International Journal of Mechanical and Electrical Engineering
ISSN: 3005-9615 (Print), ISSN: 3005-7132 (Online) | Volume 7, Number 1, Year 2025
DOI: https://doi.org/10.62051/ijmee.v7n1.08
Journal homepage: https://wepub.org/index.php/IJMEE/index

GOA-Optimized Control Strategy for Stall and Surge in Centrifugal Compressors

Qingyuan Zhang

School of Xi'an Shiyou University, Xi'an Shaanxi, 710065, China

ABSTRACT

As core power equipment, the stable operation of centrifugal compressors is critical to industrial systems. Stall and surge represent severe unstable operating conditions that threaten compressor safety, potentially leading to equipment damage and production interruptions. To address the limitations of traditional control strategies in handling system nonlinearity, time-varying characteristics, and model uncertainty, this paper proposes an intelligent control method based on the Grasshopper Optimization Algorithm (GOA). The study first establishes a nonlinear model that accurately reflects the compressor's dynamic characteristics. Subsequently, advanced controllers (such as fuzzy PID and model predictive controllers) are designed with the objectives of expanding stability margin and suppressing pressure pulsations. To tackle the challenge of tuning critical controller parameters, the GOA algorithm is employed for global optimization. Leveraging its robust optimization capabilities and convergence speed, it adaptively obtains optimal parameter combinations to enhance the system's dynamic response performance and disturbance rejection capability. Simulation and experimental results demonstrate that compared to conventional PID and empirical tuning methods, the GOA-optimized control strategy more effectively delays the surge onset boundary, rapidly suppresses pressure oscillations under disturbance conditions, and significantly enhances the operational stability and control quality of centrifugal compressors. This provides a novel solution for the safe, efficient, and intelligent operation of compressors.

KEYWORDS

Centrifugal Compressor; Stall Surge; Locust Optimization Algorithm.

1. INTRODUCTION

Today, climate change has increasingly become a global focal point, with China steadily advancing toward the early realization of its "dual carbon goals."As critical power equipment, centrifugal compressors are widely applied across industrial sectors, their importance self-evident. With the advancement of industrial modernization, the industry's demand for high-efficiency, energy-saving centrifugal compressors continues to rise. As a premium low-carbon clean energy source, natural gas demonstrates significant advantages over traditional fossil fuels in various fields. Its share in energy consumption continues to rise, and the transportation demands of long-distance gas pipelines increase year by year. Centrifugal compressors, with their simple structure, reliable operation, and wide flow regulation range, have become the core power equipment for gas pressurization in long-distance pipeline projects. However, during operation, centrifugal compressors may experience surge phenomena due to stall, threatening both the safety of the compressor units and the stable operation of the production process.

2. PRINCIPLES OF STALL AND SURGE IN CENTRIFUGAL COMPRESSORS

When the compressor operates under design conditions, the airflow through the flow path, the cross-sectional area of the flow path, and the blade geometry are well-matched. When the airflow direction aligns with the geometric angle of the blades, the compressor's stages work in coordination and achieve high efficiency.

2.1. Formation of Stall

When a compressor operates under non-design conditions, severe airflow separation occurs in the blade passages due to operating changes, leading to rotational separation and a significant deterioration in stage performance. This phenomenon is termed “rotational stall.” Based on the severity of the stall, it can be further classified into “progressive stall” and “transient stall.” When the flow rate in a stage deviates from the design value and gradually decreases, separation zones begin to form within the blade passages. These zones progressively expand in area and increase in the number of vortices. On the performance curve, this manifests as the operating point shifting from left to right, with the pressure ratio gradually decreasing. The curve becomes smooth and continuous, and this type of stall is termed “progressive stall.”

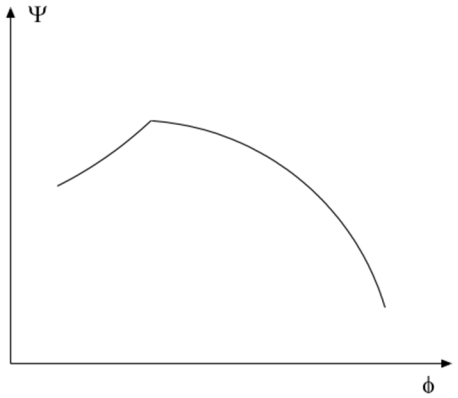

Figure 1. Progressive Stall

Another type of stall progresses rapidly. When flow rate drops below a critical threshold, large-scale separation suddenly occurs, occupying a significant proportion of the flow area. At this point, performance drops sharply and abruptly, exhibiting discontinuity on the performance curve. This phenomenon is termed “sudden stall.”

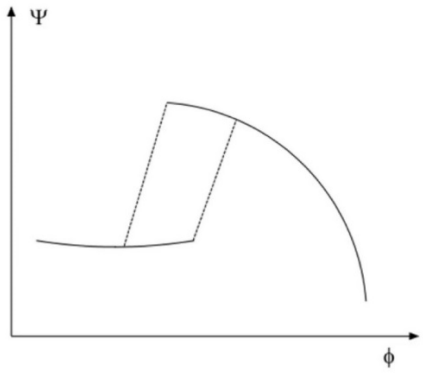

Figure 2. Sudden stall

2.2. Surge

Within the compressor flow path, significant reductions in flow rate due to operating condition changes can trigger severe rotational separation. When a sudden stall occurs, flow conditions rapidly deteriorate. Although the impeller continues rotating and performing work on the gas, it fails to increase gas pressure, leading to a significant drop in outlet pressure. Since compressors operate in conjunction with pipeline networks, if the network capacity is large, its response becomes less sensitive. In such cases, the pipeline pressure may not decrease immediately, potentially resulting in the pipeline pressure exceeding the compressor outlet pressure. This causes gas to flow back into the compressor until the pipeline pressure drops below the compressor outlet pressure. At this point, the reverse flow ceases. The gas flow resumes normally under the action of the blades, and the compressor begins supplying gas to the pipeline network again. The flow rate through the compressor increases, and the compressor returns to normal operation. However, as the pressure in the pipeline network gradually recovers to its original level, normal exhaust is obstructed, the flow rate decreases, and reverse flow of gas occurs within the system. This cycle repeats, causing periodic axial low-frequency gas flow oscillations with large amplitude throughout the system—a phenomenon known as "surge."

3. THE PRINCIPLES AND PROCESS OF GOA

3.1. Principles of GOA

The Grasshopper Optimization Algorithm (GOA) draws inspiration from the behavioral characteristics of locusts across their life stages. By simulating the distinct patterns of locust larvae and adults, GOA achieves a balance between exploration and exploitation, maintaining the breadth of global search while possessing the precision of local search.

This algorithmic design approach offers a novel and effective method for tackling complex optimization problems. Consequently, the mathematical model simulating the collective behavior of locusts in nature is as follows:

$$X_i(t+1) = S_i(t) + G_i(t) + A_t(t) \tag{1}$$

In the $t+1$ iteration,$X_i(t+1)$ represents the new position of the locust. Furthermore, during the iteration, its interactions with other locusts are jointly determined by the interactions $Si(t)$, with other locusts, the gravitational force $G_i(t)$ and the wind force$A_i(t)$.The expressed formula is as follows:

$$S_i(t) = \sum_{j=1\ j\neq i}^{N} s\,(d_{ij}(t))\hat{d}_{ij}(t) \tag{2}$$

In the above equation, j denotes the locust population. At iteration t, $d_{ij}(t)$ represents the distance between locusts i and j; $x_i(t)$ indicates the position of locust i; $x_j(t)$ denotes the position of locust j; s is the interaction function between locusts:

$$S\left(d_{ij}(t)\right) = f\cdot\exp\left(-d_{ij}(t)/l\right) - \exp\left(-d_{ij}(t)\right) \tag{3}$$

f and l denote the attraction strength and range between locusts, respectively, with f=0.5,l=1.5,$d_{ij}(t)$= [1,4]; $\hat{d}_{ij}(t) = \frac{x_j(t)-x_i(t)}{d_{ij}(t)}$. Unit vector representing the distance between locusts of populations i and j. The other two terms in Equation 1 satisfy:

$$G_i(t) = -g\hat{e}_g \tag{4}$$

$$A_i(t) = u\hat{e}_w \tag{5}$$

In the equation: g is the gravitational acceleration constant, $\hat{e}_g$is the unit vector pointing toward the center of the Earth for the locust; u is the drift constant; $\hat{e}_w$is the unit vector representing the wind force acting on the locust. Combining the above equations yields:

$$x_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} s(|x_j - x_i|)\frac{x_j - x_i}{d_{ij}} - g\hat{e}_g + u\hat{e}_w \tag{6}$$

This formula is designed for locust swarms inhabiting free space and cannot be directly applied to solve optimization problems, as the swarm in the model rapidly reaches a comfort zone and fails to converge to the specified point. To enable optimization problem solving, the formula is modified under the assumptions of zero gravitational influence and wind direction always aligned with the current optimal solution, yielding the iterative formula for GOA:

$$x_i^d = c_1\left(\sum_{\substack{j=1 \\ j\neq i}}^{N} c_2 \frac{ub_d - lb_d}{2} s(|x_j^d - x_i^d|)\frac{x_j - x_i}{d_{ij}}\right) + \hat{T}_d \tag{7}$$

Where: d denotes the dimension of the solution space, the number of variables or features to be optimized; ub_d represents the upper bound of the d-th dimension; lb_ddenotes the lower bound of the d-th dimension; x_j^d indicates the position of the i-th locust in the d-th dimension during the algorithm iteration; $\hat{T}_d$denotes the value of dimension d in the optimal solution; c_1 and c_2 represent the control parameters for the attraction and repulsion forces between locusts, respectively. Here, c_1 adjusts the algorithm's convergence process, while c_2 linearly reduces the interaction range between locusts, thereby guiding them toward the optimal solution. Treating c_1 and c_2 as a single parameter c, the calculation formula can be expressed as:

$$c = c_{max} - t\frac{c_{max} - c_{min}}{Max\_iter} \tag{8}$$

Where: t denotes the number of iterations,Max_iter indicates the maximum number of iterations for the algorithm.

3.2. The GOA Process

Configure parameters and generate the initial population: Establish the population size (number of locust individuals N), maximum iteration count (T_{max}), control parameters c_{max} and c_{min} (which balance exploration and exploitation and decrease nonlinearly with iteration count), set the objective function, and initialize the population.

1) Iterative Update Phase

For each locust i, the position update formula is:

$$X_i^t = c.\left(\sum_{j=1, j\neq i}^{N} c \cdot \frac{ub_d - lb_d}{2} \cdot s(|x_j^t - x_i^t|) \cdot \frac{x_j - x_i}{d_{ij}}\right) + T_d \tag{9}$$

Among them:

t : current iteration count

$d_{ij} = |x_j - x_i|$:the Euclidean distance between locusts i and j

T_d : current optimals olution (location with the minimum objective function value)

c : control parameters, decreasing with each iteration:

$$c = c_{max} - t \cdot \frac{c_{max} - c_{min}}{T_{max}} \tag{10}$$

$s(r)$: social interaction force function (typically an attraction-repulsion function):

$$s(r) = fe^{-r/l} - e^{-r} \tag{11}$$

$$f \text{ and } l \text{ are constants } (typically\ f\ =\ 0.5, l\ =\ 1.5)$$

2)Boundary handling

After updating the position, check whether it has exceeded the boundaries. Common methods include:

Reflective Boundary: Reverse movement upon crossing.

3) Clamping boundary: set directly to boundary value.
4) Update the optimal solution Calculate the objective function values of all locusts and update the global optimal solution Capital T subscript d

5) Termination condition Stop when the maximum number of iterations is reached Capital T subscript max or when the convergence condition change of the optimal solution is less than the threshold) is satisfied.

3.3. Multi-Objective Optimization Problems

Multi-objective optimization problems refer to mathematical challenges where multiple conflicting objectives must be optimized simultaneously, yet these objectives cannot be simultaneously maximized through a single solution. Such problems are prevalent in engineering, economics, artificial intelligence, and other fields.

3.4. Enhanced GOA

3.4.1. Introduction of Chaos Strategy

Chaos strategy, also known as chaos theory or chaotic systems, is a nonlinear dynamical theory characterized by erratic periodicity,unpredictability, and computational simplicity. Its expression formula is:

$$y_{k+1}^d = \mu y_k^d (1 - y_k^d) \tag{12}$$

Where: y_{k+1}^d and y_k^d are two new solutions obtained after chaotic computation, μdenotes a random number, y_k^d is the current solution.

Sankalap Arora [1]with Ten distinct chaotic mappings were introduced into GOA and tested across 13 benchmark functions. Results indicate that chaotic mappings enhance GOA performance, prevent GOA from getting stuck in local optima during optimization, and reduce attraction zones, comfort zones, and repulsion zones among locusts, thereby improving GOA's exploration capabilities. Using chaotic maps c_1 or c_2 individually fails to significantly improve GOA performance, but combining c_1 and c_2 achieves a noticeable enhancement in GOA performance.To enhance the global search capability of the Grasshopper Optimization Algorithm (GOA) and prevent it from getting stuck in local optima, WANG et al. [2] proposed an improved strategy. By applying the Chebyshev mapping method to iteratively update and perturb the positions of individuals surrounding the current optimal solution, the population is prompted to regenerate better solutions, thereby effectively increasing population diversity. Compared to the original GOA algorithm, this improvement significantly enhances global exploration capabilities. Additionally, the study employs an adaptive arc function to dynamically adjust the critical parameter c of GOA, enabling the algorithm to better balance the relationship between global search and local exploitation.

The Chebyshev mapping formula is:

$$x_{n+1} = cos\big(k \cdot arccos(x_n)\big)\ , x_n \in [-1,1] \tag{13}$$

k denotes the order of the mapping,x_n represents the current solution, and x_{n+1} denotes the new solution after the mapping.

3.4.2. Introduction of OBL Strategy

Ahmed A. Ewees et al. [3] improved the GOA using the OBL strategy, naming it OBLGOA. To enhance the global exploration capability and accelerate convergence of the Grasshopper Optimization Algorithm (GOA), the Opposition-Based Learning (OBL) strategy is introduced. OBL simultaneously evaluates the current solution and its opposing counterpart, leveraging the symmetry of the search space to improve the algorithm's ability to escape local optima. The specific improvements are divided into three parts:

Initialization Phase: After generating the initial population, compute its opposing population and select the fittest individuals to form the initial solution set, thereby improving solution quality.

Iteration Process: Dynamically generate opposing solutions for selected individuals based on probabilities, retaining superior solutions to balance exploration and exploitation.

Jump Rate Mechanism: Adaptively adjust OBL application frequency according to convergence status.

4. PERFORMANCE TESTING AND ANALYSIS

4.1. Test Functions and Parameter Settings

To comprehensively evaluate the optimization effectiveness of various algorithms, a comparative analysis was conducted between the Locust Algorithm, Particle Swarm Optimization, Multi-Objective Evolutionary Algorithm, and NSGA-II algorithm. Five target test functions (ZDT1, ZDT2, ZDT3, UF8, UF9) were selected for simulation testing. Population size N = 100, maximum iterations T_max = 100. Parameter settings are detailed in the table below:

Table 1. Parameter settings are detailed

Algorithm	Parameter Settings
COMOGOA	$c_{max} = 1$,$c_{min} = 0.0004$
MOGOA	$c_{max} = 1$,$c_{min} = 0.0004$
MOPSO	$w = 0.5$,$c_1 = 2$,$c_2 = 2$,$p_m = 0.1$
MOEA/D	$T = 15$,$p_c = 0.5$
NSGAII	Pc=0.7,pm=0.02

4.2. Test Results and Analysis

For the multi-objective problem at hand, the test function was evaluated using the optimization algorithms listed above to determine the mean and standard deviation of each performance metric. COMOGOA employs the Cubic chaotic map during the early evolutionary phase to initialize the population, enhancing initial diversity and preventing premature convergence to local optima. The nonlinear adaptive coefficient updates parameter c, balancing global exploration and local refinement capabilities, improving the quality of the optimal solution set. Furthermore,the introduction of the Cauchy mutation operator and reverse learning strategy optimized both global search efficiency and the distribution characteristics of the solution set.The IGD test results are presented in Table 2 (bolded values indicate the function's optimal solutions):

Table 2. IGD test results

Test	**Statistical measure**			Algorithm	
Function		**COMOGOA**	**MOGOA**	**MOPSO**	**MOEA/D**
ZDT1	**Mean**	$\mathbf{6.5 \times 10^{-5}}$	1.1×10^{-3}	2.6×10^{-4}	5.3×10^{-4}
	Var	$\mathbf{4.4\times 10^{-10}}$	1.1×10^{-6}	2.4×10^{-8}	3.4×10^{-7}
ZDT2	**Mean**	$\mathbf{5.9\times 10^{-5}}$	2.0×10^{-3}	2.7×10^{-4}	7.9×10^{-3}
	Var	$\mathbf{1.2\times 10^{-10}}$	2.3×10^{-6}	1.0×10^{-8}	6.0×10^{-5}
ZDT3	**Mean**	$\mathbf{2.2\times 10^{-3}}$	2.5×10^{-3}	5.5×10^{-3}	7.1×10^{-3}
	Var	$\mathbf{1.6\times 10^{-9}}$	7.4×10^{-7}	1.2×10^{-6}	5.5×10^{-7}
UF8	**Mean**	$\mathbf{1.6\times 10^{-3}}$	3.2×10^{-3}	2.6×10^{-3}	3.4×10^{-3}
	Var	$\mathbf{8.0\times 10^{-9}}$	4.3×10^{-6}	1.2×10^{-7}	5.5×10^{-6}
UF9	**Mean**	$\mathbf{2.2\times 10^{-3}}$	3.2×10^{-3}	2.1×10^{-3}	2.4×10^{-3}
	Var	$\mathbf{1.1\times 10^{-9}}$	2.9×10^{-6}	3.4×10^{-7}	2.4×10^{-7}

As shown in the table above, COMOGOA exhibits significantly superior convergence properties and a more stable convergence process compared to other algorithms. This enables it to better guide the population toward exploring optimal solutions, thereby enhancing the algorithm's efficiency. Furthermore, the initial population has been optimized to prevent getting stuck in local optima. Moreover, this method employs a nonlinear adaptive strategy to update the parameter c, significantly enhancing the algorithm's adaptive capability between global exploration and local refinement. This approach yields a solution set with superior performance.

5. STABILITY ANALYSIS OF STALL AND SURGE

By plotting the proximity intake angle against relative rotational speed at differentrotational speeds, the following relationship diagram can be obtained:

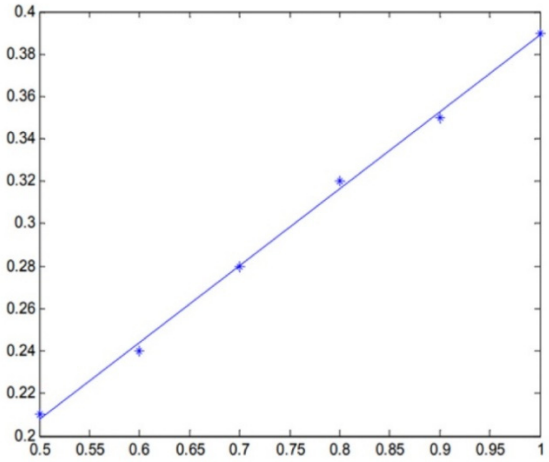

Figure 3. Intake Angle vs. RPM Relationship Chart

When the flow coefficient falls below the boundary value—that is, below the blue line shown in the figure above—the system enters a rotational stall state. This can serve as a monitoring indicator for

rotational stall. By ensuring the actual operating flow coefficient remains above the set minimum flow threshold, the compressor can be prevented from entering a stall state.

5.1. Optimization of Compressor Impeller Parameters

The research subject is a single-stage centrifugal compressor with a semi-open impeller featuring backward-curved blades.

The fixed parameters for the initial optimization are shown in Table 3:

Table 3. Fixed parameter

Parameter	Value
Outlet Diameter D1/mm	260
Inlet Diameter D2/mm	130
Outlet Width B1/mm	42
Outlet Width B2/mm	15
Impeller Speed n/(r/min)	15000
Blade Thickness h/mm	1.62
Blade Inlet Installation Angle β	30

The impeller, as a rotating machine, efficiently converts energy, increasing both the kinetic energy and pressure potential energy of the gas. Based on the infinite blade theory, assuming the flow channel cross-section is extremely small and the gas exits at a predetermined angle, highly efficient and precise energy conversion can be achieved[4].

In actual impellers, gas flow becomes complex, deviating from the blade exit angle of installation. This occurs because high-speed rotation within the flow channel induces counter-rotating axial vortices, a phenomenon explained by gas inertia[5]. The reverse vortex induces a circumferential component Δω2u in the gas velocity relative to the impeller outlet. Its magnitude is calculated by substituting the impeller rotational speed for the vortex speed and the impeller outlet width for the vortex diameter [6].

The COMOGOA method was employed to optimize the impeller parameters.

Table 4. Parameter setting for each algorithm

Algorithm	Parameter Settings
COMOGOA	X_0=0.3, ρ=2.595, β=0.05, n=30,d=5,t=30,iter=100
MOGOA	c_{max}=1,c_{min}=0.000 4,n=30, d=5, t=30 ,iter=100
MOPSO	w=1.5,c1=2,c2=2,pm=0.1 n=30,d=5,t=30,iter=100
MOEA/D	T=15, pm=0.1,n=30,d=5, t=30,iter=100
NSGAII	Pc=0.7,pm=0.02,c1=2,n=30 d=5,t=30,iter=100

The optimization variables selected were the outlet pitch angle β A and the number of blades Z, while other design variables remained constant throughout the optimization process. Based on the principle of minimizing energy loss, the energy head and energy loss were chosen as optimization objectives. Constraints were set with the outlet pitch angle ranging from [30°, 60°]and the number of blades from [14, 28] as appropriate [7].

Selected optimization algorithms commonly used for impeller design optimization MOGOA, MOPSO, MOEA/D, and NSGA-II were compared for their optimization capabilities [3].

The parameter settings for each algorithm are shown in Table 4.

Based on the objective function and constraints for the known impeller optimization design, the author applied the five optimization algorithms mentioned above to perform optimization separately. The optimization results from each algorithm were compared and analyzed, with the results shown in Table 5.

Table 5. Comparison of optimization results by algorithm

Algorithm	β2A/ (°)	z	Objective function f1/m	Objective function f2/m
NSGAII	47	19	6.42×10^5	7.67×10^3
MOEA/D	46.5	21	6.21×10^5	7.91×10^3
MOPSO	48	17	6.61×10^5	7.47×10^3
MOGOA	42	18	6.62×10^5	7.45×10^3
COMOGOA	46	20	6.99×10^5	7.08×10^3

As shown in Table 5, the objective function values f_2 of each algorithm, arranged from smallest to largest, are: COMOGOA < MOGOA < MOPSO < NSGA-II < MOEA/D.

Under the principle of ensuring that the objective function value f1 does not fall below the initial design value, the smaller the value of objective function f_2, the better the optimization results of the algorithm. Therefore, it can be concluded that the optimization results obtained by COMOGOA are superior to those of other algorithms.

This paper addresses the critical stability issue of stall and surge in centrifugal compressors by proposing an intelligent control strategy based on the Grasshopper Optimization Algorithm (GOA). By establishing a nonlinear dynamic model of the compressor and integrating advanced control methods such as fuzzy PID or model predictive control, the GOA algorithm is systematically incorporated into control parameter optimization. This approach achieves efficient, adaptive tuning of key controller parameters. Simulation and experimental results demonstrate that this strategy significantly expands stability margins, suppresses pressure pulsations, effectively delays the surge boundary, and enhances the system's dynamic response and disturbance rejection capabilities under disturbed operating conditions.

Compared to traditional PID control and empirical tuning methods, the GOA-optimized control strategy exhibits superior convergence and stability. Particularly in multi-objective optimization tests, improved GOA variants significantly outperform MOGOA, MOPSO, and MOEA/D algorithms in IGD metrics, demonstrating enhanced balance between global exploration and local exploitation. By incorporating strategies like chaotic mapping and Opponent Learning (OBL), the algorithm gains enhanced diversity and improved ability to escape local optima, offering novel insights for optimizing complex industrial systems.

This research not only delivers a practical intelligent optimization solution for anti-surge control in centrifugal compressors but also provides a reference for solving other high-dimensional, nonlinear, multi-objective engineering optimization problems. Future work may explore the integration of GOA with other intelligent algorithms, combining real-time data-driven approaches and deep learning techniques to achieve more precise operating condition identification and adaptive control. This will advance compressor systems toward higher levels of intelligence, reliability, and energy efficiency.

REFERENCES

[1] LAN Ya-xun. Grasshopper optimization algorithm based on chaos and Cauchy mutation and feature selection [J]. Microelectronics and computer,2021,38(11):21-30.

[2] MAO Qing-hua, ZHANG Qiang. Improved sparrow algorithm combining Cauchy mutation and opposition-based learning[J]. Computer Science and Exploration,2021,15(6):1155-1164.

[3] WANG Bo,LIU Lian-sheng,HAN Shao-cheng, et al.Hybrid multi-objective grasshopper optimization algorithm based onfusion of multiple strategies[J].computer application,2020,40(9): 2670-2676.

[4] LI J, LI G, FENG Z, et al. Multi-objective Optimization Approach to Turbomachinery Blades Design[C]//ASME Turbo Expo 2005: Power for Land, Sea, and Air,2005:1051-1058.

[5] ZHANG Ao.Blade Parameterization and Aerodynamic Performance Optimization of Centrifugal Compressor Impellers[D]. Dalian: School of Power Engineering, Dalian University of Technology, 2022.

[6] ZHONG Hao. Introduction and Mastery of Centrifugal Compressors[M]. Beijing: China Machine Press,2015.

[7] XU Zhong. Principle of Centrifugal Compressor[M].Beijing: China Machine Press,1990.