < 0) \\ 2\pi & (x_R < 0 \cap y_R \geq 0) \end{cases}, \\ x_R =& x_T - x, \\ y_R =& y_T - y, \\ C_T =& \psi_T - \psi. \end{aligned} \tag{6}$$

where $\boldsymbol{X}_T = [x_T, y_T, \psi_T, u_T, v_T, r_T]^{\mathrm{T}}$ are the motion states of the target ship; $v_{x_R}$ and $v_{y_R}$ are the relative speed components of the target ship on the $X$ and $Y$ axes, respectively; $C_T$ is the relative heading angle of the target ship.

Since fuzzy logic is quite suitable in dealing with linguistic representations and subjectives concept like collision risk (Ahn et al., 2012), plenty of researches (Hara and Hammer, 1993; Ahn et al., 2012) have adopted fuzzy logic to model the degree of the collision risk by using membership functions. In fuzzy logic, the membership function is a generalization of an indicator in classical sets, which represents the degree of truth as an extension of valuation. In this study, the fuzzy logic method is also used for determining the index of the collision risk based on DCPA, TCPA, $R_T$, $\theta_T$ and velocity ratio $K = V_T/V$, of which the memberships are defined as follows:

1. The membership function of DCPA:

$$u_{DCPA} = \begin{cases} 1 & |DCPA| \leq d_1, \\ \frac{1}{2} - \frac{1}{2} \sin\left[\frac{\pi}{d_2 - d_1}\left(|DCPA| - \frac{d_1 + d_2}{2}\right)\right] & d_1 < |DCPA| \leq d_2, \\ 0 & |DCPA| > d_2, \end{cases} \tag{7}$$

where, $d_1$ is the closest safety distance of the two ships, which varies with $\theta_T$ as:

$$d_1 = \begin{cases} 1.1 - \frac{\theta_T}{\pi} \times 0.2 & 0° \le \theta_T < 112.5°, \\ 1.0 - \frac{1.5\pi - \theta_T}{\pi} \times 0.8 & 112.5° \le \theta_T < 247.5°, \\ 1.1 - \frac{2\pi - \theta_T}{\pi} \times 0.4 & 247.5° \le \theta_T < 360°, \end{cases} \tag{8}$$

2. The membership of $R_T$ :

The so-called distances of the last action (DLA) and Arena are used to determine the membership of $R_T$. The Arena is proposed in (Davis et al., 1982) to describe the area for which entering of a ship should trigger a collision avoidance action so as to avoid violating the actual domain (Szlapczynski and Szlapczynska, 2017). The DLA indicates the closest distance for taking action to avoid collision (Chen et al., 2015). Then the membership of $R_T$ is:

$$u_{R_T} = \begin{cases} 1 & R_T \le D_{\mathrm{D}}, \\ \frac{1}{2} - \frac{1}{2}\sin\left[\frac{\pi}{D_{\mathrm{A}} - D_{\mathrm{D}}}\left(R_T - \frac{D_{\mathrm{D}} + D_{\mathrm{A}}}{2}\right)\right] & D_{\mathrm{D}} < R_T \le D_{\mathrm{A}}, \\ 0 & R_T > D_{\mathrm{A}}, \end{cases} \tag{9}$$

where $D_{\mathrm{D}}$ and $D_{\mathrm{A}}$ are the value of DLA and radius of the Arena respectively (Chen et al., 2015), which are set as the distance limits for collision avoidance.

3. The membership of TCPA:

$$u_{TCPA} = \begin{cases} 1 & |TCPA| \le t_1, \\ \left(\frac{t_2 - |TCPA|}{t_2 - t_1}\right) & t_1 < |TCPA| \le t_2, \\ 0 & |TCPA| > t_2, \end{cases} \tag{10}$$

where $t_1$ and $t_2$ represent the time limits for collision avoidance, which can be determined by $D_{\mathrm{D}}$ and $D_{\mathrm{A}}$:

$$\begin{aligned} t_1 &= \begin{cases} \frac{1}{V_R}\sqrt{{D_{\mathrm{D}}}^2 - DCPA^2} & DCPA \le D_{\mathrm{D}}, \\ \frac{1}{V_R}(D_{\mathrm{D}} - DCPA) & DCPA > D_{\mathrm{D}}, \end{cases} \\ t_2 &= \begin{cases} \frac{1}{V_R}\sqrt{{D_{\mathrm{A}}}^2 - DCPA^2} & DCPA \le D_{\mathrm{A}}, \\ \frac{1}{V_R}(D_{\mathrm{A}} - DCPA) & DCPA > D_{\mathrm{A}}, \end{cases} \end{aligned} \tag{11}$$

4. The membership of $\theta_T$ :

$$u_{\theta_T} = \frac{1}{2}\left[\cos(\theta_T - 19) - \frac{5}{17} + \sqrt{\frac{440}{289} + \cos^2(\theta_T - 19)}\right]. \tag{12}$$

Generally, it is most dangerous when the target ship is coming from 19° of the own ship under the same other conditions (Yan, 2002).

5. The membership of $K$ :

$$u_K = \frac{1}{1 + \frac{2}{K\sqrt{K^2 + 1 + 2K\sin C}}}. \tag{13}$$

where $C \in [0, 180]$ is a constant coefficient.

Therefore, the following risk model is established (Chen et al., 2015):

$$\begin{aligned} f_{CRI} &= \boldsymbol{\lambda}_{CRI}\boldsymbol{u}_{CRI}{}^{\mathrm{T}}, \\ \boldsymbol{\lambda}_{CRI} &= \begin{bmatrix} \lambda_{DCPA} & \lambda_{TCPA} & \lambda_{R_T} & \lambda_{\theta_T} & \lambda_K \end{bmatrix}, \\ \boldsymbol{u}_{CRI} &= \begin{bmatrix} u_{DCPA} & u_{TCPA} & u_{R_T} & u_{\theta_T} & u_K \end{bmatrix}. \end{aligned} \tag{14}$$

where $f_{CRI}$ is the collision risk index used to evaluate the risk level, which is the outcome of the collision risk model. The $\lambda_{DCPA}$ , $\lambda_{TCPA}$ ,$\lambda_{R_T}$, $\lambda_{\theta_T}$ and $\lambda_K$ are the set weights of $u_{DCPA}$ , $u_{TCPA}$ ,$u_{R_T}$, $u_{\theta_T}$ and $u_K$, respectively.

### 2.3. COLREGs

The COLREGs are the maritime traffic rules formulated by the International Maritime Organization (IMO) to prevent and avoid collisions between ships at sea (Naeem et al., 2012). COLREGs divide the encounter state of two ships into three categories, i.e., head-on (rule 14), small and large angle crossing (rule 15) and over-taking (rule 13) according to different relative position directions ($\theta_T$ in Fig. 2) between two ships, which are shown in Fig. 3.

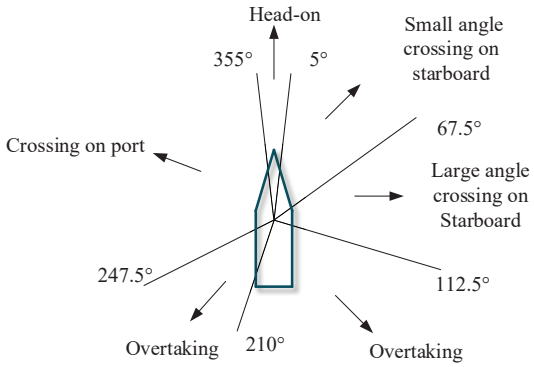


Figure 3: Encounter situations classification in typical encounter.

In Fig. 3, $\theta_T \in [355°, 360°] \cup [0°, 5°]$ represents the head-on encounter; $\theta_T \in [5°, 67.5°]$ represents the small angle crossing encounter on starboard side and $\theta_T \in [67.5°, 112.5°]$ represents the large angle crossing encounter on starboard side; $\theta_T \in [247.5°, 355°]$ represents the crossing encounter on larboard side; $\theta_T \in [112.5°, 247.5°]$ represents the over-taking encounter (Zheng and Wu, 2000).

For collision avoidance suggestions, the COLREGs stipulate that the own ship is the give-way ship, i.e., the ship which should change the course to avoid the collision, and has the duty of steering to avoid collision in head-on and starboard crossing encounter scenarios. Otherwise, the own ship is the stand-on ship, meaning that it should keep its course and speed. When the own ship is the give-way ship, it is generally recommended to turn right in head-on and starboard crossing encounters to avoid crossing ahead of the encountering ship. In large angle starboard crossing encounter, a port steering which has a better effect than a starboard steering is acceptable (Tsou and Hsueh, 2010; Tsou et al., 2010).

## 3. Predictive collision avoidance based on Improved Q-BSAS

With the ship hydrodynamic model and collision risk model, the future states and collision risks of the own ship and encountering ship can be predicted and the MPC scheme can be adopted for collision avoidance.

### 3.1. Optimization problem

For compact formulation, $f_{CRI}(\boldsymbol{X}, \boldsymbol{X}_T, \delta)$ is used to represent the nonlinear relationship between the collision risk $f_{CRI}$ and the state of two ships $(\boldsymbol{X}, \boldsymbol{X}_T)$ with the control value $\delta$ of the own ship based on (1)~(14). Then, the value of $f_{CRI}$ can be regarded as the negative safety reward of the action $\delta$ with the current state $\boldsymbol{X}, \boldsymbol{X}_T$, i.e., the safety is inversely proportional to the $f_{CRI}$. Meanwhile, the action change $\Delta\delta$ can be used as the negative economic reward, i.e., the economy is inversely proportional to the $\Delta\delta$.

Referring to the idea of reinforcement learning (RL) and MPC, the discounted sum of these two rewards at control time step $t$ can be used for optimization as:

$$\begin{aligned} F_{CRI}(t) &= \sum_{t'=t+1}^{t'=t+N_P} \gamma^{t'-t} \left| \hat{f}_{CRI}(t') \right|, \\ F_u(t) &= \sum_{t'=t+1}^{t'=t+N_P} \gamma^{t'-t} \left| \Delta\hat{\delta}(t') \right|, \\ \Delta\hat{\delta}(t') &= \begin{cases} \hat{\delta}(t') - \hat{\delta}(t'-1) & if(t' > t+1) \\ \hat{\delta}(t') - \delta(t) & if(t' = t+1) \end{cases}, \end{aligned} \tag{15}$$

where $[t+1, t+N_P]$ is the prediction horizon at control time step $t$, $t'$ is the prediction time step, $\gamma \in [0, 1]$ is a discount factor that can guarantee the convergence of the final value, $\Delta\hat{\delta}(t')$ is the action change at time $t'$ in the prediction horizon, $\hat{f}_{CRI}(t')$ is the calculated collision risk under the control of $\hat{\delta}(t')$.

The final target of MPC is to find the control sequence $\hat{\boldsymbol{\delta}}(t) = \left[\hat{\delta}(t+1), \hat{\delta}(t+2), \cdots, \hat{\delta}(t+N_P)\right]$ that maximizes the sum of positive rewards, i.e., to minimize the sum of negative rewards ($F_{CRI}$ and $F_u$) over prediction horizon. Therefore, the optimization strategy can be established as:

$$\begin{aligned} &\arg\min_{\delta} J(t) = \arg\min_{\delta} \left\{ \mu_1 F_{CRI}(t) + \mu_2 F_u(t) \right\}, \\ &subject\ to\ -\delta_{\max} \le \hat{\delta}(t') \le \delta_{\max}, \end{aligned} \tag{16}$$

where $\mu_1$ and $\mu_2$ are two positive weights, which satisfy $\mu_1 + \mu_2 = 1$, $\delta_{\max}$ is the maximum rudder angle of the own ship.

In addition, COLREGs described in Section 2.3 should be considered in the collision avoidance. The own ship is recommended to give starboard steering to avoid encountering ships in head-on, overtaking and small angle crossing encounters, otherwise, normal input constraints are set based on the maximum rudder angle in large angle crossing encounters. Then, the constraint in (16) is modified as:

$$\begin{cases} -\delta_{\max} \le \hat{\delta}(t') \le \delta_{\max} & if\ \ 67.5° < \theta_T \le 355°, \\ 0 \le \hat{\delta}(t') \le \delta_{\max} & otherwise, \end{cases} \tag{17}$$

and the optimization problem is denoted as:

$$\begin{aligned} &\arg\min_{\delta} J(t) = \arg\min_{\delta} \left\{ \mu_1 F_{CRI}(t) + \mu_2 F_u(t) \right\}, \\ &subject\ to\ (17). \end{aligned} \tag{18}$$

### 3.2. Re-sailing and state constraints

Since the collision states (e.g., the DCPA, TCPA, etc.) will change during the voyage, the ship is considered to re-sail back to the route if there is no collision risk. Moreover, the give-way ship may change to a stand-on ship after collision avoidance measures are taken, which requires a stable course angle and speed based on COLREGs. In order to solve the course keeping and re-sailing problem at the same time, the widely used line-of-sight (LOS) guidance strategy (Liu et al., 2018) is adopted in this study to transform the re-sailing problem to a course keeping problem. When two ships have passed each other (i.e., $TCPA < 0$), the LOS guidance strategy is adopted for re-sailing and course keeping, otherwise the proposed collision avoidance method is adopted for collision avoidance. Fig. 4 shows the LOS guidance strategy when the own ship is sailing from the start position $P_s(x_s, y_s)$ to the destination $P_d(x_d, y_d)$.

In LOS guidance, the ship is guided by minimizing the error $\tilde{\psi}_{\text{LOS}}$ between the actual heading angle $\psi$ and the LOS angle $\psi_{\text{LOS}}$. The LOS angle $\psi_{\text{LOS}}$ can be calculated by solving the following equations:

$$\begin{aligned} &(x_{\text{LOS}} - x)^2 + (y_{\text{LOS}} - y)^2 = R_{\text{LOS}}^2, \\ &\tfrac{y_{\text{LOS}} - y_s}{x_{\text{LOS}} - x_s} = \tfrac{y_d - y_s}{x_d - x_s}, \\ &\psi_{\text{LOS}} = \arcsin\left(\tfrac{x_{\text{LOS}} - x}{R_{\text{LOS}}}\right), \end{aligned} \tag{19}$$

where $P_{\text{LOS}}\,(x_{\text{LOS}}, y_{\text{LOS}})$ is the LOS guidance point; $R_{\text{LOS}}$ is the radius of the acceptance circle in Fig. 4. Then the tracking error is $\tilde{\psi}_{\text{LOS}} = (\psi - \psi_{\text{LOS}})$.

Note that the own ship may consider abnormal behavior to obtain the smallest cost, it is necessary to set reasonable state constraints. Since the LOS guidance strategy is used for stand-on and re-sailing, we set the constraint on the LOS tracking error as $\left|\tilde{\psi}_{\text{LOS}}\right| < \frac{\pi}{2}$ during the collision avoidance to prevent go backward behavior caused by excessive steering. Besides, the constraint on the relative distance between the own ship and encountering ship is also considered as $R_T > L_{PP}$ where $L_{PP}$ is the ship length to prevent dangerous stand-on behavior. Therefore,

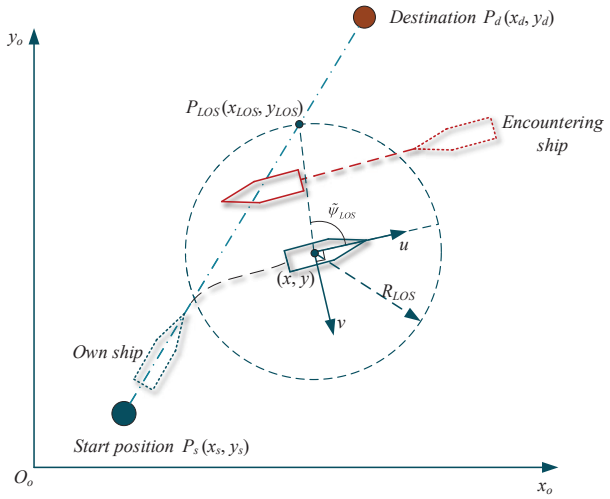


Figure 4: The LOS guidance strategy and LOS tracking error.

the optimization problem in (18) is modified as:

$$\begin{aligned} &\arg\min_{\delta} J(t) = \arg\min_{\delta} \{\mu_1 F_{CRI}(t) + \mu_2 F_u(t)\}, \\ &subject\ to\ (17), \left|\tilde{\psi}_{\text{LOS}}\right| < \tfrac{\pi}{2}, R_T > L_{PP}. \end{aligned} \tag{20}$$

In this study, the widely used penalty function and saturation approaches are adopted in the predictive collision avoidance to deal with the inequality state constraints and control input constraints in (20), respectively.

### 3.3. Improved Q-BSAS algorithm for collision avoidance

Generally, conventional optimization methods for complex nonlinear optimization tasks like (20) are very sensitive to the initialization of the optimization and usually lead to unacceptable solutions due to the local optima (Song et al., 2007). To deal with this problem, stochastic optimization algorithms, e.g., particle swarm optimization (Chen et al., 2018a; Wang et al., 2018b), have been used in MPC and been able to achieve global optimality by exploitation of knowledges from previous iterations (Bououden et al., 2015). As a stochastic search algorithm, beetle swarm antennas search (BSAS) algorithm has more concise search rules and less computation than particle swarm optimization (Wang and Chen, 2018), which is considered for MPC optimization problem in this study.

In addition, existing BSAS algorithms do not make full use of the historical trajectories of the antennas and the step of the

beetle is adjusted only based on the current performance in each iteration (Wang and Chen, 2018), which leads to the difficulty of finding a trade-off between the exploratory and exploitation in complex problem (e.g., the nonlinear optimization problem in (20)).

In order to solve the optimization problem in (20), an improved Q-BSAS (I-Q-BSAS) algorithm is proposed by introducing the historical optimums of the antennas and a behavior decision based on Q-learning. At each control step $t$ of MPC, the I-Q-BSAS algorithm takes the control sequence $\hat{\boldsymbol{\delta}}(t)$ as the variables to be optimized and the objective function $J$ of MPC as the fitness function. Then the optimization algorithm searches $\hat{\boldsymbol{\delta}}(t)$ within the set number of iterations and takes the optimal $\hat{\boldsymbol{\delta}}(t)$ corresponding to the best fitness $J$ as the optimization result of MPC.

### 3.3.1. Orignal BSAS based collision avoidance

Throughout the paper, the number of iterations in the optimization process is expressed as $k$ to distinguish from the control step $t$ in MPC. All the following vectors with respect to the left, right antennas and the centroid of the beetle represent control sequences $\hat{\boldsymbol{\delta}}$ for searching. The random value used in optimization is expressed as $r_d$.

The Original BSAS algorithm sets a certain number of beetles for optimization. Each beetle searches the optimal positions following the BAS algorithm. The update strategy of each beetle is:

$$\begin{aligned}
\hat{\boldsymbol{\delta}}_{il_k} &= \hat{\boldsymbol{\delta}}_{i_{k-1}} + \tfrac{1}{2} d^k \tfrac{\boldsymbol{d}}{\|\boldsymbol{d}\|}, \\
\hat{\boldsymbol{\delta}}_{ir_k} &= \hat{\boldsymbol{\delta}}_{i_{k-1}} - \tfrac{1}{2} d^k \tfrac{\boldsymbol{d}}{\|\boldsymbol{d}\|}, \\
\hat{\boldsymbol{\delta}}_{i_k} &= \hat{\boldsymbol{\delta}}_{i_{k-1}} + c d^k sign\left(J_{il_k} - J_{ir_k}\right) \tfrac{\boldsymbol{d}}{\|\boldsymbol{d}\|},
\end{aligned} \tag{21}$$

where $\boldsymbol{d}$ is a random vector distributed between $[-1, 1]$ with length $N_P$, i.e., $\boldsymbol{d} = rands(N_P, 1)$; $\hat{\boldsymbol{\delta}}_{il_k}$, $\hat{\boldsymbol{\delta}}_{ir_k}$ and $\hat{\boldsymbol{\delta}}_{i_k}$ are the left antenna, the right antenna and the centroid of the $i$th beetle at $k$ iteration, respectively; $J_{il_k}$ and $J_{ir_k}$ are the cost function values of $\hat{\boldsymbol{\delta}}_{il_k}$ and $\hat{\boldsymbol{\delta}}_{ir_k}$, respectively; $d^k$ is the set distance between the two antennas at $k$ iteration; $c$ is the ratio between the beetles step and $d^k$.

After each beetle has finished searching in one iteration, the global optimal position $\hat{\boldsymbol{\delta}}_g$ and the cost function value $J_g$ of the beetle swarm are updated based on greedy-strategy:

$$J_g = J_{i_k}, \hat{\boldsymbol{\delta}}_g = \hat{\boldsymbol{\delta}}_{i_k} \quad if \;\; J_{i_k} < J_g, \tag{22}$$

where $J_{i_k}$ is the current fitness value of the centroid position $\hat{\boldsymbol{\delta}}_{i_k}$ of the $i$th beetle. In addition, the so-called $\varepsilon$ - strategy is commonly used for attenuation of $d^k$ to improve the search ability as:

$$\begin{cases} l^k = \eta \cdot l^{k-1} \\ d^k = l^{k-1} + d_{\min} \end{cases} \quad if \; r_d > \varepsilon \cap \min_{k=1,2,\cdots,n} J_{i_k} \geq J_g. \tag{23}$$

where $d_{\min}$ is the set minimum of $d^k$; $n$ is the number of beetles; $0 \leq \eta \leq 1$ is the attenuation coefficient; $l$ is the set attenuation range; $0 < \varepsilon < 1$ is the set probability of the beetles to miss their best positions. Then the optimization problem in (20) can be solved with the original BSAS algorithm as denoted in Algorithm 1.

**Algorithm 1** Predictive collision avoidance based on original BSAS

**Input:** The current time, $t$; The state of the own ship and the other ship at the current time, $\boldsymbol{X}_s(t)$ and $\boldsymbol{X}_T(t)$;
**Output:** Result of collision avoidance at the current time, $\delta(t)$;

```
if t < t_end then
    Initialize the parameters of BSAS, i.e.,c,η,d_min, the number of iteration m, the number of the beetles n and the centroid of each beetle δ̂_{1_0}, δ̂_{2_0}, ··· , δ̂_{n_0}.
    Use δ̂_{1_0}, δ̂_{2_0}, ··· , δ̂_{n_0}, the ship states X_s(t) and X_T(t) and the ship motion model (4) to predict collision risks based on (5~14), then calculate the cost function values based on (15) and (16) to obtain the initial fitness of each beetle.
    Calculate the global optimal position δ̂_g and the cost function value f_g based on (22).
    while k ≤ m do
        for each beetle i ← 1 to n, where n is the beetle number do
            Update the current position δ̂_{i_k} based on (21).
        end for
        for each beetle i ← 1 to n, where n is the beetle number do
            Update the global optimal position δ̂_g and the cost function value f_g and each δ̂_{i_k} based on (22)
        end for
        Update d^k according to (23);
        k = k + 1
    end while
    Make δ(t) = δ̂_g (1).
    return δ(t)
end if
```

### 3.3.2. Improvement based on historical optimum of antennas

In this section, the original BSAS is improved based on historical optimums of two antennas. It can be seen from (21) that the locations of the two antennas are always constrained by the

position of the current centroid in the original BSAS, which can easily lead to local optima. The historical optimums of two antennas and the corresponding optimal fitness values are introduced to improve the update strategy of both the antennas and centroid as:

$$\begin{aligned}
\hat{\boldsymbol{\delta}}_{il_k} =& \hat{\boldsymbol{\delta}}_{il\text{best}} + \frac{1}{2}d^k \cdot \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|},\\
\hat{\boldsymbol{\delta}}_{ir_k} =& \hat{\boldsymbol{\delta}}_{ir\text{best}} - \frac{1}{2}d^k \cdot \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|},\\
\hat{\boldsymbol{\delta}}_{i_k} =& \hat{\boldsymbol{\delta}}_{i_{k-1}} + cr_d\left(\frac{\|J_{ir\text{best}}\|}{\|J_{il\text{best}}\| + \|J_{ir\text{best}}\|}\right)\left(\hat{\boldsymbol{\delta}}_{il\text{best}} - \hat{\boldsymbol{\delta}}_{i_{k-1}}\right)\\
&+ cr_d\left(\frac{\|J_{il\text{best}}\|}{\|J_{il\text{best}}\| + \|J_{ir\text{best}}\|}\right)\left(\hat{\boldsymbol{\delta}}_{ir\text{best}} - \hat{\boldsymbol{\delta}}_{i_{k-1}}\right),
\end{aligned} \tag{24}$$

where $r_d \in [0,1]$ is a random value, $\hat{\boldsymbol{\delta}}_{il\text{best}}$ and $\hat{\boldsymbol{\delta}}_{ir\text{best}}$ are the historical optimums of the left antennas and right antennas, respectively; $J_{il\text{best}}$ and $J_{ir\text{best}}$ are the cost function values of $\hat{\boldsymbol{\delta}}_{il\text{best}}$ and $\hat{\boldsymbol{\delta}}_{ir\text{best}}$ , respectively. Then, the $\hat{\boldsymbol{\delta}}_{il\text{best}}$, $\hat{\boldsymbol{\delta}}_{ir\text{best}}$ and the global $\hat{\boldsymbol{\delta}}_g$ are updated as:

$$\begin{cases}
J_{il\text{best}} = J_{il_k}, \hat{\boldsymbol{\delta}}_{il\text{best}} = \hat{\boldsymbol{\delta}}_{il_k} & if(J_{il_k} \le J_{il\text{best}}),\\
J_{ir\text{best}} = J_{ir_k}, \hat{\boldsymbol{\delta}}_{ir\text{best}} = \hat{\boldsymbol{\delta}}_{ir_k} & if(J_{ir_k} \le J_{ir\text{best}}),\\
J_g = J_{il_k}, \hat{\boldsymbol{\delta}}_g = \hat{\boldsymbol{\delta}}_{il_k} & if(J_{il_k} \le J_g),\\
J_g = J_{ir_k}, \hat{\boldsymbol{\delta}}_g = \hat{\boldsymbol{\delta}}_{ir_k} & if(J_{ir_k} \le J_g),\\
J_g = J_{ik}, \hat{\boldsymbol{\delta}}_g = \hat{\boldsymbol{\delta}}_{ik} & if(J_{ik} \le J_g),
\end{cases} \tag{25}$$

Compared with the update strategy of the original BSAS, the two antennas of each beetle have less position constraints. Besides, one-step movement of a set distance toward the current best direction in the original BSAS has been changed to two-step movements as shown in Table 2. It can be seen from Table 2 that the improved beetle will move a certain distance towards the historical optimal position of two antennas of which the length is determined by the optimal fitness value. Therefore, the improved strategy guarantees that the best position of the centroid is approaching the antenna which has the optimal ability in historical searching, rather than the current better antenna. According to (17), to make sure that the $\hat{\boldsymbol{\delta}}_{il\text{best}}$ and $\hat{\boldsymbol{\delta}}_{il\text{best}}$ are located on two sides of $\hat{\boldsymbol{\delta}}_i$, the initial historical optimal position of two antennas are as:

$$\begin{cases}
\underset{j=1,2,\cdots N_P}{\hat{\boldsymbol{\delta}}_{il\text{best}}}[j] = -\delta_{\max}, \underset{j=1,2,\cdots N_P}{\hat{\boldsymbol{\delta}}_{ir\text{best}}}[j] = \delta_{\max} & if\ (67.5^\circ \le \theta_T \le 355^\circ),\\
\underset{j=1,2,\cdots N_P}{\hat{\boldsymbol{\delta}}_{il\text{best}}}[i] = 0, \underset{j=1,2,\cdots N_P}{\hat{\boldsymbol{\delta}}_{ir\text{best}}}[j] = \delta_{\max} & otherwise.
\end{cases} \tag{26}$$

### 3.3.3. Behavior decision based on Q-learning

In addition to the exploration of the individual beetle, the swarm behaviors of the beetles, i.e., the attenuation of step and the feedback of global optimal position, are also important in the optimization process.

**1) Attenuation of the $d^k$**

The set valve of $\varepsilon$ used for attenuation of $d^k$ in (23) will have a greater impact on the search ability of BSAS. To obtain a better attenuation approach of $d^k$, each individual beetle is considered to be able to decide whether $d^k$ needs to decrease or not based on its own optimization performance. Then a step vector $\boldsymbol{D}^k = \left[d_1^k, d_2^k, \cdots, d_n^k\right]^{\mathrm{T}}$ is defined for the beetle swarm, and (23) is changed to:

$$\begin{cases}
l_i^k = \eta \cdot l_i^k,\\
d_i^k = l_i^k + d_{\min},
\end{cases} \quad if\ r_d > \varepsilon \cap \min_{k=1,2,\cdots,n} J_{i_k} \ge J_g, \tag{27}$$

Then the improved update strategy of $\hat{\boldsymbol{\delta}}_{il_k}$ and $\hat{\boldsymbol{\delta}}_{ir_k}$ in (21) is modified as:

$$\begin{aligned}
\hat{\boldsymbol{\delta}}_{il_k} =& \hat{\boldsymbol{\delta}}_{il\text{best}} + \frac{1}{2}d_i^k \cdot \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|},\\
\hat{\boldsymbol{\delta}}_{ir_k} =& \hat{\boldsymbol{\delta}}_{ir\text{best}} - \frac{1}{2}d_i^k \cdot \frac{\boldsymbol{d}}{\|\boldsymbol{d}\|},\\
\hat{\boldsymbol{\delta}}_{i_k} =& \hat{\boldsymbol{\delta}}_{i_{k-1}} + cr_d\left(\frac{\|J_{ir\text{best}}\|}{\|J_{il\text{best}}\| + \|J_{ir\text{best}}\|}\right)\left(\hat{\boldsymbol{\delta}}_{il\text{best}} - \hat{\boldsymbol{\delta}}_{i_{k-1}}\right)\\
&+ cr_d\left(\frac{\|J_{il\text{best}}\|}{\|J_{il\text{best}}\| + \|J_{ir\text{best}}\|}\right)\left(\hat{\boldsymbol{\delta}}_{ir\text{best}} - \hat{\boldsymbol{\delta}}_{i_{k-1}}\right).
\end{aligned} \tag{28}$$

**2) Exploitation of the global optimal position $\hat{\boldsymbol{\delta}}_g$**

In (Wang et al., 2018a), the PSO algorithm is combined with BSAS, and a velocity update strategy is introduced for BSAS. Note that the PSO algorithm uses a certain number of particles for optimization. In each iteration, the velocity of each particle is updated by learning from the global and local optimal positions, so as to update the positions of the swarm (Wang et al.,

Table 2: Movement changes of the centroid in each iteration after improvement

| Each iteration | Original strategy | Improved strategy | |
|---|---|---|---|
| | | Step1 | Step2 |
| Distance | $c \cdot d^k$ | $cr_d\left(\frac{\|J_{i\hat{r}\text{best}}\|}{\|J_{i\hat{l}\text{best}}\|+\|J_{i\hat{r}\text{best}}\|}\right)\left\|\hat{\boldsymbol{\delta}}_{i\hat{l}\text{best}}-\hat{\boldsymbol{\delta}}_{i_{k-1}}\right\|$ | $cr_d\left(\frac{\|J_{i\hat{l}\text{best}}\|}{\|J_{i\hat{l}\text{best}}\|+\|J_{i\hat{r}\text{best}}\|}\right)\left\|\hat{\boldsymbol{\delta}}_{i\hat{r}\text{best}}-\hat{\boldsymbol{\delta}}_{i_{k-1}}\right\|$ |
| Direction | $sign\left(J_{il_k}-J_{ir_k}\right)\frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}$ | $\frac{\left(\hat{\boldsymbol{\delta}}_{i\hat{l}\text{best}}-\hat{\boldsymbol{\delta}}_{i_{k-1}}\right)}{\left\|\hat{\boldsymbol{\delta}}_{i\hat{l}\text{best}}-\hat{\boldsymbol{\delta}}_{i_{k-1}}\right\|}$ | $\frac{\left(\hat{\boldsymbol{\delta}}_{i\hat{r}\text{best}}-\hat{\boldsymbol{\delta}}_{i_{k-1}}\right)}{\left\|\hat{\boldsymbol{\delta}}_{i\hat{r}\text{best}}-\hat{\boldsymbol{\delta}}_{i_{k-1}}\right\|}$ |

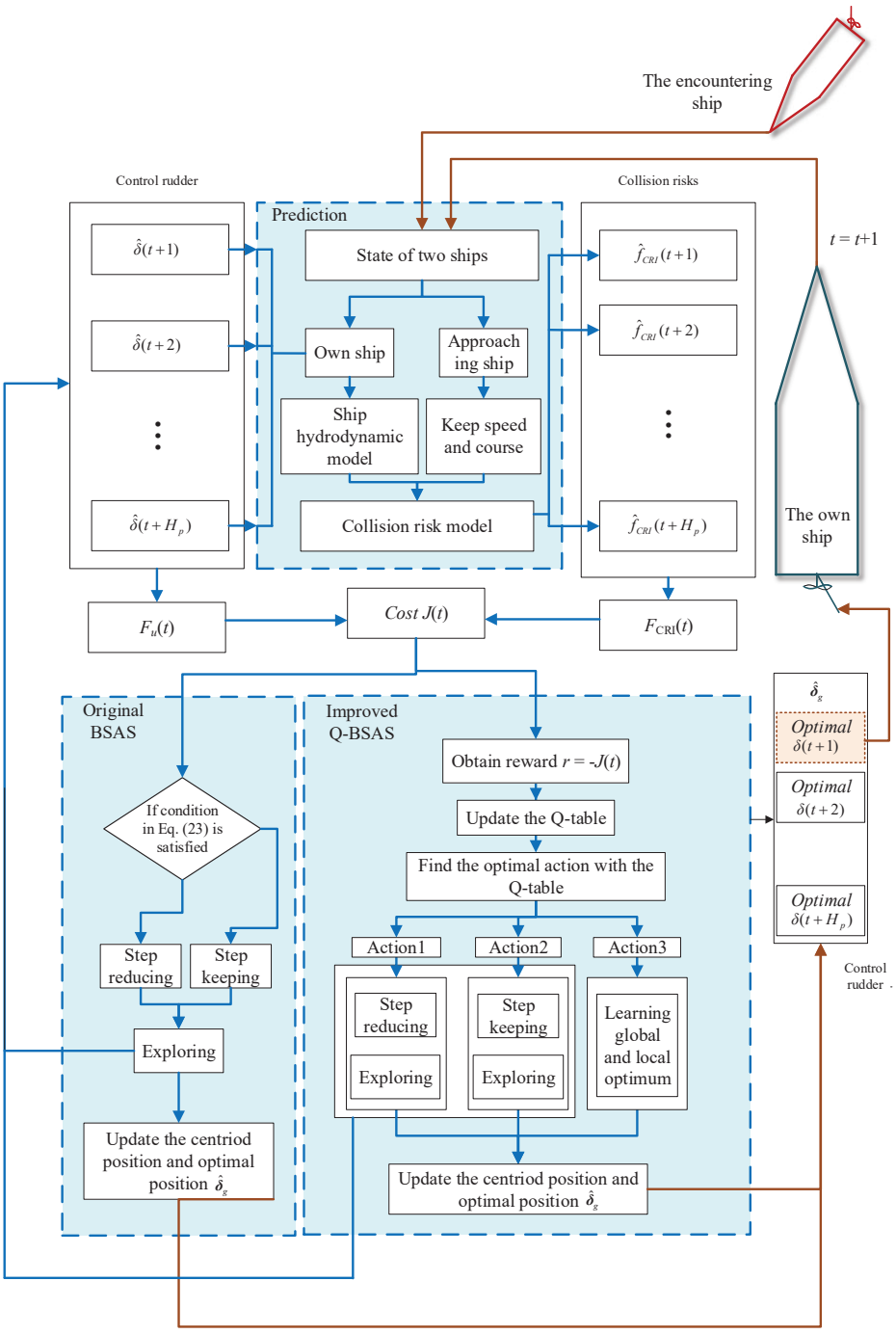


Figure 5: A schema of I-Q-BSAS and original BSAS based predictive collision avoidance.

2018b). Therefore, the exploitation of the global optimal position in PSO is reflected in the velocity of the particle. At the same time, a velocity weight is used to achieve the balance of exploration and exploitation.

However, the BSAS algorithm uses random directions of the antennas and the step of the centroid for exploration, which is different from PSO. The balance of exploration and exploitation in BSAS is achieved by the step attenuation $\eta$ in (23). Since the step attenuation strategy is adopted separately, it is considered to remove the velocity weight introduced in (Wang et al., 2018a) and directly learn the optimal position on the centroid position as:

$$\hat{\boldsymbol{\delta}}_{i_k} = \hat{\boldsymbol{\delta}}_{i_{k-1}} + \zeta_1 r_d(\hat{\boldsymbol{\delta}}_{ibest} - \hat{\boldsymbol{\delta}}_{i_{k-1}}) + \zeta_2 r_d(\hat{\boldsymbol{\delta}}_g - \hat{\boldsymbol{\delta}}_{i_{k-1}}), \tag{29}$$

where $r_d \in [0, 1]$ is a random value; $\hat{\boldsymbol{\delta}}_{ibest}$ is the historical local optimal position of the centroid, which is updated as (30); $\zeta_1$ and $\zeta_2$ are the learning factors of $\hat{\boldsymbol{\delta}}_{ibest}$ and $\hat{\boldsymbol{\delta}}_g$ , respectively.

$$\begin{cases} J_{ibest} = J_{i_k} \\ \hat{\boldsymbol{\delta}}_{ibest} = \hat{\boldsymbol{\delta}}_{i_k} \end{cases} \quad if \ J_{i_k} < J_{ibest}. \tag{30}$$

where $J_{ibest}$ is the best fitness value of each $\hat{\boldsymbol{\delta}}_{ibest}$.

**3) Behaviors decision based on Q-learning**

In order to choose the best behavior of each individual particle, a common reinforcement learning method, i.e., Q-learning, has been used for the optimization and decision in PSO algorithm (Samma et al., 2016). It is considered that the Q-learning method can choose the better behavior of each beetle based on the long-term performance compared with the adjusting method based on the current performance in (Wang and Chen, 2018). Therefore, the Q-learning method is applied for the behavior decision of each beetle by regarding the behavior decision process as a typical Markov decision-making process $M = \left\{s^M, a^M, r^M\right\}$, where $s^M, a^M$ and $r^M$ are the state, action and reward, respectively.

With respect to the state $s^M$, since the exploration ((27) and (28)) and exploitation ((29)) are the main behaviors of each beetle, the state $s_M$ of the MDP for behavior decision is defined as:

$$s^M = \begin{cases} 1 & \text{Exploring with step reducing based on (27) and (28),} \\ 2 & \text{Exploring without step reducing based on (28),} \\ 3 & \text{Learning the global and local optimum based on (29),} \end{cases} \tag{31}$$

where number 1, 2 and 3 are the codes of the behaviors.

With respect to the action $a^M$, since the target of behavior decision is to fix the beetle's current behavior to a deterministic state $s^M$, the action $a^M$ is defined as:

$$a^M = \begin{cases} 1 & s^M = 1, \\ 2 & s^M = 2, \\ 3 & s^M = 3, \end{cases} \tag{32}$$

With respect to the reward $r^M$, considering that the difficulties of contributions to the local optimal $J_{ibest}$ and the global optimal $J_g$ are different, two different positive rewards are set as shown in (33), which is different from a signal reward in (Samma et al., 2016).

$$r^M = \begin{cases} 1 & if \ J_{i_k} < J_{ibest}, \\ 2 & if \ J_{i_k} < J_g, \\ -1 & other, \end{cases} \tag{33}$$

where $r_M$ is the immediate reward for each state-action pair $(s^M, a^M)$.

To realize Q-learning, a Q-table $Q_k(s_k^M, a_k^M)$ is created for each beetle to calculate the value of the state-action pair $(s_k^M, a_k^M)$ at $k$th iteration. After each beetle takes an action $a_k^M$ for searching at the $k$th iteration, the next state $s_{k+1}^M$ and immediate reward $r_{k+1}^M$ is obtained. Then, the Q-table at the next iteration $Q_{k+1}(s_{k+1}^M, a_{k+1}^M)$ is calculated as:

$$\begin{aligned} &Q_{k+1}(s_{k+1}^M, a_{k+1}^M) = Q_k\left(s_k^M, a_k^M\right) \\ &+ \alpha^M\left[r_{k+1}^M + \gamma^M \max Q_k\left(s_{k+1}^M, a^M\right) - Q_k\left(s_k^M, a_k^M\right)\right], \end{aligned} \tag{34}$$

where $\alpha^M$ and $\gamma^M$ are the learning rate and discount factor of Q-learning. With the Q-table, the action $a_k^M$ is selected by a so-called $\epsilon$-policy to balance the exploration and exploitation:

$$a_k^M = \begin{cases} \arg\max Q_k^M\left(s_k^M, a^M\right) & r_d < \varepsilon, \\ f_{rand}\{1, 2, 3\} & r_d \geq \varepsilon. \end{cases} \tag{35}$$

where $f_{rand}\{\cdot\}$ represents random selection function, $r_d \in [0, 1]$ is a random value.

The final improved Q-learning-BSAS, i.e., I-Q-BSAS, algorithm is proposed and listed as Algorithm 2. The convergence and global optimality analysis of the improved BSAS are given in Appendices A. To better illustrate the differences between the proposed I-Q-BSAS based collision avoidance and original BSAS based collision avoidance, a schema of Algorithm 1 and Algorithm 2 is shown in Fig. 5. For comprehensive verification of the proposed I-Q-BSAS, standard optimization tests with benchmark functions are given in Appendix B.

**Algorithm 2** Predictive collision avoidance based on Improved Q-BSAS

**Input:** The current time, $t$; The state of the own ship and the other ship at the current time, $\boldsymbol{X}_s(t)$ and $\boldsymbol{X}_T(t)$;
**Output:** Result of collision avoidance at the current time, $\delta(t)$;
**if** $t < t_{end}$ **then**
  Initialize the parameters of the improved Q-BSAS, i.e.,$c$,$\eta$,$d_{\min}$,$\zeta_1$,$\zeta_2$, $\boldsymbol{D}$ the number of iteration $m$, the number of the beetles $n$, the centroid of each beetle $\hat{\boldsymbol{\delta}}_{1_0}, \hat{\boldsymbol{\delta}}_{2_0}, \cdots, \hat{\boldsymbol{\delta}}_{n_0}$ and the Q-table for each beetle.
  **for** each beetle $i \leftarrow 1$ to $n$, where $n$ is the beetle number **do**
    Initialize the historical best position of the left antenna $\hat{\boldsymbol{\delta}}_{il\text{best}}$ and the right antenna $\hat{\boldsymbol{\delta}}_{ir\text{best}}$ of each beetle based on (26).
    Use $\hat{\boldsymbol{\delta}}_{i_0}$ , $\hat{\boldsymbol{\delta}}_{il\text{best}}$ , $\hat{\boldsymbol{\delta}}_{ir\text{best}}$ , the ship states $\boldsymbol{X}_s(t)$ and $\boldsymbol{X}_T(t)$ and the ship motion model (4) to predict risks based on (4~14), then calculate the cost function values based on (15) and (16) to obtain the corresponding fitness $J_{i_0}$ , $J_{il\text{best}}$ and $J_{ir\text{best}}$, respectively.
    Calculate the initial local optimal positions $\hat{\boldsymbol{\delta}}_{ibest}$ and fitness $J_{ibest}$ based on (30).
    Calculate the initial global optimal positions $\hat{\boldsymbol{\delta}}_g$ and fitness $J_g$ based on (25).
  **end for**
  **while** $k \le m$ **do**
    **for** each beetle $i \leftarrow 1$ to $n$, where $n$ is the beetle number **do**
      Update the two antennas positions $\hat{\boldsymbol{\delta}}_{il_k}, \hat{\boldsymbol{\delta}}_{ir_k}$ based on (28).
      Update the historical best positions of the left antennas $\hat{\boldsymbol{\delta}}_{il\text{best}}$ and right antennas $\hat{\boldsymbol{\delta}}_{ir\text{best}}$ , and the corresponding fitness $J_{il\text{best}}$ and $J_{ir\text{best}}$ based on (25).
      Use Q-Learning method and (33) to choose the current optimal action of each beetle and obtain the best state of each beetle based on (35).
      **switch** the best state **do**
        **case** 1
          Update the centroid position $\hat{\boldsymbol{\delta}}_{i_k}$ based on (28).
        **case** 2
          Update the centroid position $\hat{\boldsymbol{\delta}}_{i_k}$ based on (28).
          Update $d_i$ based on (27).
        **case** 3
          Update the centroid position $\hat{\boldsymbol{\delta}}_{i_k}$ based on (29).
      Update the local optimal position$\hat{\boldsymbol{\delta}}_{ibest}$ and fitness $J_{ibest}$ based on (30).
      Update the global optimal position $\hat{\boldsymbol{\delta}}_g$ and fitness $J_g$ based on (25).
    **end for**
    $k = k + 1$
  **end while**
  Make $\delta(t) = \hat{\boldsymbol{\delta}}_g(1)$.
  **return** $\delta(t)$
**end if**

## 4. Reinforced inverse method based on neural networks

Based on the proposed I-Q-BSAS algorithm and MPC strategy, collision avoidance in typical encounters can be realized efficiently. Note that the MPC strategy still needs to solve the optimization problem at each time step, which will increase the time cost in multi-ship encounters. Referring to the function approximation approach in reinforcement learning, we use neural networks to learn an inverse relationship model between the inputs and outputs of the proposed MPC strategy in this section, which is regarded as an inverse approximation of the proposed MPC strategy. Then the time cost can be reduced by end-to-end output of the neural network model. Besides, the network model is re-trained in the long term based on an experience reply technique to deal with the uncertain dynamics problem.

### 4.1. Direct inverse system and optimal policy

Conventionally, the direct inverse system of an original system is obtained by establishing an inverse relationship model between the output and input of the original system directly (Dirion et al., 1995). Then the direct inverse system of the collision avoidance system can be established by constructing the inverse relationship between the original system input (i.e., the rudder), the system states (i.e., ship motion, DCPA, TCPA) and the target (i.e., target CRI). Then, the obtained inverse system can be connected with the original system to form a pseudo linear system as shown in Fig. 6.

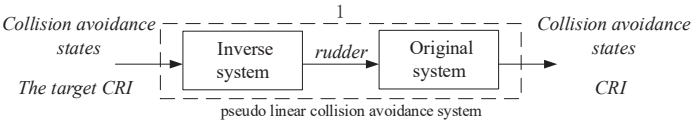


Figure 6: Inverse model of collision avoidance.

It can be seen from (4)~(15) that the direct inverse system of the ship collision avoidance process has a strong nonlinearity and cannot be expressed clearly. As a widely used black-box modeling method, neural networks (Dirion et al., 1995; He et al., 2016) or deep neural networks (Finn et al., 2016) have been successfully applied in the inverse control area, and its strong approximation ability for arbitrary nonlinear models has been verified. In this study, a back propagation neural network (BPNN) is used to approximate the relationship between the input and output of the direct inverse system for collision avoidance, i.e., the inverse model.

Firstly, representative state features in collision avoidance can be extracted to construct the sample pairs for the inverse model. Since a set of relative motion states can correspond to several sets of original motion states of the ships, the relative parameters (i.e., DCPA, TCPA, relative distance $R_T$, relative position direction $\theta_T$, relative heading direction $C_T$, relative speed ratio $K$) are more suitable for training than the original ship motions ($\boldsymbol{X}$, $\boldsymbol{X}_T$). Thus, the following sample pair is designed for the inverse model:

$$\begin{cases} S_{in}(t) = [DCPA(t), TCPA(t), R_T(t), \theta_T(t), C_T(t), \\ \qquad K(t), f_{CRI}(t+1)], \\ S_{out}(t) = [\delta(t)], \end{cases} \tag{36}$$

where $S_{in}(t)$ and $S_{out}(t)$ are the sample input and output, respectively. Then, a parameterized conditionally policy $\mu_{\boldsymbol{\theta}}(S)$ represented by a neural network with weight matrix $\boldsymbol{\theta}$ can be trained by minimizing the least square error as:

$$\min_{\theta} \frac{1}{2} \sum \| S_{out}(t) - \mu_{\boldsymbol{\theta}}(S_{in}(t)) \|_2^2. \tag{37}$$

After training with the data samples based on the proposed I-Q-BSAS predictive avoidance method in typical encounters, the optimal control value of the own ship can be output directly from the policy $\mu_{\boldsymbol{\theta}}(S)$ by setting a minimum collision risk at the future moment $f_{CRI}(t+1) = f_{CRI\,\min}$, so as to achieve the inverse collision avoidance for each encountering ship.

### *4.2. Reinforced inverse method for multi-ship encounters*

Conventional optimization method for multi-ship collision avoidance encounters needs to optimize the cost including the CRI between the own ship and each encountering ship (Lazarowska, 2015), which suffers from a large computation and difficulty of finding an optimum in multi-object optimization. In this study, the direct inverse model (i.e., the optimal policy) in typical encounters is used to design a general multi-ship collision avoidance method with less computation and reinforced based on on-policy data. Firstly, the weighting method is applied on the outputs of the inverse model based on CRI for multi-ship encounters. Then, the on-policy data are collected and aggregated with the data generated by pre-simulation for long-term re-training of the inverse model.

**1) Output weighting based on CRI**

To deal with general control problems, the inverse model is usually combined with a conventional feedback method (e.g., PID). Then the outputs of the inverse model and feedback method can be combined by weighting method directly to obtain the final output (Son et al., 2017). In addition, the weights of different outputs are commonly set based on the control error.

For collision avoidance, the weighting method can also be applied to combine different outputs of the inverse model with different encountering ships. Then, the error between the current risk and the target risk $\tilde{f}_{CRI}(t) = f_{CRI}(t) - f_{CRI\,\min}$ is defined as the collision risk error, and the final output can be obtained as:

$$\delta(t) = \sum_{i=1}^{n_s} \frac{\tilde{f}_{CRI_i}(t)}{\sum_{i=1}^{n_s} \tilde{f}_{CRI_i}(t)} \cdot \delta_i(t). \tag{38}$$

where $n_s$ is the number of the encountering ships; $\tilde{f}_{CRI_i}(t)$ is the collision risk error of the $i$th encountering ship with the own ship; $\delta_i(t)$ is the output of the inverse model for the $i$th encountering ship.

**2) Long-term learning for the optimal policy**

In practical navigation environment, uncertain disturbances (e.g., ship dynamics perturbation) will change the optimal policy over time. In order to improve the performance of the inverse method, both off-policy and on-policy datasets, i.e., the dataset $D_{MPC}$ of simulation or actual collision avoidance in typical encounters based on Algorithm 2 and the dataset $D_S$ in actual multi-ship avoidance based on the inverse model, can be aggregated for long-term retraining of the optimal policy $\mu_{\boldsymbol{\theta}}(S)$ at every time step $N_{re}$, which can mitigate the mismatch between the distribution of $D_{MPC}$ and the actual data (Nagabandi et al., 2018).

Furthermore, to overcome the problems of correlation data and non-stationary distribution of data. The inverse model is

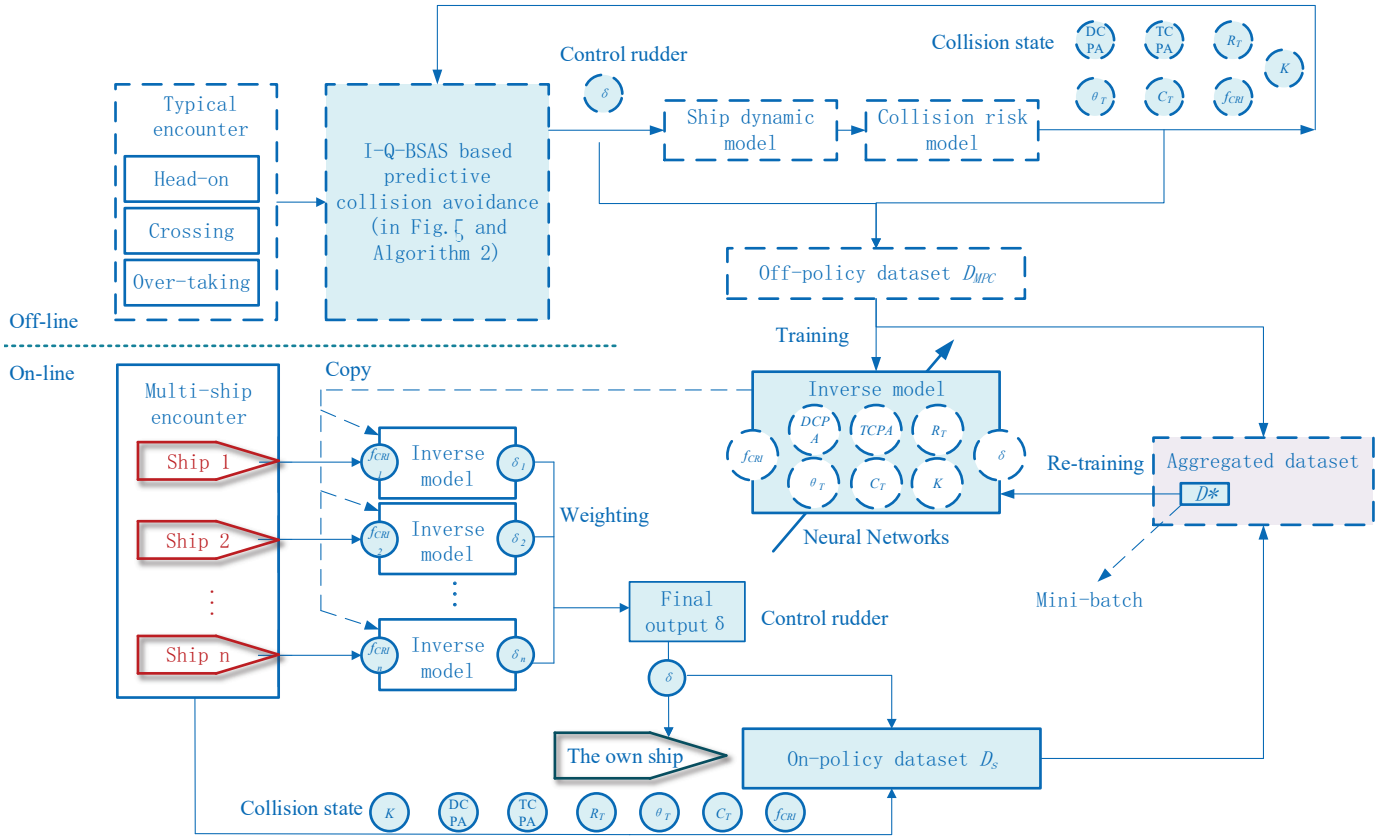


Figure 7: The work flow of the reinforced inverse collision avoidance method for multi-ship encounters.

**Algorithm 3** Reinforced inverse collision avoidance method for multi-ship encounters

**for** each typical encounters **do**
  Initial a virtual encountering ship to form the encounter scene.
  Use the proposed improved Q-BSAS algorithm for collision avoidance simulation based on Algorithm 2
  Collect the simulate sample data with the form of (36) as dataset $D_{MPC}$
**end for**
Use a neural network to train the data $D_{MPC}$ and obtain the policy $\mu_{\theta}(S)$.
Initialize the sample pool $D_s$ and the minimum size of $D_s$ for training $n_{D_s}$.
**while** $t < t_{end}$ **do**
  **for** Each encountering ship $i \leftarrow 1$ to $n_s$, where $n_s$ is the number of the encountering ships **do**
    Calculate the relative motion parameters based on (5)~(13) and the collision risk $f_{CRI_i}(t)$ based on (14).
    Use the trained policy $\mu_{\theta}(S)$ to output the optimal input $\delta_i(t)$, directly.
  **end for**
  Calculate the final optimal input $\delta(t)$ based on (38) and conduct the collision avoidance.
  Collect the actual sample data with each encountering ship, and add it into $D_s$.
  **if** the size of $D_s$ is larger than $n_{D_s}$ and $t = k \cdot N_{re}$ (where $k$ is a positive integer) **then**
    Aggregate the data in $D_s$ with size of $n_{D_s}$ and $D_{MPC}$ for retraining the policy $\mu_{\theta}(S)$ for $N_{ep_{\min}}$ epochs.
    **while** The training epoch $N_{ep} < N_{ep_{\max}}$ and **do**
      Continuously train the policy $\mu_{\theta}(S)$ for $N_{ep_{\min}}$ epochs.
      $N_{ep} = N_{ep} + N_{ep_{\min}}$
      Use the output of the current policy for one-step state prediction and calculate the state constraints.
      **if** The state constraints in (20) are satisfied **then**
        Break while and end the re-training process.
      **end if**
    **end while**
    **end if**
    $t = t + 1$.
  **end while**

trained by random sampling from previous state transition (i.e., the experience replay policy (Mnih et al., 2015)). Then, a mini-batch dataset $D*$ selected randomly from $D_S$ is aggregated with $D_{MPC}$. By this long-term learning, an adaptive optimal policy $\mu_{\theta}(S)$ can be obtained without huge computation. During the retraining, the control constraints on $\delta_i(t)$ in (17) are also

considered based on the saturation approach and the state constraints are added to the end conditions of the re-training. A minimum training epoch $N_{ep_{\min}} = 10$ and a maximum epoch $N_{ep_{\max}} = 100$ are set for each re-training time in this study. At every time step $N_{re}$, the re-training process is conducted for at least $N_{ep_{\min}}$ epochs and stopped if the state constraints are satisfied or the maximum epoch $N_{ep_{\max}}$ is reached.

To illustrate the relation ship between the proposed I-Q-BSAS based predictive collision avoidance algorithm and the reinforced inverse method, the work flow of the reinforced inverse method for multi-ship encounters is shown in Fig. 7 and denoted in Algorithm 3.

In summary, the optimal collision avoidance policy based on the proposed MPC strategy is approximated by an inverse neural network model. The MPC strategy generates superior trajectories for the training of the neural network model, and the output of the neural network is used directly for collision avoidance to reduce the time cost.

## 5. Simulation experiments

Simulation experiments for ship collision avoidance in multi-ship encounters are conducted using the KVLCC2 ship model. The cost function value $J$ in predictive collision avoidance is the fitness of the proposed I-Q-BSAS optimization method. A smaller fitness reflects a better optimization performance (Kennedy and Eberhart, 1995). Therefore, the optimal fitness values and real-time performance (i.e., the calculation time of each optimization process) are both compared in the experiments.

In addition, it is considered that the collision avoidance is completed when the collision risk $f_{CRI}$ between each encountering ship has been reduced to a certain threshold. In this study, the risk threshold was set to $f_{CRIT} = 0.01$, and the line-of-sight strategy is adopted for ship resuming when the collision avoidance is finished. In all simulation experiments, the control interval is set as $h = 0.5s$, the control horizon and prediction horizon are set as $h_C = 0.5s$ and $h_P = 4s$ (Liu et al., 2018). In this study, we consider more safety collision avoidance results referring to several related optimization-based and learning-based researches (Yin and Zhang, 2018; Chen and Huang, 2012a). The weights are set as $\mu_1 = 0.95$ and $\mu_2 = 0.05$ referring to the basic collision avoidance reward function in (Yin and Zhang, 2018).

Besides, the original BSAS and LDWPSO methods are used for comparisons in the simulation experiments, since LDWPSO is very similar to BSAS and has been widely used for comparisons (Xin et al., 2009; Taherkhani and Safabakhsh, 2016). Throughout this section, the OS and ES in each figure represent the own ship and encountering ship, respectively.

### 5.1. Collision avoidance in typical encounters

The initial relative state of the encountering ship is set based on the relative course angle $C_{T0}$, the position direction $\theta_{T0}$, the distance $R_{T0}$, and speed ratio $K_v$, as:

$$\begin{aligned}
&\boldsymbol{X}_{T0} = [x_{T0}, y_{T0}, \psi_{T0}, u_T, 0, 0],\\
&x_{T0} = x_0 + K_v R_{T0} \sin(\theta_{T0}),\\
&y_{T0} = y_0 + K_v R_{T0} \cos(\theta_{T0}),\\
&\psi_{T0} = \psi_0 + C_{T0},\\
&u_T = u_0 K_v,
\end{aligned} \tag{39}$$

where $\boldsymbol{X}_0 = [x_0, y_0, \psi_0, u_0, 0, 0]$ is the initial state of the own ship, which is set as $\boldsymbol{X}_0 = [0, 0, 0, 1.174, 0, 0]$, where 1.174 m/s is the service speed of KVLCC2 model ship. Since the scale of KVLCC2 is 45:1, the distance $R_{T0}$ in head-on, small angle crossing and overtaking encounters are set as $R_{T0}$ = 4nm/45 = 164m, and $R_{T0}$ in large angle crossing encounter is set as $R_{T0}$ = 2nm/45 = 82m to establish a more close-quarters scenario. $C_{T0}$, $\theta_{T0}$ and $K_v$ in different encounters are set as Table 3.

Table 3: Initial states in typical encounters.

| | Head-on | Crossing | | Over-taking |
|---|---|---|---|---|
| | | Small angle | Large angle | |
| $C_{T0}/^\circ$ | 180 | 220 | 270 | 0 |
| $\theta_{T0}/^\circ$ | 0 | 20 | 70 | 0 |
| $K_v$ | 1 | 1 | 0.8 | 0.3 |

Furthermore, the design parameters of the original BSAS, the proposed I-Q-BSAS and LDWPSO are given in Table 4. Based on the convergence analysis in Appendix A, the constants in BSAS and I-Q-BSAS are set as $c = 1$. The learning

Table 4: The design parameters of original BSAS, the proposed I-Q-BSAS and LDWPSO.

| | Common parameters | | | Parameters of BSAS and I-Q-BSAS | | | | | | Parameters of LDWPSO | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Parameters | $m$ | $P_{max}$ | $P_{min}$ | $n_{BSAS}$ | $\eta$ | $d_{min}$ | $c$ | $\zeta_1$ | $\zeta_2$ | $n_{PSO}$ | $w_v$ | $c_1$ | $c_2$ |
| Values | 15 | based on (20) | | 3 | 0.9 | 0.1 | 1 | 1.3 | 1.8 | 9 | 1-0.6*i/m [1] | 1.3 | 1.8 |
| Significance | Max iteration | Constrains | | Pop size | Attenuation coefficient | minimum step | step ratio | learning factors | | Pop size | weight of the speed | local learning factor | global learning factor |

[1] $i$ is the current iteration.

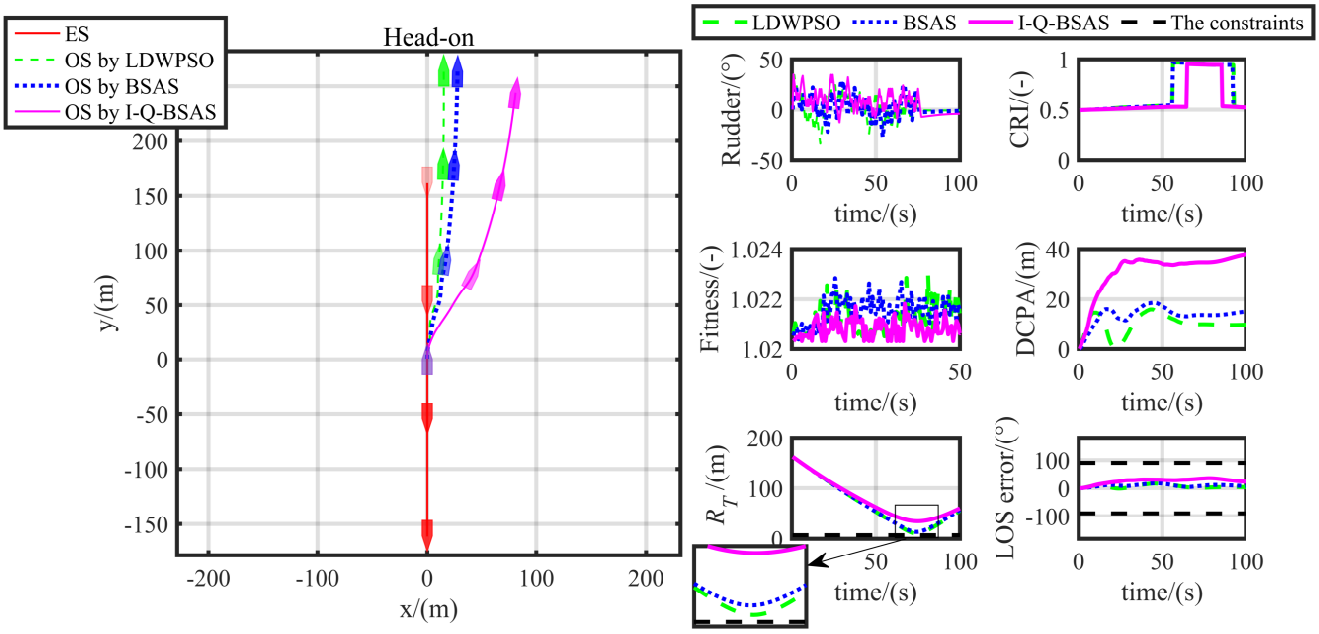


Figure 8: Trajectories, rudder angles and fitness values in head-on encounter.

factors of BSAS and I-Q-BSAS are set the same as those in the LDWPSO for fair comparisons. Other hyper-parameters are set referring to the BAS and LDWPSO baselines in (Jiang and Li, 2018; Wang and Chen, 2018; Xin et al., 2009). Note that the fitness is calculated three times per iteration in BSAS and I-Q-BSAS, so the population size of LDWPSO is set to three times that of BSAS and I-Q-BSAS for fairness.

Since the inverse model is trained with the samples generated by predictive collision avoidance in typical encounters as mentioned in section 4.1, both the performance of the predictive collision avoidance and the inverse model are verified.

### *5.1.1. Verification of the predictive collision avoidance*

The ship trajectories, the collision avoidance states (i.e., the rudder, CRI, relative distances and DCPA) and the mean optimal fitness values in 20 repeated tests in head-on, small angle crossing, large angle crossing and over-taking encounters are shown in Fig. 8, Fig. 9, Fig. 10 and Fig. 11, respectively. Besides, the computation time cost of the optimization process at each time step are shown in Fig. 12. Furthermore, the average optimal fitness values ($Ave_f$), the minimum relative distance ($Min_d$) and the maximum DCPA after avoidance ($Max_{DCPA}$) are regarded as the evaluation indicators of the optimization and collision avoidance performance. The average time cost ($Ave_T$) of the optimization is regarded as the indicator of real-time performance. The indicator results are calculated as shown in Table 5.

It can be seen form Fig. 8~Fig. 12 and Table 5 that:

1) With respect to the optimization performance, the proposed I-Q-BSAS algorithm can obtain smaller fitness values in all encounter scenarios compared with the original BSAS and LDWPSO algorithms, which indicates the better optimization performance of the proposed method for collision avoidance. It

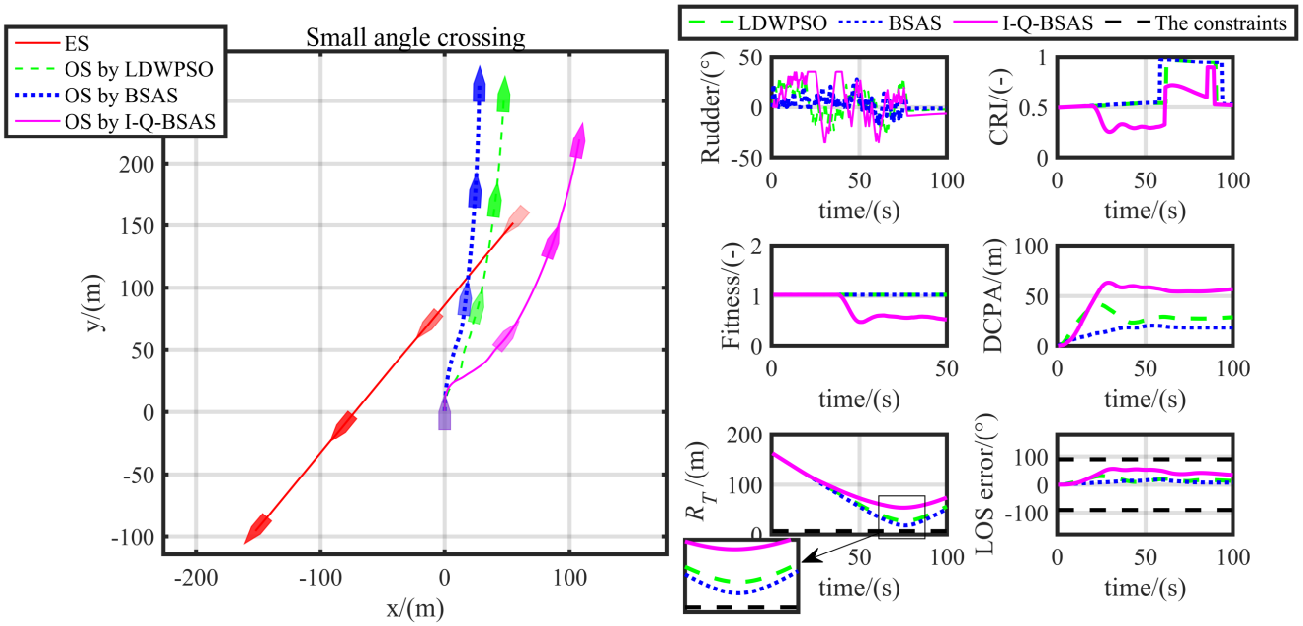


Figure 9: Trajectories, rudder angles and fitness values in small angle crossing encounter.

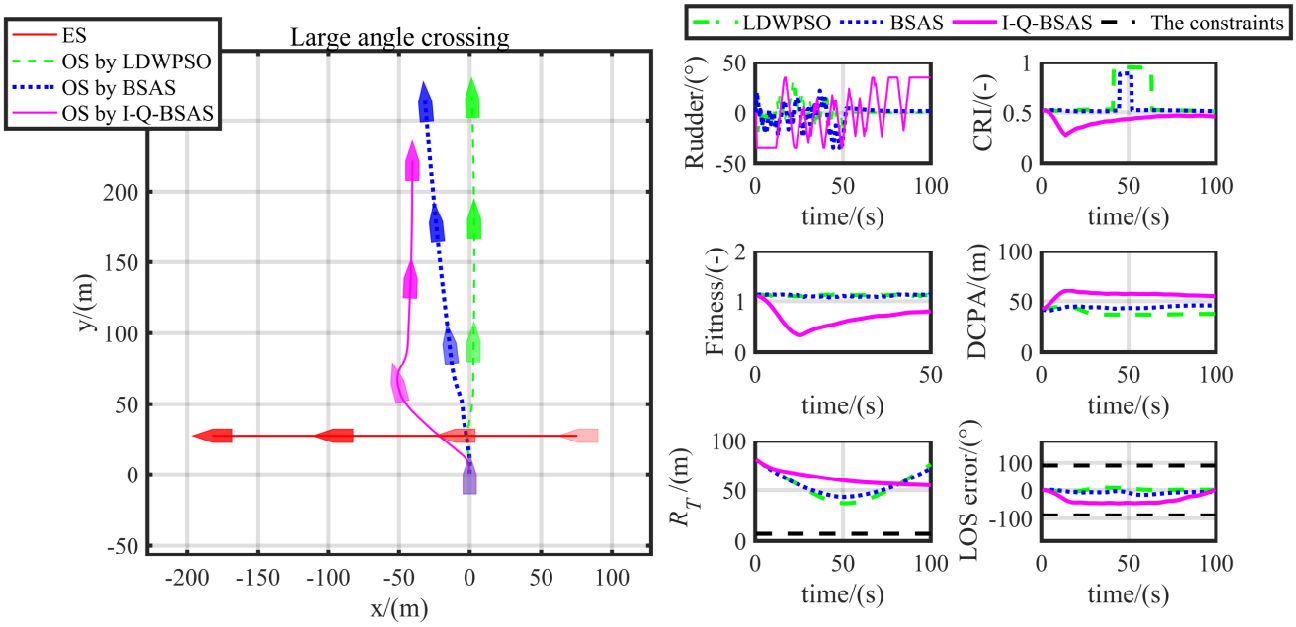


Figure 10: Trajectories, rudder angles and fitness values in large angle crossing encounter.

Table 5: The indicator results of collision avoidance in typical encounters.

| | $Ave_f$ | | | $Min_d$/m | | | $Max_{DCPA}$/m | | | $Ave_T$/s | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | I-Q-BSAS | LDWPSO | BSAS | I-Q-BSAS | LDWPSO | BSAS | I-Q-BSAS | LDWPSO | BSAS | I-Q-BSAS | LDWPSO | BSAS |
| ho[1] | **1.0208** | 1.0213 | 1.0215 | **34.603** | 9.772 | 13.544 | **59.640** | 16.206 | 24.544 | 0.337 | 0.466 | **0.319** |
| sac[1] | **0.7890** | 1.0239 | 1.0239 | **54.507** | 27.763 | 18.525 | **88.671** | 42.485 | 27.120 | 0.432 | **0.329** | 0.446 |
| lac[1] | **0.7847** | 1.1160 | 1.1081 | **54.370** | 36.951 | 43.243 | **60.282** | 44.842 | 49.438 | **0.336** | 0.506 | 0.337 |
| ot[1] | **1.1001** | 1.2294 | 1.2292 | **45.558** | 13.349 | 22.934 | **65.530** | 23.198 | 38.124 | 0.452 | 0.470 | **0.417** |

[1] ho: head-on; sac: small angle crossing; lac: large angle crossing; ot: over-taking

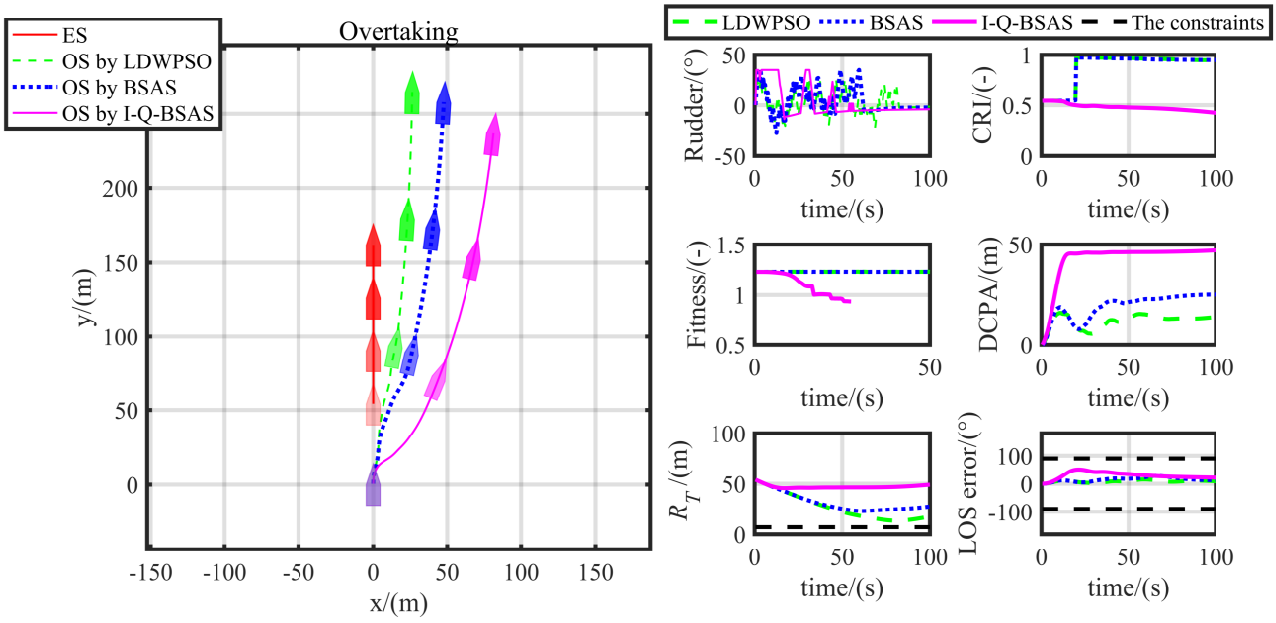


Figure 11: Trajectories, rudder angles and fitness values in over-taking encounter.

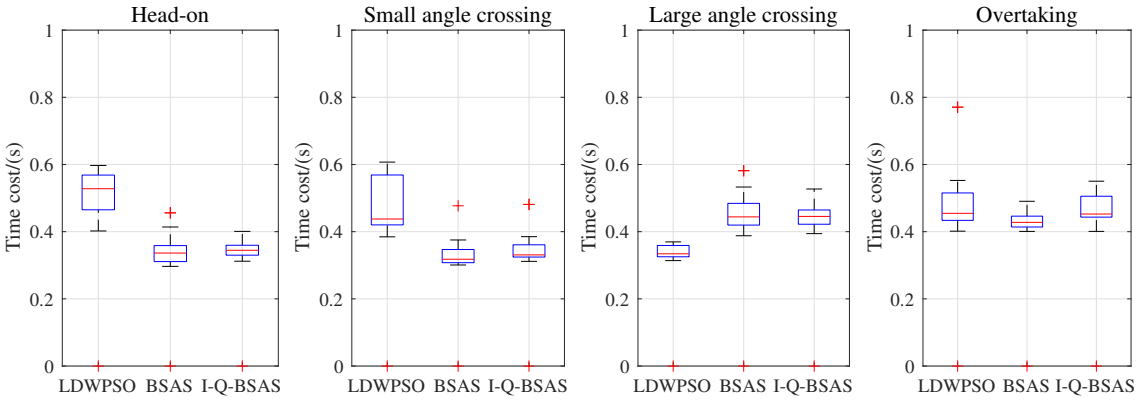


Figure 12: Time cost of each optimization process in typical encounters.

can be seen from the data in Table 5 that the minimum relative distances $Min_d$ and the maximum DCPAs $Max_{DCPA}$ of the I-Q-BSAS are much larger than those of BSAS with the setting weights, which are more obvious in head-on, small angle crossing encounters and over-taking. The possible reason is that the collision risks in head-on, small angle crossing and over-taking encounters are relatively higher than those in large angle crossing, which can be seen from the CRI curves around the minimum distance in Fig. 8~Fig. 12. Besides, as can be seen from the constraints of the relative distances $R_T$, LOS tracking error in Fig. 8~Fig. 12 and the control constraints in Fig. 13 that, both the state and control constraint conditions in (20) are satisfied in all encounters, which indicates that the proposed I-Q-BSAS obtains better collision avoidance results than BSAS and LDW-PSO with the same constraint conditions.

2) With respect to the real-time performance, it can be seen from Table 5 that the average time costs $Ave_T$ of the I-Q-BSAS algorithm are slightly higher than those of the original BSAS algorithm in all encounters except the large angle crossing encounter. In addition, the mean value of the average computation time cost of the I-Q-BSAS increases 2.45% than the original BSAS, which indicates that the improvement in I-Q-BSAS has

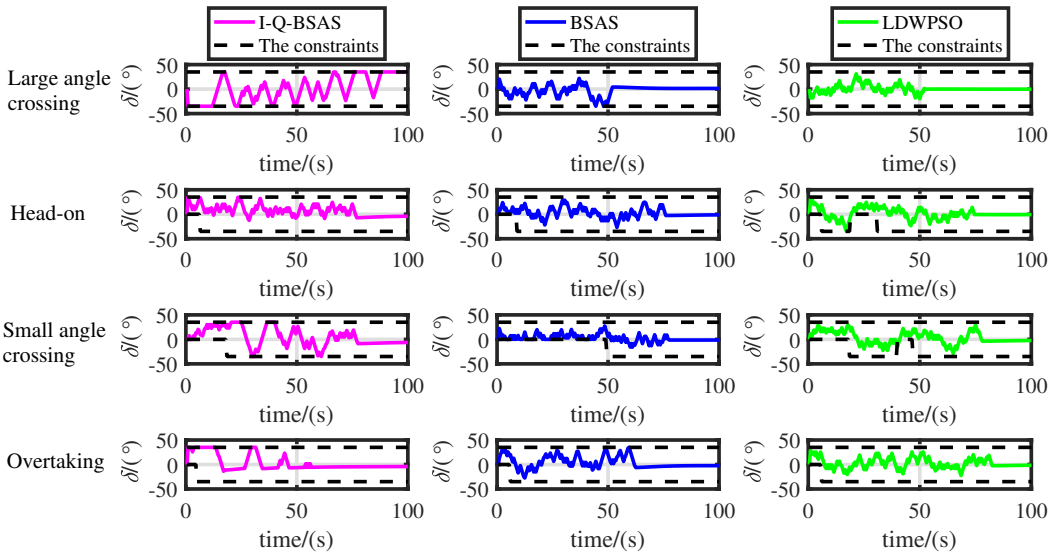


Figure 13: The control constraints during collision avoidance in typical encounters.

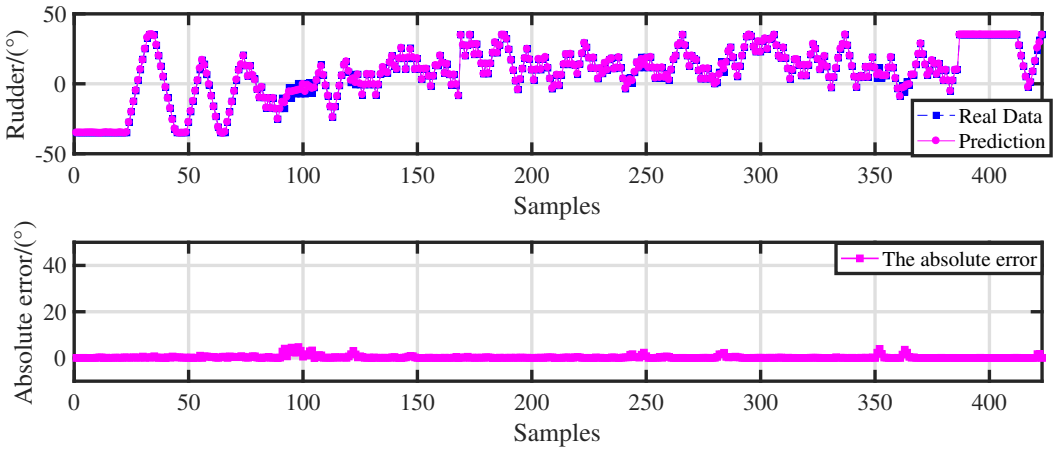


Figure 14: Inverse model prediction.

a certain influence on its real-time performance.

In summary, the proposed I-Q-BSAS algorithm can obtain better optimization and collision avoidance results than the original BSAS and LDWPSO with the same level of time cost and state constraints, which performs better in ship predictive collision avoidance.

### *5.1.2. Verification of the inverse model*

The obtained simulation results based on the proposed I-Q-BSAS method in typical encounters (set in Table 3) are used to train the inverse model. The real rudder output, the predictions and absolute errors of the trained inverse model are shown in Fig. 14. The mean square error of the predictions is calculated to be 0.55°. After the training, initial states for typical encounters in Table 6 are set to verify the effectiveness of the proposed inverse model, using the I-Q-BSAS-based predictive collision avoidance method in Algorithm 2 for comparisons.

Table 6: Initial states for inverse model validation.

| | Head-on | Crossing | | Over-taking |
|---|---|---|---|---|
| | | Small angle | Large angle | |
| $C_{T0}/°$ | 186 | 210 | 280 | 2 |
| $\theta_{T0}/°$ | 3 | 15 | 75 | 2 |
| $K_v$ | 1 | 1 | 1 | 0.3 |

Ship trajectories of the inverse model and I-Q-BSAS method are shown in Fig. 15. The collision avoidance results and time

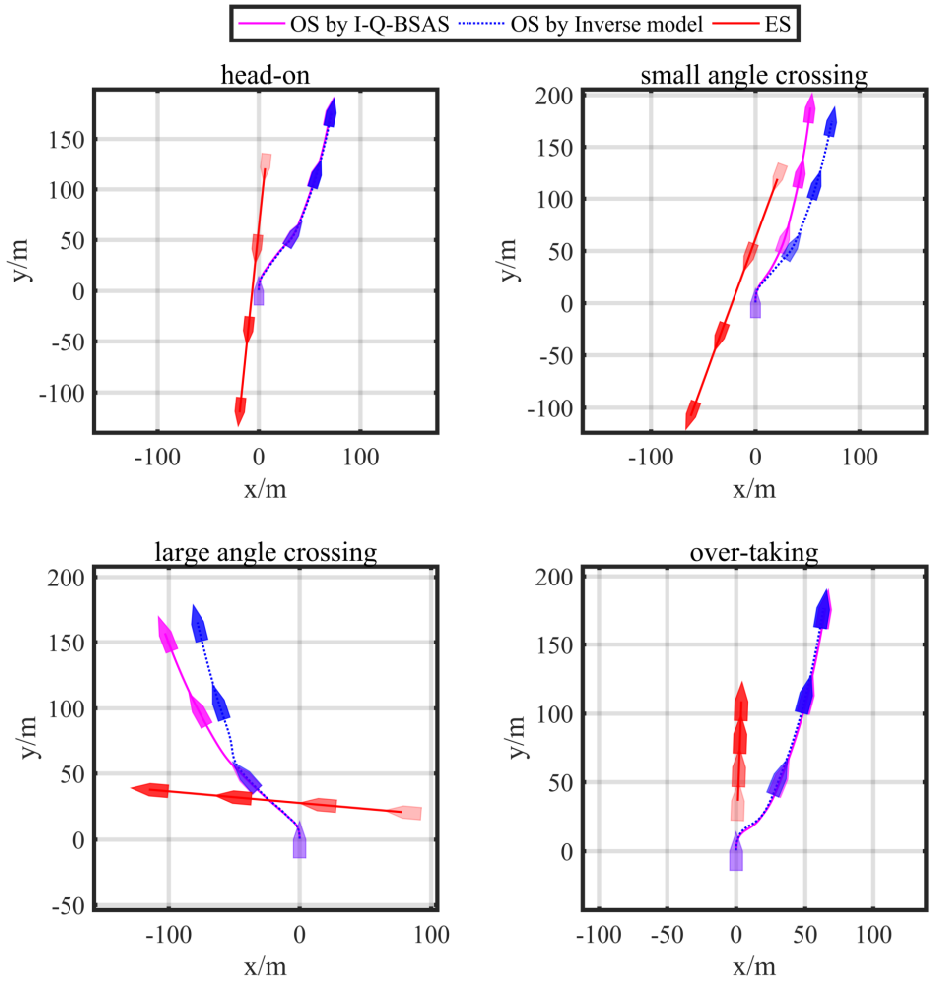


Figure 15: Inverse predictive collision avoidance trajectory.

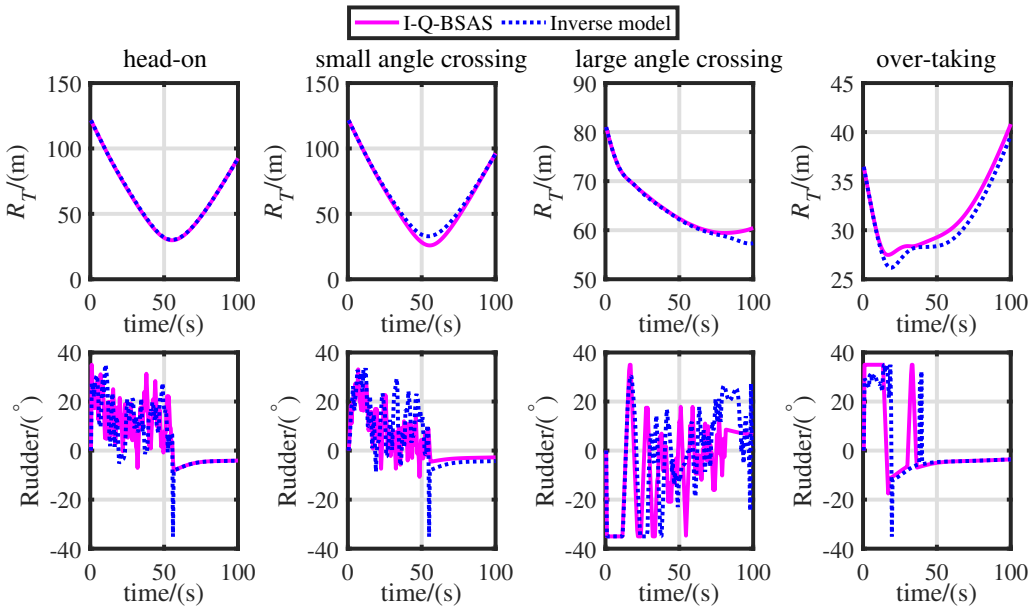


Figure 16: Relative distance of inverse predictive collision avoidance.

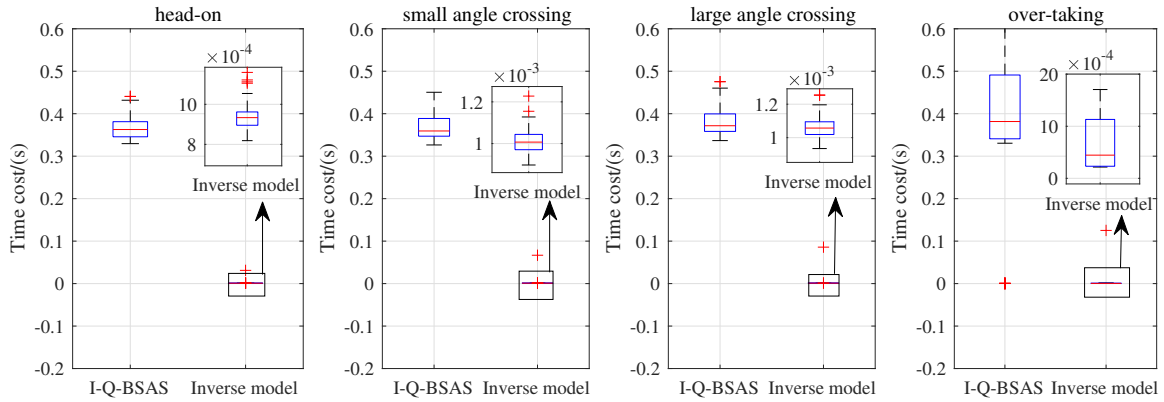


Figure 17: Time cost of inverse predictive collision avoidance.

costs are both compared since the advantage of the inverse model is that the optimal rudder can be directly output without the optimization process. The control inputs, relative distances are shown in Fig. 16, and the time cost are shown in Fig. 17. The minimum relative distance $Min_d$, the maximum DCPA after avoidance $Max_{DCPA}$, and the average time cost $Ave_T$ are calculated as shown in Table 7.

Table 7: Collision avoidance indicators of inverse method in typical encounters

| | $Min_d$/m | | $Max_{DCPA}$/m | | $Ave_T$/s | |
|---|---|---|---|---|---|---|
| | I-QBSAS | Inverse model | I-QBSAS | Inverse model | I-QBSAS | Inverse model |
| ho[1] | 29.824 | 30.194 | 48.044 | 48.514 | 0.365 | 0.002 |
| sac[1] | 25.854 | 32.868 | 38.635 | 51.936 | 0.373 | 0.003 |
| sac[1] | 59.429 | 57.274 | 64.354 | 62.791 | 0.370 | 0.003 |
| ot[1] | 27.480 | 26.168 | 43.844 | 42.632 | 0.481 | 0.004 |

[1] ho: head-on; sac: small angle crossing; lac: large angle crossing; ot: over-taking

It can be seen from the results in Fig. 15 ~ 16 that, in case of using one set of typical encounter samples (Table 3), it is possible to realize the collision avoidance by the proposed inverse model in another different set of encounters (Table 6). With the direct output of the inverse model, the time cost of collision avoidance can be significantly reduced while a good avoidance result similar to that of I-Q-BSAS can be obtained, which can be seen in Fig. 17 and Table 7.

### 5.2. Collision avoidance in multi-ship encounters

In order to verify the effectiveness of the proposed reinforce inverse method for multi-ship encounters, different encountering ships are set at the same time for collision avoidance. The proposed reinforced inverse method denoted in Algorithm 3 is used for multi-ship collision avoidance experiments. Both the proposed I-Q-BSAS based method and the off-line inverse model are used for comparisons to verify the effectiveness of neural network approximation in the inverse model and long-term learning in the reinforced inverse method. In I-Q-BSAS based method, the rudders with respect to different encountering ships are optimized by I-Q-BSAS and weighted based on (38) to generate the final rudder action. To simplify the illustration, RI and IM are used to represent the proposed reinforced inverse method and inverse model, respectively.

#### 5.2.1. Collision avoidance without ship dynamics perturbation

In the comparative test, the time step for retraining the optimal policy is set as $N_{re} = 5s$. Four different encountering ships are set at the same time to form the multi-ship encounter. Since the optimization process is replaced with the output of the inverse model, the fitness in multi-ship encounter is calculated as:

$$\begin{aligned} J_{multi}(t) =& \mu_1 \cdot \max\left\{f_{CRI_1}(t), f_{CRI_2}(t), \cdots, f_{CRI_n}(t)\right\} \\ &+ \mu_2 \cdot |\Delta\delta(t)|, \end{aligned} \tag{40}$$

where $J_{multi}(t)$ is the fitness in multi-ship encounter at time $t$; $n$ is the number of the encountering ship and $f_{CRI_i}(t)$ is the collision risk between the own ship and the $i$th encountering ship; $\mu_1$ and $\mu_2$ are set the same as in (16). Besides, the minimum rela-

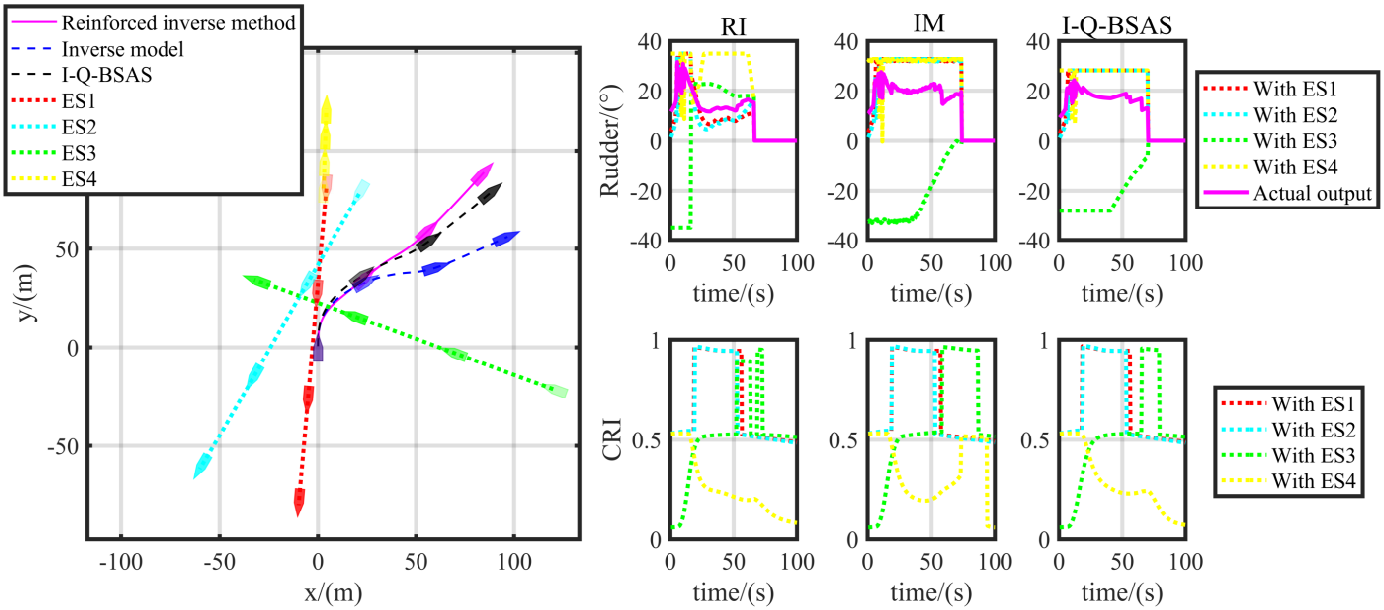


Figure 18: Collision avoidance results without ship dynamics perturbation.

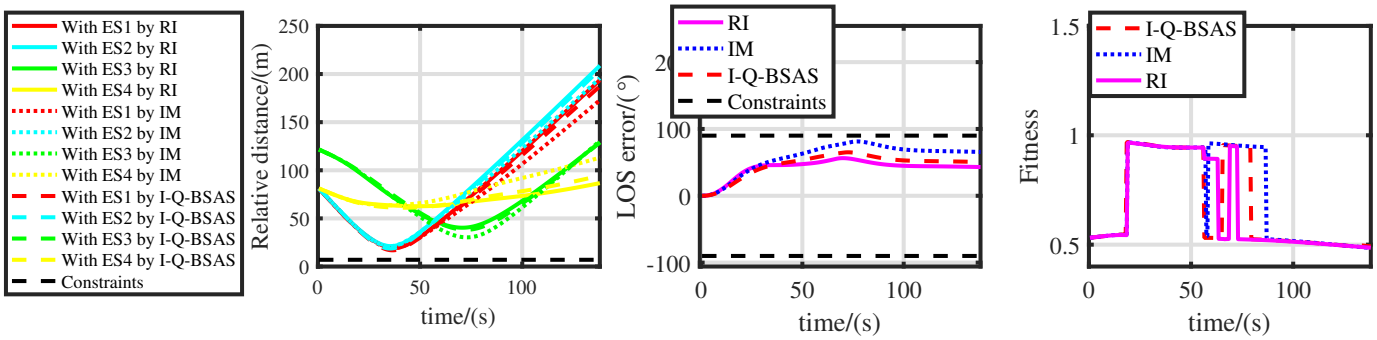


Figure 19: Relative distances, LOS tracking errors and fitness results without ship dynamics perturbation.

tive distance between the own ship and each encountering ship is taken as the evaluation index of actual collision avoidance performance.

The ship trajectories, the collision risks, the rudder inputs for each encountering ship, and the final rudder values are shown in Fig. 18. To fully compare the network based methods (i.e., RI and IM) and the I-Q-BSAS method, the relative distances between the own ship and each encountering ship, the LOS tracking errors and the final fitness results are shown in Fig. 19, as well as the state constraints. The time cost of I-Q-BSAS, the inverse model and reinforced method are shown in Fig. 20. The final indicators, i.e., the minimum distances, mean fitness values and mean time cost are calculated as shown in Table 8.

Table 8: Collision avoidance indicators in multi-ship encounters without parameter perturbation.

| | $R_{T_{\min}}$/(m) | | | | $J_{multi_{mean}}$/(-) | $T_{mean}$/(s) |
|---|---|---|---|---|---|---|
| | ES1 | ES2 | ES3 | ES4 | | |
| I-Q-BSAS | 16.86 | 18.73 | 38.16 | 61.10 | 0.681 | 0.450 2 |
| IM[1] | 19.16 | 21.10 | 30.32 | 63.65 | 0.727 | 0.001 1 |
| RI[1] | 19.11 | 21.19 | 40.34 | 62.67 | 0.665 | 0.007 6 |

[1] IM: the inverse model; RI: the reinforced inverse method

It can be seen from Fig. 18 and Fig. 19 that the own ship can realize effective collision avoidance based on the reinforced inverse method, the inverse model and the I-Q-BSAS method. At the beginning of the collision avoidance, the CRIs with ship 1 and 2 are both higher than those with ship 3 and 4. Correspond-

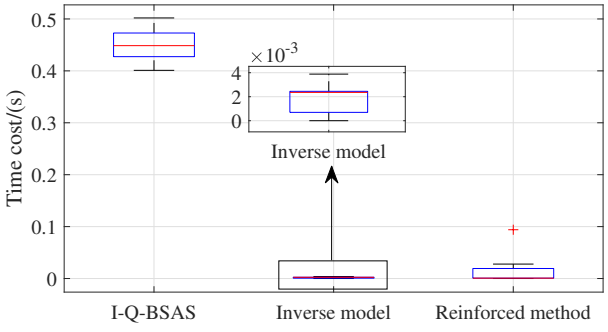


Figure 20: Time cost without ship dynamics perturbation.

ingly, the final rudder outputs have the highest fitting degree with the outputs for ship 1 and 2. After that, the CRIs with ship 1 and 2 are reduced and the CRI with ship 3 is increased and becomes even higher than the CRIs with ship 1 and 2. Therefore, the final rudder value locates between the outputs for ship 1, 2 and ship 3 until the collision avoidance is finished.

Compared with the inverse model, the outputs of the reinforced method are still different from those in the inverse model since the optimal policy is re-trained in the reinforced method. It can be seen from Fig. 18 that the CRI with ship 3 of the reinforced method is reduced faster than that of the inverse model, which results in larger distance with ship 3 and better final optimal fitness of the reinforced method as shown in Fig. 19 and Table 8. Note that the re-training process in the reinforced method also results in larger time cost than the direct inverse model, which can be seen in Fig. 20 Table 8.

Compared with the proposed I-Q-BSAS method, it can be seen from Fig. 20 and Table 8 that both the reinforced method and inverse model have much lower time costs than I-Q-BSAS since the optimization processes are approximated by the neural networks. Without ship dynamics perturbation, the final relative distance results and fitness results of the reinforced method and I-Q-BSAS method are very similar, which can be seen in Table 8. The minimum distances and LOS tracking errors also satisfy the constraints, which can be seen in Fig. 19.

Therefore, it can be concluded that the reinforced method performs better than the inverse model and the proposed I-Q-BSAS method in collision avoidance without ship dynamics perturbation.

### 5.2.2. *Collision avoidance with ship dynamics perturbation*

Several dynamic parameters of the own ship are perturbed to a certain extent at the beginning of multi-ship collision avoidance. In view of the collision avoidance in this study is mainly based on steering, two coefficients related to the rudder angle in the sway and yaw directions, i.e., $Y_\delta$ and $N_\delta$ in (3) are increased by 50%. Then the final ship trajectories, rudder results and CRIs are shown in Fig. 21, the relative distances, the LOS tracking errors and fitness results are shown in Fig. 22. The time costs are shown in Fig. 23. The minimum relative distances, the mean fitness and time cost results are calculated in Table 9.

Table 9: Collision avoidance indicators in multi-ship encounters with parameter perturbation.

| | $R_{T_{\min}}$/(m) | | | | | |
|---|---|---|---|---|---|---|
| | ES1 | ES2 | ES3 | ES4 | $J_{multi_{mean}}$/(-) | $T_{mean}$/(s) |
| I-Q-BSAS | 19.99 | 21.87 | 27.99 | 64.11 | 0.733 | 0.451 3 |
| IM[1] | 27.83 | 30.01 | 0.68 | 69.05 | 0.753 | 0.001 5 |
| RI[1] | 26.68 | 29.23 | 30.16 | 69.11 | 0.715 | 0.012 7 |

[1] IM: the inverse model; RI: the reinforced inverse method

It can be seen from Fig. 21 that, after the perturbation of ship dynamics, the rudder outputs by the reinforced method are still similar to those in Fig. 18, while the inverse model without re-training is failed to reduce the CRI with encountering ship 3 effectively, which results in a minimum distance close to zero with ship 3 as shown in Table 9, which is infeasible under the constraints shown in Fig. 22.

Compared with the inverse model, the proposed reinforced inverse method obtains an adaptive optimal policy which is more suitable for the current ship model and avoids collision with encountering ship 3. It can be seen from Fig. 22 and Table 9 that the reinforced method obtains obviously larger minimum distance with ES3 than the inverse model while the minimum distances with other encountering ships are at the same level. Thus, the reinforced method also obtains a better mean fitness value than the inverse model, which can be seen in Table 9. Besides, the time costs of the reinforced method are larger than those of the direct inverse model, which is similar to the results without parameter perturbation.

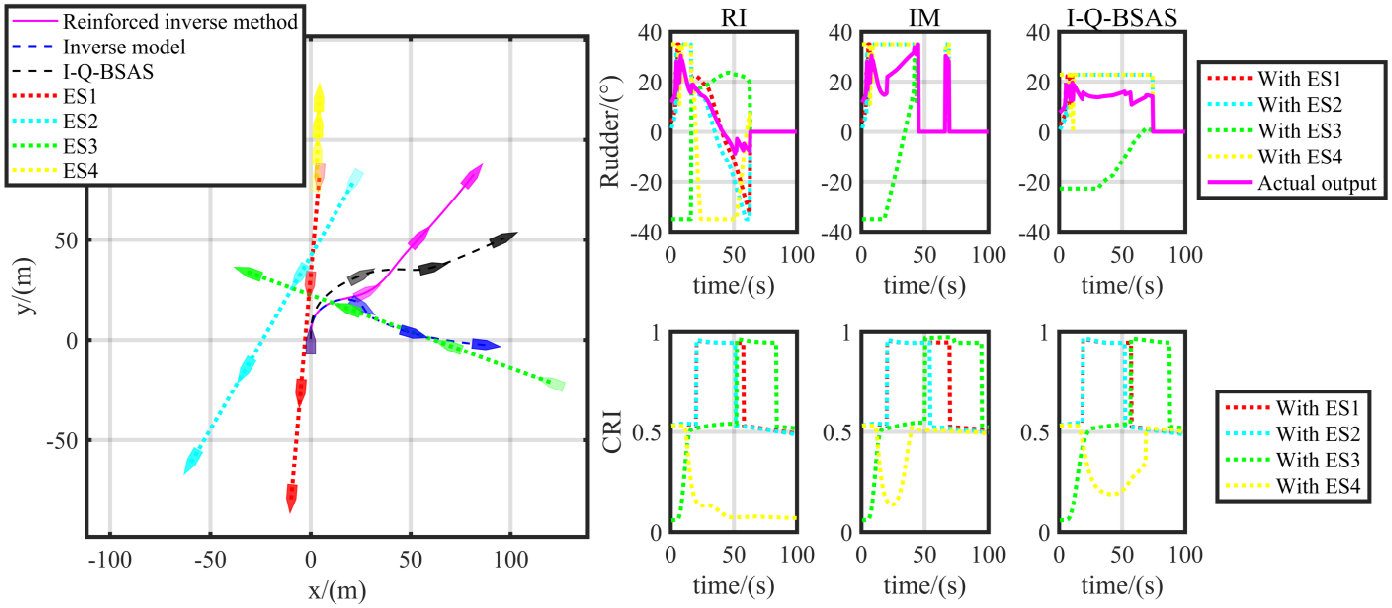


Figure 21: Collision avoidance results with ship dynamics perturbation.

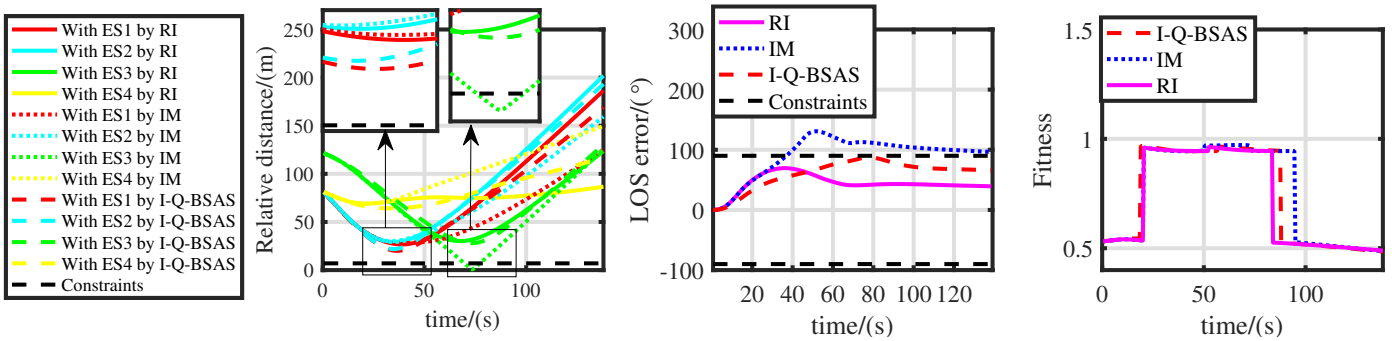


Figure 22: Relative distance, LOS tracking errors and fitness results with ship dynamics perturbation.

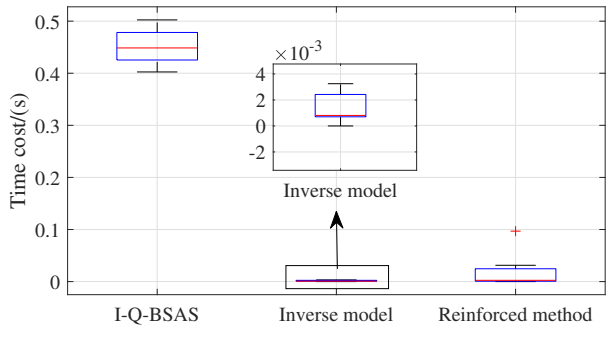


Figure 23: Time cost with ship dynamics perturbation.

Compared with the proposed I-Q-BSAS method, it can be seen from Fig. 21 and Fig. 22 that the relative distances and LOS tracking errors of both the reinforced method and I-Q-BSAS satisfy the constraints since the state constraints are still considered in the re-training process, while those of the inverse model violates the constraints. In spite of this, the reinforced method still obtains larger minimum distances and better mean fitness results than the I-Q-BSAS, which can be seen in Table 9. Moreover, the reinforced method also has much fewer time costs than the I-Q-BSAS approach, which is similar to the results without ship dynamics perturbation.

Generally speaking, both the proposed I-Q-BSAS and the inverse model can achieve multi-ship collision avoidance under the known ship dynamics. Benefiting from the function approx-

imation and re-training with on-policy data, the proposed reinforced inverse method can reduce the time cost of I-Q-BSAS significantly, and obtain better collision avoidance performance than the direct inverse model when the ship model dynamics are perturbed or unknown.

## 6. Conclusions and future research

In view of the real-time ship collision avoidance problem, a reinforced inverse method for predictive collision avoidance is proposed by combining MPC, an improved Q-BSAS (I-Q-BSAS) algorithm and a neural network. MPC is applied for establishing the predictive collision avoidance strategy, and the proposed I-Q-BSAS algorithm is used for solving the MPC optimization problem by combining an improved BSAS and Q-learning. The neural network is used to learn an inverse model that is used to approximate the optimal policy in MPC for real-time collision avoidance. In addition, the inverse model is reinforced through long-term retraining with aggregated on-policy and off-policy data. The following conclusions are drawn from the simulation experiments using the model of KVLCC2:

(1) In terms of collision avoidance in typical encounters, the proposed I-Q-BSAS can obtain smaller optimal fitness valves and better collision avoidance performance than the original BSAS and LDWPSO with the same level of time cost. Moreover, the proposed inverse model can significantly reduce the time cost of the proposed I-Q-BSAS with the approximated performance.

(2) In terms of collision avoidance in multi-ship encounters, the proposed reinforced method still has lower time cost than the proposed I-Q-BSAS and performs better than the direct inverse model with ship dynamics perturbation, which verifies the effectiveness of the neural network approximation and the long-term policy retraining, respectively.

Future works can be carried out on the following aspects:

1) Note that only one ship is considered for collision avoidance in this study, the distributed model predictive control frame work (Zheng et al., 2017b, 2016, 2017a) could be considered for multi-ship collision avoidance if all the ships are controlled.

2) In this study, the weighting approach is used to consider both safety and economy in ship collision avoidance. This requires repeated adjustments. A multi-objective Q-BSAS algorithm can be considered for more reasonable collision avoidance results as part of future research.

3) The uncertain environment (i.e., unknown static obstacles and encountering ship states) can be considered, and the obtained optimal policy will be combined with several model-free reinforcement learning methods to deal with the uncertain environment.

## Acknowledgements

This research is supported by the China Scholarship Council under Grant (201806950097), the Fundamental Research Funds for the Central Universities (No.2019-YB-022), the High Technology Ship Project of Ministry of Industry and Information Technology (No.2016050001), the Key Project of Science and Technology of Wuhan (201701021010132), the Joint WUT (Wuhan University of Technology) -TUDelft (Delft University of Technology) Cooperation, the ResearchLab Autonomous Shipping Delft, the Netherlands, and financially supported by the Double First-rate Project of WUT.

## References

Abdelaal, M., Frnzle, M., Hahn, A., 2016. NMPC-based trajectory tracking and collison avoidance of underactuated vessels with elliptical ship domain. IFAC Papers online 49 (23), 22–27.

Abdelaal, M., Frnzle, M., Hahn, A., 2018. Nonlinear model predictive control for trajectory tracking and collision avoidance of underactuated vessels with disturbances. Ocean Engineering 160, 168 – 180.

Ahn, J. H., Rhee, K. P., You, Y. J., 2012. A study on the collision avoidance of a ship using neural networks and fuzzy logic. Applied Ocean Research 37 (4), 162–173.

Bououden, S., Chadli, M., Karimi, H., 2015. An ant colony optimization-based fuzzy predictive control approach for nonlinear processes. Information Sciences 299, 143 – 158.

Chen, D., Dai, C., Wan, X., Mou, J., 2015. A research on AIS-based embedded system for ship collision avoidance. In: Proceedings of the 2015 International Conference on Transportation Information and Safety (ICTIS). pp. 512–517.

Chen, L., Du, S., He, Y., Liang, M., Xu, D., 2018a. Robust model predictive control for greenhouse temperature based on particle swarm optimization. Information Processing in Agriculture 5 (3), 329 – 338.

Chen, L., Hopman, J. J., Negenborn, R. R., 2018b. Distributed model predictive control for vessel train formations of cooperative multi-vessel systems. Transportation Research Part C: Emerging Technologies 92, 101 – 118.

Chen, L., Huang, L., 2012a. Ship collision avoidance path planning by pso based on maneuvering equation. Future Wireless Networks and Information Systems, 675–682.

Chen, L., Huang, L., 2012b. Ship collision avoidance path planning by PSO based on maneuvering equation. In: Future Wireless Networks and Information Systems. Springer, pp. 675–682.

Chen, L., Negenborn, R. R., Lodewijks, G., 2016. Path planning for autonomous inland vessels using A* BG. In: Proceedings of the 2016 International Conference on Computational Logistics. Springer, pp. 65–79.

Chen, T., Zhu, Y., Teng, J., 2018c. Beetle swarm optimisation for solving investment portfolio problems. The Journal of Engineering 2018 (16), 1600–1605.

Cheng, L., Liu, C., Yan, B., 2014. Improved hierarchical A-star algorithm for optimal parking path planning of the large parking lot. In: 2014 IEEE International Conference on Information and Automation (ICIA). IEEE, pp. 695–698.

Dai, L., Cao, Q., Xia, Y., Gao, Y., 2017. Distributed mpc for formation of multi-agent systems with collision avoidance and obstacle avoidance. Journal of the Franklin Institute 354 (4), 2068–2085.

Davis, P., Dove, M., Stockel, C., 1982. A computer simulation of multi-ship encounters. Journal of Navigation 35 (2), 347–352.

Dirion, J., Cabassud, M., Le Lann, M., Casamatta, G., 1995. Design of a neural controller by inverse modelling. Computers & Chemical Engineering 19, 797–802.

Finn, C., Levine, S., Abbeel, P., 2016. Guided cost learning: Deep inverse optimal control via policy optimization. In: Proceedings of the 2016 International Conference on Machine Learning. pp. 49–58.

Hara, K., Hammer, A., 1993. A safe way of collision avoidance manoeuvre based on manoeuvring standard using fuzzy reasoning model. Fuzzy Systems 1, 163–171.

He, W., Chen, Y., Yin, Z., 2016. Adaptive neural network control of an uncertain robot with full-state constraints. IEEE Transactions on Cybernetics 46 (3), 620–629.

Inaishi, M., Matsumura, H., Imazu, H., Sugisaki, A. M., 1992. Basic research on a collision avoidance system using neural networks. Navigation 1 (112), 22–27.

Jamil, M., Yang, X. S., 2013. A literature survey of benchmark functions for global optimization problems. International Journal of Mathematical Modelling & Numerical Optimisation 4 (2), 150–194.

Jiang, X., Li, S., 2017. Beetle antennae search without parameter tuning (BAS-WPT) for multi-objective optimization. arXiv preprint arXiv:1807.10470.

Jiang, X., Li, S., 2018. BAS: Beetle antennae search algorithm for optimization problems. International Journal of Robotics and Control 1 (1), 1–5.

Kadirkamanathan, V., Selvarajah, K., Fleming, P. J., 2006. Stability analysis of the particle dynamics in particle swarm optimizer. IEEE Transactions on Evolutionary Computation 10 (3), 245–255.

Kennedy, J., Eberhart, R., 1995. Particle swarm optimization (pso). In: Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia. pp. 1942–1948.

Kim, M.-H., Heo, J.-H., Wei, Y., Lee, M.-C., 2011. A path planning algorithm using artificial potential field based on probability map. In: 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI). IEEE, pp. 41–43.

Lazarowska, A., 2015. Ship's trajectory planning for collision avoidance at sea based on ant colony optimisation. Journal of Navigation 68 (2), 291–307.

Lazarowska, A., 2018. A new potential field inspired path planning algorithm for ships. In: 2018 23rd International Conference on Methods & Models in Automation & Robotics (MMAR). IEEE, pp. 166–170.

Li, S., Liu, J., Negenborn, R. R., 2019. Distributed coordination for collision avoidance of multiple ships considering ship maneuverability. Ocean Engineering 181, 212–226.

Li, X., Bo, N., Dai, J., 2017. Study on collision avoidance path planning for multi-UAVs based on model predictive control. Journal of Northwestern Polytechnical University 35 (3), 513–522.

Lin, X., Liu, Y., Wang, Y., 2018. Design and research of dc motor speed control system based on improved BAS. In: 2018 Chinese Automation Congress (CAC). IEEE, pp. 3701–3705.

Liu, C., Negenborn, R. R., Chu, X., Zheng, H., 2018. Predictive path following based on adaptive line-of-sight for underactuated autonomous surface vessels. Journal of Marine Science and Technology 23 (3), 483–494.

Liu, L., He, D., Ying, M., Li, T., Li, J., 2017. Research on ships collision avoidance based on chaotic particle swarm optimization. In: Proceedings of the International Conference on Smart Vehicular Technology. pp. 230–239.

Luo, W., Li, X., 2017. Measures to diminish the parameter drift in the modeling of ship manoeuvring using system identification. Applied Ocean Research 67, 9 – 20.

Lyu, H., Yin, Y., 2018. Fast path planning for autonomous ships in restricted waters. Applied Sciences 8 (12), 2592–2616.

Ma, Y., Gan, L., Zheng, Y., Zhang, J., 2014. Autonomous ship safe navigation using smoothing A* algorithm. Open Cybernetics & Systemics Journal 8 (1), 72–76.

Ma, Y., Hu, M., Yan, X., 2018. Multi-objective path planning for unmanned surface vehicle with currents effects. ISA Transactions 75, 137 – 156.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature

518 (7540), 529–533.

Mu, Y., Li, B., An, D., Wei, Y., 2019. Three-dimensional route planning based on the beetle swarm optimization algorithm. IEEE Access 7, 117804–117813.

Naeem, W., Henrique, S. C., Liang, H., 2016. A reactive colregs-compliant navigation strategy for autonomous maritime navigation. IFAC Papers online 49 (23), 207–213.

Naeem, W., Irwin, G. W., Yang, A., 2012. Colregs-based collision avoidance strategies for unmanned surface vehicles. Mechatronics 22 (6), 669–678.

Nagabandi, A., Kahn, G., Fearing, R. S., Levine, S., 2018. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. In: Proceedings of the IEEE 2018 International Conference on Robotics and Automation (ICRA). pp. 7559–7566.

Negenborn, R. R., De Schutter, B., Wiering, M. A., Hellendoorn, H., 2005. Learning-based model predictive control for Markov decision processes. IFAC Proceedings Volumes 38 (1), 354 – 359, 16th IFAC World Congress.

Negenborn, R. R., Maestre, J. M., 2014. Distributed model predictive control: An overview and roadmap of future research opportunities. IEEE Control Systems Magazine 34 (4), 87–97.

Ou, L.-l., Zhang, W.-d., Yu, L., 2009. Low-order stabilization of lti systems with time delay. IEEE Transactions on Automatic Control 54 (4), 774–787.

Perera, L., Carvalho, J., Soares, C. G., 2011. Fuzzy logic based decision making system for collision avoidance of ocean navigation under critical collision conditions. Journal of Marine Science and Technology 16 (1), 84–99.

Perera, L. P., Carvalho, J. P., Guedes, C., 2012. Intelligent ocean navigation and fuzzy-bayesian decision/action formulation. IEEE Journal of Oceanic Engineering 37 (2), 204–219.

Perera, L. P., Ferrari, V., Santos, F. P., Hinostroza, M. A., Soares, C. G., 2015. Experimental evaluations on ship autonomous navigation & collision avoidance by intelligent guidance. IEEE Journal of Oceanic Engineering 40 (2), 374–387.

Perizzato, A., Farina, M., Scattolini, R., 2015. Formation control and collision avoidance of unicycle robots with distributed predictive control. IFAC Papers online 48 (23), 260–265.

Samal, N. R., Konar, A., Das, S., Abraham, A., 2007. A closed loop stability analysis and parameter selection of the particle swarm optimization dynamics for faster convergence. In: Proceedings of the 2007 IEEE Congress on Evolutionary Computation. IEEE, pp. 1769–1776.

Samma, H., Lim, C. P., Saleh, J. M., 2016. A new reinforcement learning-based memetic particle swarm optimizer. Applied Soft Computing 43 (C), 276–297.

Simsir, U., Bal, M., Ertugrul, S., 2014. Decision support system for collision avoidance of vessels. Applied Soft Computing Journal 25 (C), 369–378.

Solis, F. J., Wets, R. J.-B., 1981. Minimization by random search techniques. Mathematics of operations research 6 (1), 19–30.

Son, N. N., Van Kien, C., Anh, H. P. H., 2017. A novel adaptive feed-forward-pid controller of a scara parallel robot using pneumatic artificial muscle actuator based on neural network and modified differential evolution algorithm. Robotics and Autonomous Systems 96, 65–80.

Song, Y., Chen, Z., Yuan, Z., 2007. New chaotic pso-based neural network predictive control for nonlinear process. IEEE transactions on neural networks 18 (2), 595–601.

Sun, Y., Zhang, J., Li, G., Wang, Y., Sun, J., Jiang, C., 2019. Optimized neural network using beetle antennae search for predicting the unconfined compressive strength of jet grouting coalcretes. International Journal for Numerical and Analytical Methods in Geomechanics 43 (4), 801–813.

Szlapczynski, R., Szlapczynska, J., 2017. Review of ship safety domains: Models and applications. Ocean Engineering 145, 277–289.

Taherkhani, M., Safabakhsh, R., 2016. A novel stability-based adaptive inertia weight for particle swarm optimization. Applied Soft Computing 38, 281–295.

Tsou, M.-C., Hsueh, C.-K., 2010. The study of ship collision avoidance route planning by ant colony algorithm. Journal of marine science and technology 18 (5), 746–756.

Tsou, M.-C., Kao, S.-L., Su, C.-M., 2010. Decision support from genetic algorithms for ship collision avoidance route planning and alerts. The Journal of Navigation 63 (1), 167–182.

Wang, H., Zhou, J., Zheng, G., Liang, Y., 2014. HAS: Hierarchical A-star algorithm for big map navigation in special areas. In: 2014 5th International Conference on Digital Home. IEEE, pp. 222–225.

Wang, J., Chen, H., 2018. BSAS: Beetle swarm antennae search algorithm for optimization problems. arXiv preprint arXiv:1807.10470.

Wang, T., Yang, L., Liu, Q., 2018a. Beetle swarm optimization algorithm: Theory and application. arXiv preprint arXiv:1808.00206.

Wang, Y., Wang, X., Sun, Y., You, S., 2018b. Model predictive control strategy for energy optimization of series-parallel hybrid electric vehicle. Journal of Cleaner Production 199, 348 – 358.

Xin, J., Chen, G., Hai, Y., 2009. A particle swarm optimizer with multi-stage linearly-decreasing inertia weight. In: 2009 International Joint Conference on Computational Sciences and Optimization. Vol. 1. IEEE, pp. 505–508.

Xue, Y., Clelland, D., Lee, B. S., Han, D., 2011. Automatic simulation of ship navigation. Ocean Engineering 38 (17), 2290–2305.

Xue, Y. Z., Wei, Y., Qiao, Y., 2012. The research on ship intelligence navigation in confined waters. Advanced Materials Research 442, 398–401.

Yan, Q., 2002. A model for estimating the risk degrees of collisions. Journal of Wuhan University of Technology 26 (2), 77–79.

Yasukawa, H., Yoshimura, Y., 2015. Introduction of mmg standard method for ship maneuvering predictions. Journal of Marine Science and Technology 20 (1), 37–52.

Yin, C., Zhang, W., 2018. Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels. Neurocomputing 272.

Zhang, G., Cai, Y., Zhang, W., 2016. Robust neural control for dynamic positioning ships with the optimum-seeking guidance. IEEE Transactions on Systems, Man, and Cybernetics: Systems 47 (7), 1500–1509.

Zhang, G., Zhang, X., 2017. Practical robust neural path following control for underactuated marine vessels with actuators uncertainties. Asian Journal of

Control 19 (1), 173–187.

Zhang, W., Gu, D., Wang, W., Xu, X., 2004. Quantitative performance design of a modified smith predictor for unstable processes with time delay. Industrial & engineering chemistry research 43 (1), 56–62.

Zheng, H., Negenborn, R. R., Lodewijks, G., 2016. Predictive path following with arrival time awareness for waterborne AGVs. Transportation Research Part C: Emerging Technologies 70, 214–237.

Zheng, H., Negenborn, R. R., Lodewijks, G., 2017a. Closed-loop scheduling and control of waterborne AGVs for energy-efficient inter terminal transport. Transportation Research Part E: Logistics and Transportation Review 105, 261–278.

Zheng, H., Negenborn, R. R., Lodewijks, G., 2017b. Robust distributed predictive control of waterborne AGVs-A cooperative and cost-effective approach. IEEE transactions on cybernetics 48 (8), 2449–2461.

Zheng, Z., Wu, Z., 2000. Decision-making of vessel collision avoidance. Publishing Company of Dalian Maritime University.

Zhu, Z., Zhang, Z., Man, W., Tong, X., Qiu, J., Li, F., 2018. A new beetle antennae search algorithm for multi-objective energy management in microgrid. In: Proceedings of the 2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA). pp. 1599–1603.

## Appendices

## A. Convergence and global optimality analysis of the improved BSAS

In this section, convergence and global optimality analysis of the improved BSAS denoted in Section 3.3.2 are given.

### *A.1. Convergence analysis*

The improved BSAS algorithm uses random directions of the beetle antennas for optimization, which is a kind of stochastic optimization algorithm. Linear time-invariant discrete system stability analysis is a widely used method for convergence analysis of stochastic optimization algorithms (Samal et al., 2007; Kadirkamanathan et al., 2006). Since $J_{il\text{best}}$ and $J_{ir\text{best}}$ are updated at every iteration and remain the same between two adjacent iterations, which can be regarded as discrete variables changing over iterations as a dynamic system:

$$\begin{aligned} \varphi_1(k) &= \frac{\|J_{ir\text{best}}\|}{\|J_{il\text{best}}\| + \|J_{ir\text{best}}\|}, \\ 1-\varphi_1(k) &= \frac{\|J_{il\text{best}}\|}{\|J_{il\text{best}}\| + \|J_{ir\text{best}}\|}, \end{aligned} \tag{41}$$

where $k$ is the iteration number.

Then the following equation is obtained by combining (24) and (41):

$$\begin{aligned} \hat{\boldsymbol{\delta}}_{i_k} =& \hat{\boldsymbol{\delta}}_{i_{k-1}} + cr_d\varphi_1(k)\left(\hat{\boldsymbol{\delta}}_{il\text{best}} - \hat{\boldsymbol{\delta}}_{i_{k-1}}\right) \\ &+ cr_d(1-\varphi_1(k))\left(\hat{\boldsymbol{\delta}}_{ir\text{best}} - \hat{\boldsymbol{\delta}}_{i_{k-1}}\right), \end{aligned} \tag{42}$$

Furthermore, another discrete variable is defined as:

$$\varphi_2(k) = cr_d\varphi_1(k)\,\hat{\boldsymbol{\delta}}_{il\text{best}} + cr_d(1-\varphi_1(k))\,\hat{\boldsymbol{\delta}}_{il\text{best}} \tag{43}$$

Based on (42) and (43), the update strategy of the centroid can be simplified to a typical linear time-invariant discrete system as:

$$\hat{\boldsymbol{\delta}}_{i_k} = (1-cr_d)\,\hat{\boldsymbol{\delta}}_{i_{k-1}} + \varphi_2(k), \tag{44}$$

Then, the necessary and sufficient condition for the convergence of the system denoted by (24) is that the only eigenvalue of the coefficient matrix $(1-cr_d)$ satisfies the following condition:

$$|1-cr_d| < 1 \Leftrightarrow 0 < cr_d < 2. \tag{45}$$

Thus, the sufficient condition for convergence of the centroid position is $0 < c < 2$.

### *A.2. Global optimality analysis*

Define $\boldsymbol{S}$ as the searching space of the improved BSAS and $\mu[\cdot]$ as the probability measure. Assuming that the theoretical global optimum of the problem to be solved is $\boldsymbol{\xi}^*$, which has a probability in any subset $\forall A \in \boldsymbol{S}$. The global optimality of the proposed improved BSAS algorithm can be guaranteed by the following condition since the global fitness $J_g$ is monotonous and non-increasing (Solis and Wets, 1981):

$$\mu\{\hat{\boldsymbol{\delta}}_g = \boldsymbol{\xi}^* | \forall \boldsymbol{\xi}^* \in \boldsymbol{S}\} > 0, \tag{46}$$

Jiang and Li (2017); Wang and Chen (2018) have proposed that the initialization of the step $l^0$ of the beetle is very important for

optimization. In Jiang and Li (2017), the $l^0$ is given as:

$$l^0 = 2\left\|\hat{\boldsymbol{\delta}}_{\max} - \hat{\boldsymbol{\delta}}_{\min}\right\|, \tag{47}$$

Where $[\hat{\boldsymbol{\delta}}_{\min}, \hat{\boldsymbol{\delta}}_{\max}] = \boldsymbol{S}$ is the searching space. However, the proof of the global optimality of BSAS algorithm has not been given at present. In this study, we attempt to proof the conditional global optimality of the improved BSAS algorithm as follows:

**Theorem 1**: Assuming that the step $l^0$ of the beetle in the improved BSAS algorithm is initialized as (47), the condition denoted in (46) can be satisfied and the global optimality of the improved BSAS is guaranteed:

**Proof**: Based on (27), the step of the beetle at $k$th iteration $d^k$ satisfies:

$$\mu\left[d^k = d|\forall d \in \left[d_{\min}, 2\left\|\hat{\delta}_{\max} - \hat{\delta}_{\min}\right\| + d_{\min}\right]\right] > 0, \tag{48}$$

It can be seen from (28) that $d^k$ affects the exploring of two antennas directly. The probability of $\{\hat{\boldsymbol{\delta}}_{il}^k = \boldsymbol{\xi}^*\}$ is:

$$\begin{aligned}\mu\left[\hat{\boldsymbol{\delta}}_{il}^k = \boldsymbol{\xi}^*\right] &= \mu\left[\hat{\boldsymbol{\delta}}_{il\text{best}}^k + \frac{1}{2}d^k\frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} = \boldsymbol{\xi}^*\right] \\ &= \mu\left[\frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} = \frac{2\left(\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right)}{d^k}\right],\end{aligned} \tag{49}$$

Based on the law of total probability, we get:

$$\begin{aligned}&\mu\left[\frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} = \frac{2\left(\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right)}{d^k}\right] \\ \geq&\mu\left[\frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} = \frac{2\left(\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right)}{d^k}|\left\{d^k > 2\left\|\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right\|\right\}\right] \\ &\cdot\mu\left[d^k > 2\left\|\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right\|\right],\end{aligned} \tag{50}$$

where $\|\cdot\|$ represents the 2-norm. Since $\frac{\boldsymbol{d}}{\|\boldsymbol{d}\|}$ is a random vector with an uniform distribution between $[-1, 1]$, the following equation is obtained:

$$\mu\left[\frac{\boldsymbol{d}}{\|\boldsymbol{d}\|} = \frac{2\left(\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right)}{d^k}|\left\{d^k \geq 2\left\|\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right\|\right\}\right] > 0, \tag{51}$$

In addition, for $\forall\tilde{d} \in \left(2\left\|\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right\|, 2\left\|\hat{\delta}_{\max} - \hat{\delta}_{\min}\right\| + d_{\min}\right)$, we have the following equation based on (48) and the law of total probability:

$$\begin{aligned}&\mu\left[d^k > 2\left\|\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right\|\right] \\ =&\mu\left[d^k = \tilde{d}\right]\mu\left[\tilde{d} > 2\left\|\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right\|\right] \\ &+\mu\left[d^k \neq \tilde{d}\right]\mu\left[d^k > 2\left\|\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right\||\{d^k \neq \tilde{d}\}\right] \\ \geq&\mu\left[d^k = \tilde{d}\right]\mu\left[\tilde{d} > 2\left\|\boldsymbol{\xi}^* - \hat{\boldsymbol{\delta}}_{il\text{best}}^k\right\|\right] \\ =&\mu\left[d^k = \tilde{d}\right] > 0,\end{aligned} \tag{52}$$

Then, combining (49)~(52), we get:

$$\mu\left[\hat{\boldsymbol{\delta}}_{il}^k = \boldsymbol{\xi}^*|\forall\boldsymbol{\xi}^* \in \boldsymbol{S}\right] > 0, \tag{53}$$

Based on (25) and (53), we get:

$$\begin{aligned}&\mu\left[\hat{\boldsymbol{\delta}}_g = \boldsymbol{\xi}^*|\forall\boldsymbol{\xi}^* \in \boldsymbol{S}\right] \\ \geq&\mu\left[\hat{\boldsymbol{\delta}}_g = \hat{\boldsymbol{\delta}}_{il}^k|\hat{\boldsymbol{\delta}}_{il}^k = \boldsymbol{\xi}^*\right]\mu\left[\hat{\boldsymbol{\delta}}_{il}^k = \boldsymbol{\xi}^*|\forall\boldsymbol{\xi}^* \in \boldsymbol{S}\right] \\ =&\mu\left[\hat{\boldsymbol{\delta}}_{il}^k = \boldsymbol{\xi}^*|\forall\boldsymbol{\xi}^* \in \boldsymbol{S}\right] > 0.\end{aligned} \tag{54}$$

Therefore, the condition in (46) is proofed to be satisfied, the global optimality of the proposed improved BSAS algorithm can be guaranteed with the initial step $l^0$ setting as (47). **Theorem 1** proof finishes.

### B. Comparisons of the proposed I-Q-BSAS, original BSAS and LDWPSO with benchmark functions

As mentioned in Section 3.3.3, to verify the optimization performance of the improved Q-learning beetle swarm antenna search (I-Q-BSAS) algorithm comprehensively, several benchmark tests are conducted in this section. The original beetle swarm antenna search (BSAS) algorithm and linear decreasing weight particle swarm optimization (LDWPSO) algorithm are used for comparisons. 6 benchmark functions with different dimensions (Jamil and Yang, 2013), i.e., 'Ackley', 'Rastrigin', 'Sum Squares', 'Rosenbrock', 'Griewank' and 'Salomon', are selected since the dimension of the benchmarks has influences on the optimization performance.The detailed conditions and equations of the benchmarks are shown in Table 10. The mean

Table 10: Benchmark functions.

| Benchmark functions | Equations | Search area |
|---|---|---|
| Ackley | $f_1(\boldsymbol{x}) = 20 + e - 20\exp\left(-0.2\sqrt{\frac{1}{N}\sum_{n=1}^{N} x_n}\right) - \exp\left(\sqrt{\frac{1}{N}\sum_{n=1}^{N}\cos(2\pi x_n)}\right)$ | $x_n \in \left[\ -10 \quad 10\ \right]$ |
| Rastrigin | $f_3(\boldsymbol{x}) = \sum_{n=1}^{N}\left({x_n}^2 - 10\cos(2\pi x_n) + 10\right)$ | $x_n \in \left[\ -5.12 \quad 5.12\ \right]$ |
| Sum Squares | $f_5(\boldsymbol{x}) = \sum_{n=1}^{N}\left(n{x_n}^2\right)$ | $x_n \in \left[\ -5.12 \quad 5.12\ \right]$ |
| Rosenbrock | $f_4(\boldsymbol{x}) = \sum_{i=1}^{N-1} 100\left(x_{i+1} - x_i^2\right)^2 + (x_i - 1)^2$ | $x_n \in \left[\ -10 \quad 10\ \right]$ |
| Griewank | $f_2(\boldsymbol{x}) = \frac{1}{4000}\sum_{n=1}^{N} {x_n}^2 - \prod_{n=1}^{N}\cos\left(\frac{x_n}{\sqrt{n}}\right) + 1$ | $x_n \in \left[\ -600 \quad 600\ \right]$ |
| Salomon | $f_6(\boldsymbol{x}) = 1 - \cos\left(2\pi\sqrt{\sum_{i=1}^{N} x_i^2}\right) + 0.1\sqrt{\sum_{i=1}^{N} x_i^2}$ | $x_n \in \left[\ -10 \quad 10\ \right]$ |

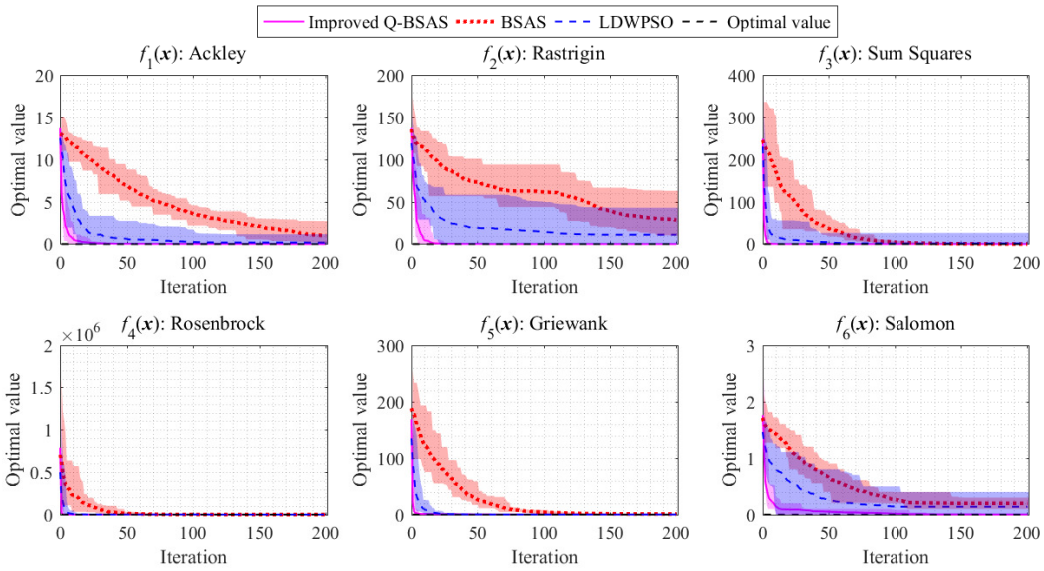


Figure 24: Comparison of the proposed I-Q-BSAS, BSAS and LDWPSO with 10 dimensions.

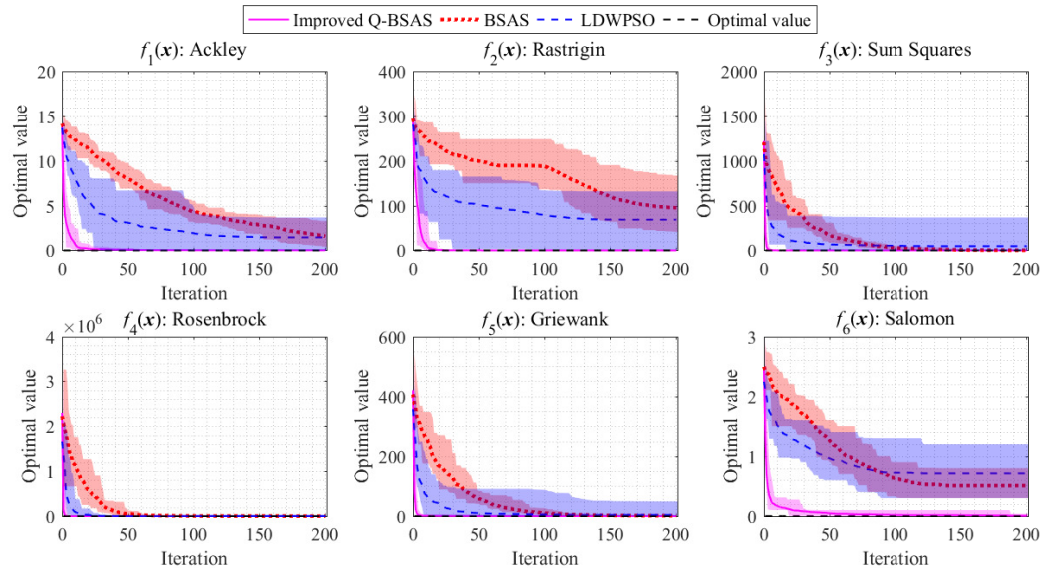


Figure 25: Comparison of the proposed I-Q-BSAS, BSAS and LDWPSO with 20 dimensions.

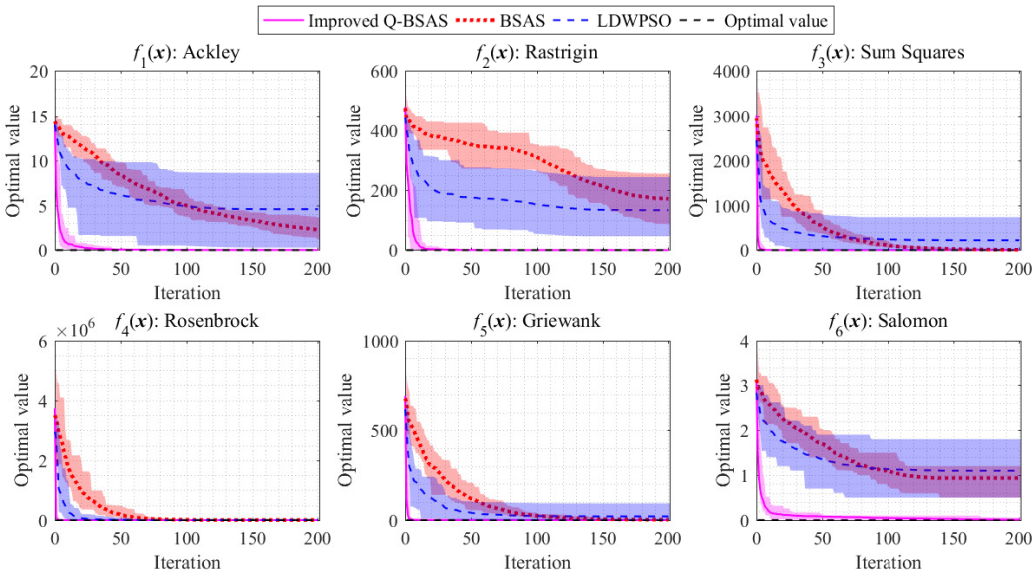


Figure 26: Comparison of the proposed I-Q-BSAS, BSAS and LDWPSO with 30 dimensions.

Table 11: Mean and variance values of optimal fitness of the proposed I-Q-BSAS, original BSAS and LDWPSO.

| Dimensions | | 10 | | | 20 | | | 30 | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Methods | | I-Q-BSAS | BSAS | LDWPSO | I-Q-BSAS | BSAS | LDWPSO | I-Q-BSAS | BSAS | LDWPSO |
| $f_1$ | mean | 3.67E-03 | 4.56E-01 | 1.60E-01 | 7.35E-03 | 1.60E+00 | 1.45E+00 | 8.23E-03 | 2.26E+00 | 4.56E+00 |
| | var | 1.18E-05 | 2.95E-02 | 2.61E-01 | 3.23E-05 | 7.11E-01 | 1.71E+00 | 6.96E-05 | 3.41E-01 | 7.60E+00 |
| $f_2$ | mean | 2.41E-03 | 2.42E+01 | 1.63E+01 | 7.98E-03 | 9.51E+01 | 6.85E+01 | 6.35E-03 | 1.71E+02 | 1.34E+02 |
| | var | 8.02E-06 | 1.38E+02 | 2.16E+02 | 1.82E-04 | 9.91E+02 | 1.00E+03 | 8.73E-05 | 1.71E+03 | 2.26E+03 |
| $f_3$ | mean | 1.48E-05 | 5.63E-02 | 1.31E+00 | 2.93E-04 | 1.01E+00 | 4.66E+01 | 1.01E-03 | 6.47E+00 | 2.20E+02 |
| | var | 3.67E-10 | 8.48E-04 | 3.41E+01 | 1.02E-07 | 2.25E-01 | 9.96E+03 | 2.14E-06 | 1.42E+01 | 4.94E+04 |
| $f_4$ | mean | 9.33E+00 | 4.14E+01 | 5.58E+02 | 2.35E+01 | 9.75E+01 | 9.21E+02 | 3.88E+01 | 2.95E+02 | 1.52E+03 |
| | var | 5.01E-01 | 4.77E+03 | 5.00E+06 | 1.33E+02 | 8.96E+03 | 6.45E+06 | 1.51E+02 | 4.25E+04 | 9.86E+06 |
| $f_5$ | mean | 4.80E-03 | 9.78E-01 | 4.47E-02 | 3.79E-02 | 1.21E+00 | 3.49E+00 | 3.33E-02 | 1.74E+00 | 2.06E+01 |
| | var | 2.67E-05 | 2.46E-02 | 5.69E-03 | 8.99E-03 | 7.31E-03 | 1.16E+02 | 3.36E-03 | 1.04E-01 | 1.03E+03 |
| $f_6$ | mean | 2.13E-03 | 2.10E-01 | 1.80E-01 | 4.87E-03 | 5.10E-01 | 7.15E-01 | 8.31E-03 | 9.35E-01 | 1.10E+00 |
| | var | 4.07E-06 | 2.00E-03 | 2.27E-02 | 7.52E-05 | 1.99E-02 | 7.19E-02 | 1.45E-04 | 5.08E-02 | 1.38E-01 |

fitness curves and fitness ranges in 20 repeated tests with 10, 20 and 30 dimensions are shown in Fig. 24, Fig. 25 and Fig. 26, respectively.

**Remark**: The dimension of the benchmark functions is the dimension of the target variables to be optimized in the functions.

Since random values are used in both I-Q-BSAS, BSAS and LDWPSO, the fitness results of repeated tests at the same iteration are different. The mean and variance values of the final optimal fitness results represent the average ability and stability of the optimization. Therefore, both the mean and variance values are calculated and shown in Table 11. It can be seen from Fig. 24~26 and Table 11 that the proposed I-Q-BSAS can obtain smaller mean and variance values of the optimal fitness results than original BSAS and LDWPSO in all benchmark functions with the same dimension, which indicates the proposed I-Q-BSAS outperforms the original BSAS and LDWPSO in both average ability and stability of optimization.