

基于改进飞蛾扑火优化算法的船机桨匹配设计研究

陈振霖, 罗亮, 郑龙, 姬胜晨, 陈顺怀

引用本文

陈振霖, 罗亮, 郑龙, 姬胜晨, 陈顺怀. 基于改进飞蛾扑火优化算法的船机桨匹配设计研究[J]. 计算机科学, 2024, 51(6A): 230500157-9.

CHEN Zhenlin, LUO Liang, ZHENG Long, JI Shengchen, CHEN Shunhuai. Study on Matching Design of Ship Engine and Propeller Based on Improved Moth-Flame Optimization Algorithm [J]. Computer Science, 2024, 51(6A): 230500157-9.

相似文章推荐 (请使用火狐或 IE 浏览器查看文章)

Similar articles recommended (Please use Firefox or IE to view the article)

[基于混沌映射的改进金枪鱼群优化算法对比研究](#)

Comparative Study on Improved Tuna Swarm Optimization Algorithm Based on Chaotic Mapping
计算机科学, 2024, 51(6A): 230600082-10. <https://doi.org/10.11896/jsjcx.230600082>

[改进双延迟深度确定性策略梯度的多船协调避碰决策](#)

Multi-ship Coordinated Collision Avoidance Decision Based on Improved Twin Delayed Deep Deterministic Policy Gradient
计算机科学, 2023, 50(11): 269-281. <https://doi.org/10.11896/jsjcx.221000131>

[融合麻雀搜索和随机差分的双向学习平衡优化器算法](#)

Bidirectional Learning Equilibrium Optimizer Combining Sparrow Search and Random Difference
计算机科学, 2023, 50(11): 248-258. <https://doi.org/10.11896/jsjcx.221100143>

[SWIPT-MISO动态能量消耗模型下能效规划](#)

Energy Efficiency Planning with SWIPT-MISO Dynamic Energy Consumption Model
计算机科学, 2023, 50(6A): 220400185-7. <https://doi.org/10.11896/jsjcx.220400185>

[一种新的全局优化算法:碳循环算法](#)

New Global Optimization Algorithm: Carbon Cycle Algorithm
计算机科学, 2023, 50(6A): 220300131-7. <https://doi.org/10.11896/jsjcx.220300131>

基于改进飞蛾扑火优化算法的船机桨匹配设计研究

陈振霖 罗亮 郑龙 姬胜晨 陈顺怀

高性能船舶技术教育部重点实验室 武汉 430063

武汉理工大学船海与能源动力工程学院 武汉 430063

(319018@whut.edu.cn)

摘要 基于改进飞蛾扑火优化(Improved Moth-Flame Optimization, IMFO)算法,以两艘现有船舶为计算实例,展开了综合考虑螺旋桨推进效率、空泡性能和桨叶强度的船机桨匹配工作。以遗传算法(Genetic Algorithm, GA)和原始飞蛾扑火优化(Moth-Flame Optimization, MFO)算法为对比算法,分析了IMFO辅助船机桨匹配工作时的性能。数值实验的结果表明,在解决船机桨匹配问题时,IMFO算法的收敛时间相比GA算法在两个算例中分别缩短了44.24%和54.14%,相比MFO算法分别缩短了23.9%和23.12%。此外,在求解精度方面,在计算示例1中,IMFO算法相比GA算法和MFO算法略有提升;而在计算示例2中,IMFO算法相比GA算法提高了3.66%,较MFO算法提高了0.98%。最后,通过对两个算例的可行解空间进行可视化表示,进一步讨论了IMFO算法的求解性能。上述结果对比证明了IMFO算法具备强大的全局搜索能力,在解决船机桨匹配问题时具有良好的竞争力和鲁棒性。

关键词:改进飞蛾扑火优化算法;优化设计;群智能优化算法;船机桨匹配;船用螺旋桨

中图分类号 U664.33

Study on Matching Design of Ship Engine and Propeller Based on Improved Moth-Flame Optimization Algorithm

CHEN Zhenlin, LUO Liang, ZHENG Long, JI Shengchen and CHEN Shunhuai

School of Naval Architecture, Ocean and Energy Power Engineering, Wuhan University of Technology, Wuhan 430063, China

Key Laboratory of High Performance Marine Technology, Ministry of Education, Wuhan 430063, China

Abstract This paper develops an improved moth-flame optimization(IMFO) algorithm for the ship propeller-matching problem, which comprehensively considers propeller efficiency, cavitation, and strength for two existing ships as calculation examples. Genetic algorithm(GA) and the original moth-flame optimization(MFO) algorithm are used as comparison algorithms to analyze the performance of the IMFO-assisted propeller-matching task. Numerical experiment results show that the convergence time of the IMFO algorithm in solving the propeller-matching problem is reduced by 44.24% and 54.14% compared to the GA algorithm in the two examples, and by 23.9% and 23.12% compared to the MFO algorithm, respectively. In addition, in terms of solution accuracy, the IMFO algorithm is slightly better than the GA and MFO algorithms in calculation example 1. In calculation example 2, the IMFO algorithm is improved by 3.66% compared to the GA algorithm and by 0.98% compared to the MFO algorithm. Finally, by visualizing the feasible solution space of the two examples, the performance of the IMFO algorithm is further discussed. The above results demonstrate that the IMFO algorithm has strong global search capability and is competitive and robust in solving the propeller-matching problem.

Keywords Improved moth-flame optimization algorithm, Optimized design, Swarm intelligence optimization algorithm, Matching of ship engine and propeller, Marinepropeller

船用螺旋桨的设计在整个船舶的设计建造工作中占有举足轻重的地位,一个优秀的螺旋桨设计结果可以极大地改善船舶的推进效率。图谱设计法是一种常见的螺旋桨设计方法,该方法使用图谱资料结合船舶的有效马力曲线进行螺旋桨设计,目标是达成船-机-桨之间的匹配^[1]。随着计算机计算能力的不断提高,螺旋桨的优化设计得到了快速发展。Zeng等^[2]使用螺旋桨图谱数据训练人工神经网络模型(Back

Propagation Artificial Neural Networks, BPANNs),得到了一种可以计算螺旋桨水动力性能的近似模型,结合遗传算法的高效求解能力构建了螺旋桨优化设计方法。Wang等^[3]使用荷兰船模试验水池对B系列螺旋桨的回归分析结果,利用群智能优化算法的搜索求解能力,进行了船用螺旋桨优化设计。Jang等^[4]基于希尔伯特空间理论,通过最大化螺旋桨效率函数的方法优化了螺旋桨的螺距分布。Takekoshi^[5]以恒定推

基金项目:国家自然科学基金(52101368)

This work was supported by the National Natural Science Foundation of China(52101368).

通信作者:罗亮(luoliang@whut.edu.cn)

力以及规定的螺旋桨表面空化余量作为约束条件,提高了螺旋桨的敞水效率。Wang 等^[6]以 KP505 桨作为优化对象,使用试验设计方法和椭圆基神经网络(Elliptic Basis Function Probabilistic Neural Network)近似模型结合的方式计算螺旋桨水动力性能,以遗传算法为优化算法,将提高螺旋桨优化设计的质量和效率作为优化目标,提出了一种螺旋桨优化设计方法。Zeng^[7]根据水面舰船的多工况运行需求,借助遗传算法构建了多工况条件下的螺旋桨叶剖面优化设计方法。Huang 等^[8]将 Seiu-Maru HSP 螺旋桨看作优化设计对象,以降低螺旋桨激振力为优化目标,将螺旋桨的侧斜分布看作优化变量,选择粒子群优化算法作为求解算法,进行了螺旋桨优化工作。可以看出,在设计螺旋桨时通常需要综合考虑螺旋桨的敞水效率、螺旋桨的空泡性能、螺旋桨激振力及桨叶强度等多种性能指标要求。在求解多个设计约束的复杂工程问题时,需要利用高效优化算法进行求解。

优化算法是在工程应用中常见的数学算法,分为确定性优化和随机优化两大类,由于确定性优化算法在工程应用中的使用条件十分苛刻,因此随机优化算法得到了迅速发展,出现了一系列的群智能优化算法,如萤火虫算法(Firefly Algorithm, FA)^[9]、谷鸟优化算法(Cuckoo Optimization Algorithm, COA)^[10]、人工蜂群优化算法(Artificial Bee Colony algorithm, ABC)^[11]等。群智能优化算法具有操作简单、易于并行处理和鲁棒性强等特点^[12]。

本文通过改进飞蛾扑火优化(IMFO)算法来辅助进行船用螺旋桨的优化设计工作,构建了综合考虑螺旋桨推进效率、空泡性能以及桨叶强度的船用螺旋桨优化设计方法来解决船机桨匹配问题。以两艘现有船舶为计算实例,以遗传算法和原飞蛾扑火算法为对比算法,详细分析了 IMFO 算法辅助船机桨匹配设计的过程与优势。

1 问题描述

1.1 问题定义与求解框架

通常螺旋桨的设计问题是在给出主机、对应螺旋桨转速 N 以及船舶有效马力曲线的情況下求出最佳推进效率的船机桨匹配问题^[13]。综合考虑敞水效率、空泡性能及桨叶强度等因素,选定系列桨类型和桨叶数,以螺旋桨直径、螺距比及盘面比等螺旋桨几何参数为设计变量进行螺旋桨设计。整个设计过程的分析如下。假设在 IMFO 算法一次迭代下的螺旋桨参数为:

$$V_{prop} = [D, P/D, A_c/A_0] \quad (1)$$

其中, V_{prop} 为螺旋桨速度, P/D 为螺旋桨螺距比, A_c/A_0 为螺旋桨盘面比。

此时 V_{prop} 可以看作已知量,由于已知螺旋桨转速 N 和螺旋桨直径 D ,并且船舶匀速航行时受到的阻力与螺旋桨的推力相等,因此存在关系式(2),为螺旋桨设计时的约束条件,也是力的平衡条件。对于式(2)可以通过迭代法求解出变量 V_m ,该值便为船舶的速度,该值应大于船舶的设计航速,此为另一个约束条件,如式(3)所示。得到船舶速度后便可以确定螺旋桨的进速系数 J ,进而确定螺旋桨效率。确定螺旋桨效率后,由于有效马力 P_E 与有效功率 EHP 相匹配,因此可以通过对应的 EHP 再次反向计算所需的主机功率。若此时反算

出的所需主机功率与给定的主机功率相比更大,则不满足设计条件,需要对当前的螺旋桨几何参数进行调整;若此时反算出的主机功率与给定的主机功率相比更小,大于给定的主机功率的部分被称为储备功率。同样可以将其看作一个约束条件,如式(4)所示。

$$R(V_m) = K_t(V_m, n, D, P/D, A_c/A_0) * \rho n^2 D^4 (1-t) \quad (2)$$

其中, R 为船体受到的阻力, K_t 为推力系数, V_m 为所求速度, ρ 为对应液体密度, t 为推力减额。

$$V_m \geq V_d \quad (3)$$

其中, V_m 为计算出的船舶速度, V_d 为船舶设计航速。

$$P_n \leq P_s \quad (4)$$

其中, P_n 为计算出的所需主机功率, P_s 为给定的主机功率。

对于船机桨匹配问题求解过程中的约束条件,根据涉及约束源头的不同可以将其分为两类:基础类约束和法规约束。基础类约束主要来自对螺旋桨设计过程的理论分析,体来说,基础类约束大多来自设计过程本身需要考虑的条件,这类约束已经通过解约束的方式在螺旋桨设计推导过程中有所体现。法规约束主要来源于相关法律法规,例如中国船级社《钢质海船入级和建造规范》中对螺旋桨的强度做出了一定的要求,除此之外还包括空泡性能约束条件。

上述过程可以看作在使用 IMFO 算法辅助螺旋桨优化设计一次迭代时进行的过程,因此整个 IMFO 算法辅助螺旋桨优化设计计算流程如图 1 所示。

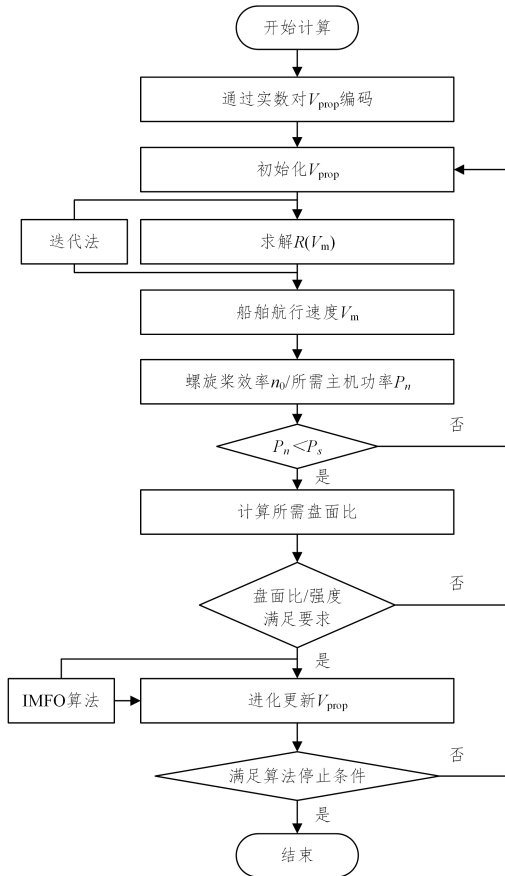


图 1 IMFO 算法辅助螺旋桨优化设计计算流程

Fig. 1 Calculation process of IMFO algorithm assisted propeller optimization design

IMFO 算法辅助螺旋桨优化设计计算的基本流程如下:

1) 对 V_{prop} 进行实数编码计算,将解向量转变为 IMFO 算

法中的主要计算对象;

- 2)初始化 V_{prop} ;
- 3)通过式(2)求解船舶航行速度 $R(V_m)$;
- 4)根据船舶的航行速度,求解螺旋桨效率 η_0 和所需主机功率 P_n ;
- 5)如式(4)判断给定主机功率是否满足所求主机功率,满足要求则进行下一步运算,反之则返回步骤 2);
- 6)计算当前条件下所需盘面比;
- 7)对于满足要求的计算结果,将其输入 IMFO 算法迭代计算后更新 V_{prop} ;
- 8)最终判断结果是否满足终止条件,若满足算法终止要求,则输出问题最优解向量,算法结束;若不满足则返回步骤 2)重新初始化一组 V_{prop} ,重复上述计算。

1.2 螺旋桨性能计算和校核模型

1.2.1 敞水性能计算

在对螺旋桨进行参数化设计时,敞水曲线是必要的计算已知量,其数值的精准度将直接影响螺旋桨优化计算的结果,因此本文通过编写螺旋桨的敞水性能求解器,从而进一步验证现有已知图谱数据的准确性。

在本文螺旋桨的敞水性能计算中,通过系列桨敞水性能图谱数据编制螺旋桨的敞水性能求解器,该求解器可以计算敞水性能的螺旋桨范围,如表 1 所列。

表 1 计算敞水性能的螺旋桨范围

Table 1 Calculation of propeller range for open water performance

	B 系列 5 叶桨		MAU 系列 4 叶桨	
	下界	上界	下界	上界
螺旋桨盘面比	0.45	1.05	0.40	0.70
螺旋桨螺距比	0.60	1.40	0.60	1.60

用于数据插值拟合的螺旋桨敞水数据条目分别为 B 系列 5 叶桨推力系数 649 条,B 系列 5 叶桨转矩系数 666 条,MAU 系列 4 叶桨推力系数 464 条,MAU 系列 4 叶桨转矩系数 438 条,该求解器的准确性分析如图 2—图 5 所示。可以看出,该求解器能够满足计算要求。

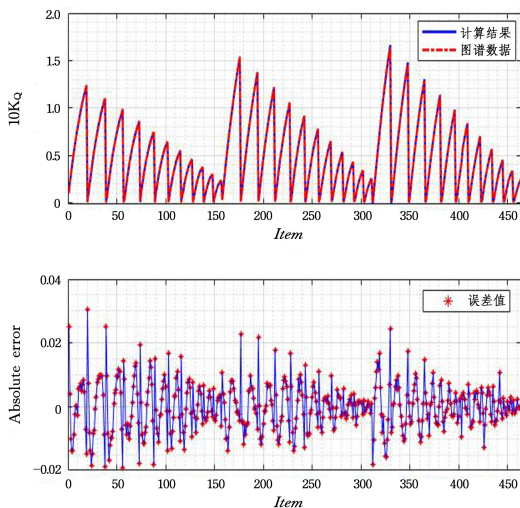


图 2 求解器计算 MAU 系列 4 叶桨 $10K_Q$ 值准确性分析
Fig. 2 Accuracy analysis of solver calculation of $10K_Q$ value of MAU series 4-blade propeller

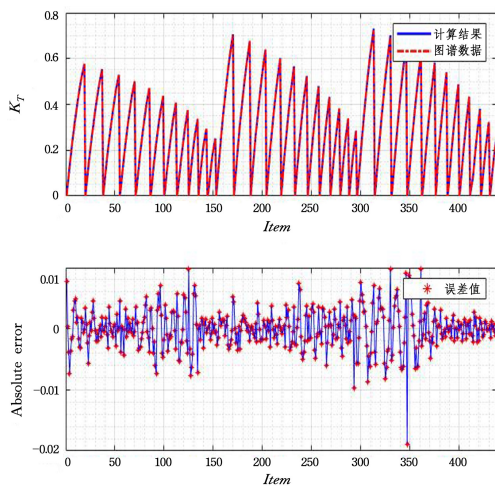


图 3 求解器计算 MAU 系列 4 叶桨 K_T 值准确性分析
Fig. 3 Accuracy analysis of solver calculation of K_T value of MAU series 4-blade propeller

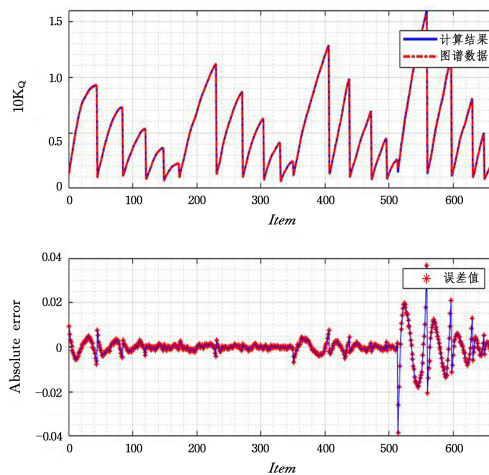


图 4 求解器计算 B 系列 5 叶桨 $10K_Q$ 值准确性分析
Fig. 4 Accuracy analysis of solver calculation of $10K_Q$ value of B series 5-blade propeller

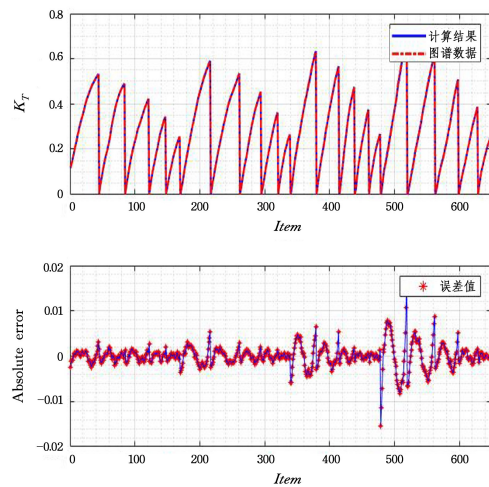


图 5 求解器计算 B 系列 5 叶桨 K_T 值准确性分析
Fig. 5 Accuracy analysis of solver calculation of K_T value of B series 5-blade propeller

1.2.2 空泡校核计算

目前螺旋桨空泡计算技术有基于势流和粘流理论的数值分析方法,然而这些方法的时间成本相对较高。因此,在优化

设计的迭代过程中需要选择一种响应速度较快且可以保证正确推算方向的计算方式。Burrill 根据各类船舶螺旋桨的统计资料提出了不同背空泡面积比率下的商船空泡限界,根据柏利尔限界(Burrill cavitation line)可以计算出满足空泡性能要求的最小盘面比,计算式如式(5)所示:

$$A_r/A_0 = \frac{8K_i}{\pi(1.067 - 0.229P/D)(J^2 + 0.49\pi^2)\tau_{c\min}} \quad (5)$$

其中, K_i 为推力系数, P 为螺旋桨螺距, D 为螺旋桨直径, J 为进速系数, τ_c 为允许的桨叶推力负荷系数, 该值可以通过柏利尔限界线查得。

根据空泡数在限界线图所在的位置分析可得允许的桨叶推力负荷系数, 进而得到该桨的背空泡比率, 背空泡比率小于 10% 时对推进性能基本没有影响, 比率小于 2.5% 时可以认为不会发生空泡。2.5% 和 10% 背空泡限界线可以根据式(6)和式(7)计算^[14]。

$$\tau_{c2.5\%} = \exp(0.5425(\ln\sigma_{0.7R})^3 + 0.0906(\ln\sigma_{0.7R})^2 + 0.4965\ln\sigma_{0.7R} - 0.4306) \quad (6)$$

$$\tau_{c10\%} = \exp(0.2901(\ln\sigma_{0.7R})^3 + 0.1723(\ln\sigma_{0.7R})^2 + 0.4624\ln\sigma_{0.7R} - 0.6075) \quad (7)$$

其中, $\sigma_{0.7R}$ 为螺旋桨半径 0.7R 处空泡数, 计算式如式(8)所示:

$$\sigma_{0.7R} = \frac{P_0 - P_V}{0.5\rho V_{0.7R}^2} \quad (8)$$

其中, $V_{0.7R}$ 为螺旋桨半径 0.7R 处的来流速度, P_0 为桨轴中心线处的静压强, P_V 为水的汽化压强。

1.2.3 桨叶强度校核计算

根据中国船级社《钢质海船入级和建造规范》对螺旋桨的强度进行约束, 如式(9)所示:

$$t' \geq t^r = \sqrt{\frac{Y}{k-x}} \quad (9)$$

其中, t' 和 t^r 分别为规范中所需的桨叶厚度和计算出的桨叶厚度(定距桨为 0.25 r/R 和 0.6 r/R, 变距桨为 0.35 r/R 和 0.6 r/R, Y 为功率系数, k 为材料系数, x 为转速系数)。

2 算法设计

2.1 飞蛾扑火优化(MFO)算法

飞蛾扑火优化(MFO)算法主要灵感来源于飞蛾被称为“横定向”的导航方法。如图 6 所示, 飞蛾在夜间飞行时会与附近较强光线成一固定角度飞行, 这是一种非常有效的远距离直线飞行机制, 月光离地球足够远因此可视为平行光, 但是在人造光周围, 飞蛾却容易陷入致命的螺旋路径, MFO 算法主要就是模拟飞蛾的这种特点。

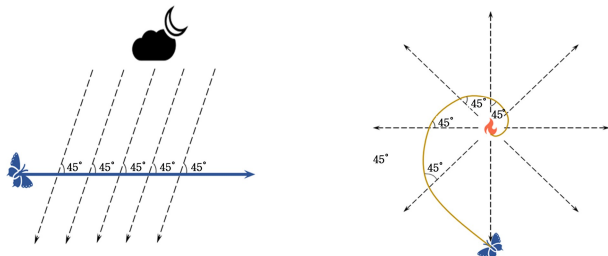


图 6 MFO 算法中的飞蛾趋光飞行模型

Fig. 6 Model of moth phototactic flight in MFO algorithm

在 MFO 算法中, 单个“飞蛾” m 所在的矢量位置为待求解问题的候选解, 可通过式(10)表示:

$$M = [m_1, m_2, m_3, \dots, m_i] \quad (10)$$

其中, i 表示待求解变量的维度大小, 即待求的优化变量的个数, MFO 算法属于群体智能优化算法的范围, 因此在计算有 i 个“飞蛾”的“飞蛾”种群时可通过式(11)所示的矩阵来表示。

$$M = \begin{bmatrix} m_{1,1} & m_{1,2} & \dots & m_{1,j} \\ m_{2,1} & m_{2,2} & \dots & m_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ m_{i,1} & m_{i,2} & \dots & m_{i,j} \end{bmatrix} \quad (11)$$

其中, i 表示“飞蛾”数量, j 表示“飞蛾”飞行维度大小, 即待求的优化变量个数。在 MFO 算法中, 每只“飞蛾”都有一个对应的“火焰”, 因此两者维度相同, 则描述“火焰”位置的矩阵为:

$$F = \begin{bmatrix} F_{1,1} & F_{1,2} & \dots & F_{1,j} \\ F_{2,1} & F_{2,2} & \dots & F_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ F_{i,1} & F_{i,2} & \dots & F_{i,j} \end{bmatrix} \quad (12)$$

对于所有的“飞蛾”和“火焰”, 假设存在一个用于存储两者适应度值的数组分别为 OM 和 OF 。

$$OM = [OM_1, OM_2, \dots, OM_n]^T \quad (13)$$

$$OF = [OF_1, OF_2, \dots, OF_n]^T \quad (14)$$

此时, MFO 算法可用一个三元组表示:

$$MFO = (I, P, T) \quad (15)$$

其中, I 为随机生成的种群及其对应适应度值的函数 $I: \mathcal{O} \rightarrow \{M, OM\}$; P 为主函数, 其主要功能为在搜索空间内移动“飞蛾”; P 函数接收矩阵 M 并最终返回更新后的矩阵 $MP: M \rightarrow M$; T 函数用于判断是否返回终止条件 $T: M \rightarrow \{\text{True}, \text{False}\}$ 。

初始化完成后, 运行 P 函数直至 T 返回 True 值, 其作用就是模拟横向往定向移动, 每只“飞蛾”相对于“火焰”的位置按照式(16)进行更新。

$$M_i = S(M_i, F_j) \quad (16)$$

其中, M_i 表示第 i 只“飞蛾”, F_j 表示其对应的第 j 个“火焰”, 函数 S 为螺旋函数, 其定义为:

$$S(M_i, F_j) = D_i \cdot e^{i\theta} \cdot \cos(2\pi t) + F_j \quad (17)$$

式(8)模拟了“飞蛾”的螺旋飞行路径, “飞蛾”的下一位置也是根据“火焰”定义的。图 7 模拟了“飞蛾”在火焰周围的位置更新, 纵轴显示了一个解维度, “飞蛾”可在如图 7 所示的空间内探索“火焰”周围的搜索空间。

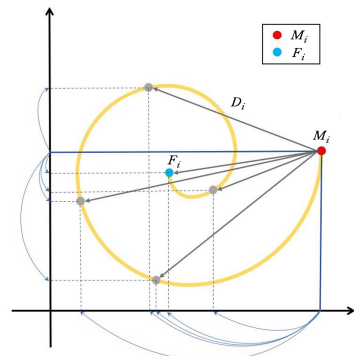


图 7 MFO 算法中的飞蛾飞行轨迹图

Fig. 7 Diagram of moth flight trajectory in MFO algorithm

图 7 中, D_i 为第 i 只“飞蛾”与第 j 个“火焰”的距离, b 是定义螺旋桨形状的常量, t 是 $[-1, 1]$ 之间的随机数。 D 的计算式如下:

$$D_i = |F_j - M_i| \quad (18)$$

此处定义两个向量 ub, lb 用于定义搜索变量的上下界:

$$ub = [ub_1, ub_2, ub_3, \dots, ub_{n-1}, ub_n] \quad (19)$$

$$lb = [lb_1, lb_2, lb_3, \dots, lb_{n-1}, lb_n] \quad (20)$$

其中, n 为变量维度。

为避免“飞蛾”在搜索域内 n 个位置的位置更新可能会降低对最优解的利用, 提出“火焰”自适应机制, 即:

$$F_n = \text{round}\left(N - t * \frac{N-1}{T}\right) \quad (21)$$

其中, t 为当前迭代次数, N 为最大“火焰”数, T 为最大迭代次数, F_n 为当前“火焰”数量。

MFO 算法的主要流程如图 8 所示, 具体描述如下。

- 1) 计算对象编码, 将优化问题的解向量通过实数编码的方式转变为 MFO 算法中的主要计算对象;
- 2) 初始化计算对象, 在编码空间中按照算法初始化规则生成一组初始解向量作为初始主要计算对象, 同时按照算法计算规则生成一组次要计算对象用于控制算法计算进程;
- 3) 根据优化问题设定的优化目标计算主要计算对象的个体适应度;
- 4) 判断算法是否满足终止条件, 若不满足则更新“火焰”数量及“火焰”和“飞蛾”的位置信息进入下一次迭代过程, 若满足则直接输出寻优结果;
- 5) 判断算法是否满足终止条件, 若不满足则返回步骤 3), 进入下一次迭代过程, 若满足则直接输出寻优结果;
- 6) 对主要计算对象进行解码操作, 并输出问题最优解向量, 算法结束。

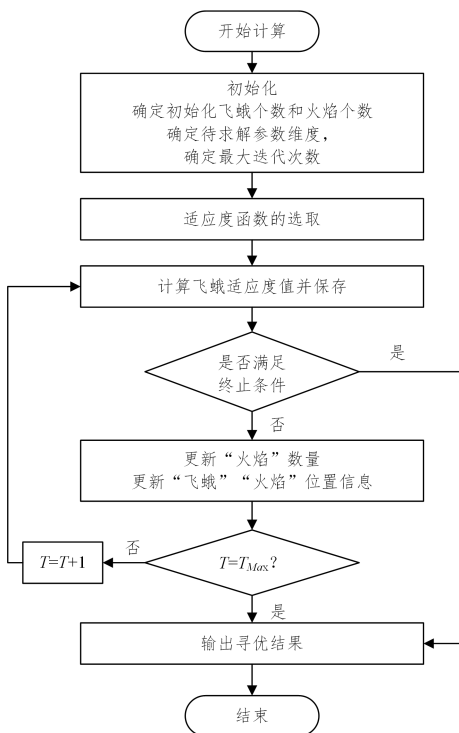


图 8 MFO 算法的计算流程

Fig. 8 Calculation process of MFO algorithm

2.2 基于 Sobol 序列的改进飞蛾扑火算法

2.2.1 基于 Sobol 序列初始化种群位置

飞蛾扑火优化算法属于元启发式算法, 在处理工程问题时, 这类算法的初始化在很大程度上会影响结果的准确性^[15]。飞蛾扑火算法进行初始化时要随机生成种群, 然而传统随机数的生成会受限于电脑位数或 CPU 的随机生成算法等因素, 在大于某固定周期时易出现重复的情况, 这种伪随机数策略并不能保证其随机数可以保持统计意义上的独立性。因此, 本文基于 Sobol 序列对种群进行初始化, 对原始 MFO 算法进行改进。

Sobol 序列是一种随机序列, 与传统随机数相比, Sobol 采样使用了不同的采样方式, 算法着重于将空间均匀填满, 而这种均匀填满不同于网格, 其本质仍是使用随机方法。伪随机数与 Sobol 随机数空间分布如图 9 所示。

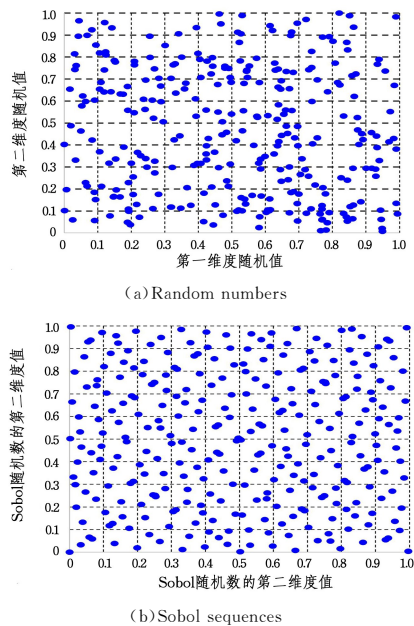


图 9 传统随机数与 Sobol 序列对比图

Fig. 9 Comparison chart of traditional random numbers and Sobol sequences

采样点的位置数据在采样空间内是否为服从均匀分布的随机数对于“飞蛾”种群的初始化有着重要意义, 本文在处理多维度的“飞蛾”种群时, 假设最优解的取值范围为 $[X_{lb}, X_{ub}]$, Sobol 序列产生的随机数为 $K_n \in [0, 1]$, 则“飞蛾”的初始位置可定义为:

$$X_n = X_{lb} + K_n * (X_{ub} - X_{lb}) \quad (22)$$

2.2.2 改进飞蛾扑火算法惯性权重

更新“飞蛾”与“火焰”的位置信息是飞蛾扑火算法的重要一环, 原始算法对搜索范围和收敛速度都未进行相关参数控制, 因此在位置更新的过程中容易产生个体之间的碰撞, 导致陷入局部最优解。附加变量 A 对防止所有个体迅速向最优解靠拢及平衡算法有着重要作用。惯性权重 A 的计算式为:

$$A = \exp\left(1 - \frac{t_{\max} + t}{t_{\max} - t}\right) \quad (23)$$

图 10 给出了迭代次数为 300 次的 A 值, 其中在迭代前期由于搜索范围大, 因此 A 值也较大, 有利于全局搜索最优解, 迭代后期 A 值较小, 有利于在已发现的解空间区域内提高收敛速度, 找到最优值。

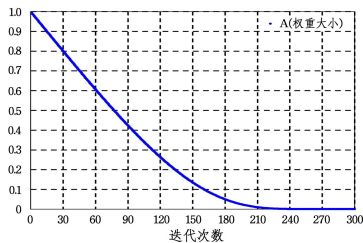


图 10 权重随迭代次数变化图

Fig. 10 Graph of weight change with the number of iterations

综上所述,改进后种群位置的更新方式如下:

$$M_i = A * S(M_i, F_j) \tag{24}$$

3 IMFO 算法辅助船机桨匹配工作实验验证

3.1 实验环境及算法的参数设置

该项实验使用的操作系统为 Windows 11,使用的 MATLAB 版本为 Matlab2021a,在硬件方面,CPU 为八核 12th Gen Intel(R) Core(TM) i7-12700H 2.70 GHz,内存 48 GB, GPU 为 Nvidia Tesla P100,显存 16 GB。

为了分析 IMFO 算法在辅助解决船机桨匹配问题的优势,选择原始 MFO 算法和 GA 算法作为对比。GA 算法的参数设置如表 2 所列,MFO 算法的参数设置如表 3 所列。

表 2 GA 算法的参数设置

Table 2 Parameter setting of GA algorithm

参数名称	数值
种群数量	30
终止进化代数	300
交叉概率	0.6
变异概率	0.1

表 3 MFO 算法的参数设置

Table 3 Parameter setting of MFO algorithm

参数名称	数值
种群数量	30
终止进化代数	300

在进行 IMFO 算法的参数设置时需要保持与 GA 算法和 MFO 算法参数设置下适应度函数的调用次数一致,因此 IMFO 算法的参数设置如表 4 所列。

表 4 IMFO 算法的参数设置

Table 4 Parameter setting of IMFO algorithm

参数名称	数值
种群数量	30
终止进化代数	300

3.2 计算示例 1

在该算例中以某小型货船为设计对象进行螺旋桨优化设计分析,该船的主尺度等相关参数如表 5 所列。

表 5 小型货船主尺度等相关参数

Table 5 Relevant parameters such as main dimensions of small cargo ship

船舶参数	符号	单位	数值
垂线间长	Loa	m	20.8
型宽	B	m	4.50
型深	D	m	1.50
满载平均吃水	T	m	0.75
满载排水量	Δ	t	39.844
螺旋桨数目	Z	个	2
桨轴中心距基线	Zp	m	0.30
螺旋桨材料	—	—	Cu3 镍铝青铜

经过测量,该船设计满载吃水阻力换算结果如表 6 及图 11 所示,推力减额及伴流分数如表 7 所列。

表 6 小型货船满载吃水阻力与有效功率

Table 6 Draft resistance and effective power of small cargo ship with full load

Vs/kn	1/2 阻力/kgf	1/2EHP/hp
3.0	23.90	0.544
4.0	41.70	1.088
5.0	69.20	2.312
6.0	112.90	4.624
7.0	168.20	8.024
8.0	247.80	13.600
9.0	356.80	22.032

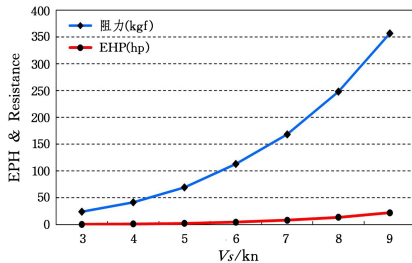


图 11 小型货船阻力和有效功率曲线

Fig. 11 Resistance and effective power curves of small cargo ship

表 7 小型货船的推力减额及伴流分数

Table 7 Thrust derating and wake fraction for small cargo ship

名称	数值
推力减额	0.1
伴流分数	0.15

根据设计经验及螺旋桨的敞水性能求解器参数范围,设定设计变量的取值范围为:

$$D \in [0.4, 0.8], A_E/A_0 \in [0.4, 1.0], P/D \in [0.6, 1.4]$$

GA 算法、MFO 算法和 IMFO 算法求解该问题的迭代过程如图 12 所示。

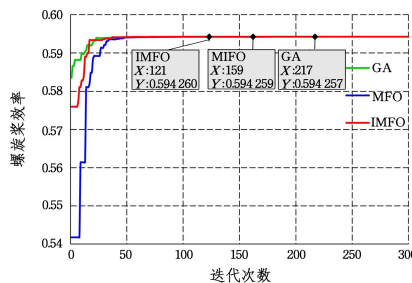


图 12 算例 1 中 3 种算法迭代过程对比图

Fig. 12 Comparison chart of iterative process of three algorithms in example 1

3 种算法的设计计算结果对比如表 8 所列。

表 8 算例 1 设计计算结果对比

Table 8 Comparison of design and calculation results in example 1

参数名称	GA	MFO	IMFO
直径/m	0.5546	0.5546	0.5546
盘面比	0.4281	0.4281	0.4281
螺距比	0.8000	0.8000	0.8000
螺旋桨效率	0.5943	0.5943	0.5943
迭代次数	217	159	121

表9 算例1IMFO算法与另两种算法运算效果对比

Table 9 Comparison of operation effect of IMFO algorithm and other two algorithms in example 1

对比算法	运算时间提升 (%)
GA	44.24
MFO	23.9

由表8及表9可知,算例1中,在相同计算条件下IMFO算法的运算精度与另外两种算法的差距不大。IMFO算法在运算时间方面相比GA算法节省了44.24%,相比MFO算法节省了23.9%的运算时间,因此IMFO算法的运算效果明显优于GA算法和MFO算法。

3.3 计算示例2

在该算例中以某中型货船为设计对象进行螺旋桨优化设计分析,该船的主尺度等相关参数如表10所列。

表10 中型货船主尺度等相关参数

Table 10 Main dimensions and other relevant parameters of medium-sized cargo ship

船舶参数	符号	单位	数值
垂线间长	Loa	m	54.80
型宽	B	m	11.60
型深	D	m	3.80
满载平均吃水	T	m	2.50
满载排水量	Δ	t	875.00
螺旋桨数目	Z	个	2
桨轴中心距基线	Z _P	m	1.50
螺旋桨材料	—	—	Cu3 镍铝青铜

经过测量,该船设计满载吃水阻力换算结果如表11及图13所示,推力减额及伴流分数如表12所列。

表11 中型货船满载吃水阻力与有效功率

Table 11 Full load draft resistance and effective power of medium-sized cargo ship

V _s /kn	阻力/kgf	EHP/hp
10.80	5417	401.2
12.42	8249	703.12
14.04	12262	1180.48
15.12	15220	1578.96
16.20	19457	2162.40
17.28	826347	3122.56

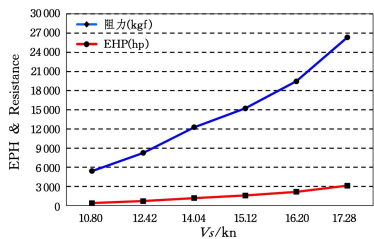


图13 中型货船阻力和有效功率曲线

Fig. 13 Resistance and effective power curves of medium-sized cargo ship

表12 中型货船的推力减额及伴流分数

Table 12 Thrust deduction and wake fraction of medium-sized cargo ship

名称	数值
推力减额	0.03
伴流分数	0.03

同样选择MFO算法与GA算法进行对比。根据设计经验及螺旋桨的敞水性能求解器参数范围,设定设计变量

的取值范围为:

$$D \in [1.5, 2.5], A_E/A_0 \in [0.45, 1.05], P/D \in [0.6, 1.4]$$

GA算法、MFO算法和IMFO算法求解该问题的迭代过程如图14所示。

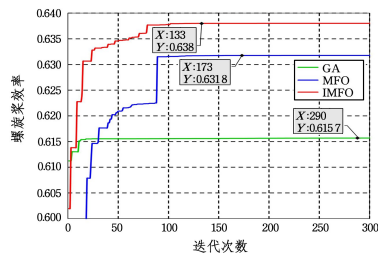


图14 算例2中3种算法迭代过程对比图

Fig. 14 Comparison chart of iterative process of three algorithms in example 2

3种算法的计算结果如表13和表14所列。

表13 算例2设计计算结果对比

Table 13 Comparison of design and calculation results in example 2

参数名称	GA	MFO	IMFO
直径/m	2.0388	1.8784	2.0840
盘面比	0.7920	0.7934	0.5631
螺距比	0.9610	1.0000	0.8000
螺旋桨效率	0.6155	0.6318	0.6380
迭代次数	290	173	133

表14 算例2IMFO算法与另两种算法运算效果对比

Table 14 Comparison of operation effect of IMFO algorithm and other two algorithms in example 2

对比算法	运算时间提升 (%)	运算精度提升 (%)
GA	44.24	3.66
MFO	23.9	0.98

在算例2中,IMFO算法在运算精度方面较MFO算法提升了0.98%,较GA算法提升了3.66%。在运算时间方面,IMFO算法较MFO节省了23.12%的运算时间,较GA节省了54.14%的运算时间,因此不难看出IMFO算法无论是在运算精度方面还是运算时间方面都明显优于MFO和GA算法。

4 算例分析

4.1 IMFO算法及其中间变体的计算案例分析

本文针对算法的改进分为两部分,分别为对初始化随机序列的改进和对飞蛾扑火算法惯性权重的改进。为进一步验证本文改进算法的有效性,对比了仅采用Sobol序列初始化种群位置的MFO算法,以及仅采用改进惯性权重的MFO算法这两种算法在算例1和算例2上的求解性能,计算参数均和第3章中的计算参数相同,求解结果如图15和图16所示。

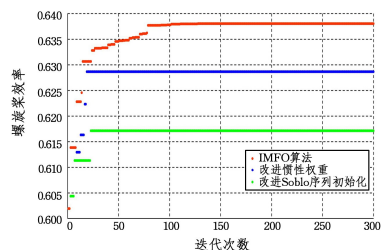


图15 算例1算法中间变体迭代过程对比图

Fig. 15 Comparative diagram of iteration process for intermediate variant of algorithm in example 1

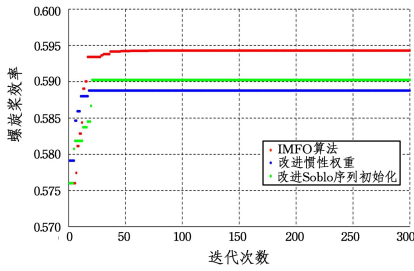


图 16 算例 2 算法中间变体迭代过程对比图

Fig. 16 Comparative diagram of iteration process for intermediate variant of algorithm in example 2

表 15 IMFO 算法与另两种中间变体运算结果对比

Table 15 Comparison of operation results between IMFO algorithm and two other intermediate variants

算法变体	算例 1	算例 2
IMFO 算法	0.5943	0.6380
仅采用 Sobol 序列初始化种群位置的 MFO 算法	0.5887	0.6287
采用改进惯性权重的 MFO 算法	0.5902	0.6172

上述计算结果可明显看出,IMFO 算法在计算准确性上均优于两种中间变体,在两种变体算法中仅采用 Sobol 序列初始化种群位置的 MFO 算法在计算时能均匀地将种群初始化,但容易陷入局部最优而停止搜索,因此计算结果比 IMFO 算法差,而采用改进惯性权重的 MFO 算法可在计算过程中快速收敛到某一值,由于在对初始值进行初始化时未搜索到最优解,因此计算结果差于 IMFO 算法。

4.2 解空间分析

对比分析上述两算例可以发现,算例 1 中 IMFO 算法、MFO 算法和 GA 算法找到相同最优解,算例 2 中 IMFO 算法找到最优解而另外两种算法未找到最优解。为进一步探索其中原因,本文采用蒙特卡洛分析的方法将算例 1 和算例 2 的解空间可视化进行分析。

蒙特卡洛法也叫做统计模型法,是把概率现象作为研究对象进行数值模拟,并通过抽样计算求得统计值来将某一计算特性进行可视化的计算方法。在上文中,式(10)与式(11)用于设定最优解的求解范围,本文从螺距比、盘面比和直径 3 个维度进行求解,因此可将解空间可视化为一立方体,遍历搜索空间中的所有解并将有解的部分进行标注,则可将解空间进行可视化,如图 17 和图 18 所示。

在优化算法中,当解空间中有解的部分占比相对较大时,随机初始化后找到可行解就更加容易,则寻找到最优解的可能性也更高;反之解空间越小,则随机初始化后无法计算得到有效解,后续就很难找到最优解。从图 17 和图 18 的对比中可以看出,算例 1 的解空间有解区域相对较大,因此上述 3 种优化算法均能找到对应的最优解向量。IMFO 算法在求解速度上相对另外两种算法有显著优势。算例 2 的解空间明显小于算例 1 的解空间,优化算法寻找解变得相对困难,此时 IMFO 算法的初始化方法和寻求最优解的探索方案要明显优于另外两种算法。因此,上述计算结果表明,在解空间相对狭小的区域,IMFO 算法较另外两种算法有着求解速度快、求解精度高的优点。

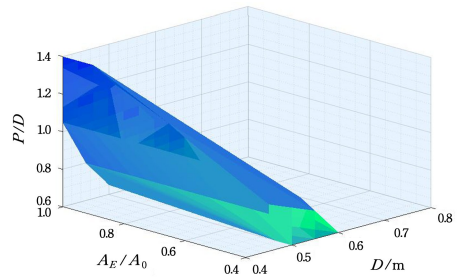


图 17 算例 1 解空间

Fig. 17 Solution space of example 1

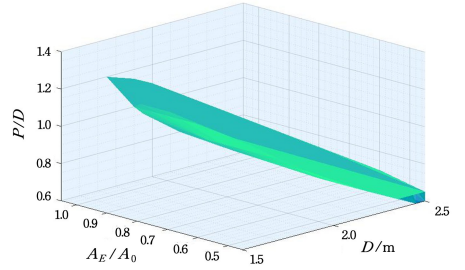


图 18 算例 2 解空间

Fig. 18 Solution space of example 2

结束语

本文针对船机桨匹配问题建立了有效的快速计算模型,基于原始 MFO 计算模型提出了 IMFO 算法,使“飞蛾”种群在初始化的过程中能进行更加有效的初始化位置分布,对“飞蛾”种群的多样性进行了很好的修正,同时,有效避免了局部最优解的产生。此外,本文将 GA 算法、MFO 算法与 IMFO 算法进行了对比分析,从两个算例的计算结果中可以看出,改进后的飞蛾优化算法在计算精度和收敛时间方面较 GA 算法和 MFO 算法具有一定的提升。最后,本文对两个算例进一步进行了分析讨论。在相对较大的解空间中 IMFO 算法和其他两种算法都可以找到相似的最优解,但在收敛时间方面 IMFO 算法表现出一定的竞争力;在相对较小的解空间中 IMFO 算法的计算精度和收敛时间都有着显著优势。

目前,本文螺旋桨水动力性能评估部分是基于图谱资料的方法,因此具有一定的局限性,在未来的工作中将加入更加灵活的水动力性能评估方法,以实现更加全面的船机桨匹配设计工作。

参考文献

- [1] SHENG Z B, LIU Y Z. Principles of ships(below) [M]. Shanghai Jiaotong University Press, 2004.
- [2] ZENG Z B, DING E B, TANG D H. Optimal Design of Ship Propeller Based on BP Artificial Neural Network and Genetic Algorithm[J]. Ship Mechanics, 2010, 14(1): 20-27.
- [3] WANG P, HUANG S, ZHU Z Q. Application of Swarm Intelligence Algorithm in Optimal Design of Propeller Parameters[J]. Computer Science, 2013, 40(2): 73-76.
- [4] JANG T S, KINOSHITA T, YAMAGUCHI H. A new functional optimization method applied to the pitch distribution of a marine propeller [J]. Journal of Marine Science and Technology, 2001, 6(1): 23-30.
- [5] TAKEKOSHI Y, KAWAMURA T, YAMAGUCHI H, et al.

- Study on the design of propeller blade sections using the optimization algorithm[J]. *Journal of Marine Science and Technology*, 2005, 10(2): 70-81.
- [6] WANG C, HAN K, SUN C, et al. Optimal Design and Parameter Analysis of Marine Propeller[J]. *Journal of Huazhong University of Science and Technology(Natural Science Edition)*, 2020, 448(4): 102-107.
- [7] ZENG Z. Optimal Design of Propeller Blade Sections of High-Speed Surface Ships under Multiple Operating Conditions[J]. *China Shipbuilding*, 2018, 59(1): 26-35.
- [8] HUANG B, XIONG Y, WANG B. Optimization of propeller skew distribution based on particle swarm optimization[J]. *China Ship Research*, 2016(6).
- [9] LIU C, YE C. A Novel Bionic Swarm Intelligent Optimization Algorithm: Firefly Algorithm [J]. *Computer Application Research*, 2011, 28(9): 3295-3297.
- [10] RAJABIOUN R. Cuckoo Optimization Algorithm[J]. *Applied Soft Computing Journal*, 2011, 11(8): 5508-5518.
- [11] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm[J]. *Global Optimization*, 2007, 39(3): 459-471.
- [12] MIRJALILI S. Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm[J]. *Knowledge-Based Systems*, 2015, 89: 228-249.
- [13] LIU C. Multidisciplinary Optimal Design of Marine Propeller Based on Complex System Modeling Theory[D]. *Jiangsu University of Science and Technology*, 2014.
- [14] YANG L C, YANG C J, LI X B. Research on Optimal Design of Propeller Based on Multi-objective Evolutionary Algorithm and Decision-Making Technology [J]. *China Shipbuilding*, 2019, 60(3): 12.
- [15] DOKEROGLU T. A survey on new generation metaheuristic algorithms[J]. *Computers & Industrial Engineering*, 2019, 137(C): 106040-106040.



CHEN Zhenlin, born in 1998, postgraduate. His main research interests include intelligent computing and simulation integration.



LUO Liang, born in 1980, Ph.D, associate professor. His main research interests include digital ship design, system simulation integration, intelligent computing, and more.