Dean&Francis

ISSN 2959-6157

Intelligent Optimization of PID Controller Parameters Using Enhanced Parallel Genetic Algorithm

Jiajun Gu

Institut Hohai-Lille, Hohai University, Nanjing, Jiangsu, 210024, China
jiajungu@hhu.edu.cn

Abstract:

Traditional Proportional-Integral-Derivative (PID) parameter tuning often employs the Ziegler-Nichols method, but it suffers from limitations such as reliance on experience and trial-and-error. With the development of intelligent algorithms, although the Simple Genetic Algorithm (SGA) can achieve automatic parameter adjustment, it still faces issues like insufficient global search capability and premature convergence. This paper proposes an Enhanced Parallel Genetic Algorithm (EPGA), which constructs a parallel evolutionary architecture and a multi-dimensional performance index system. By integrating dynamic tournament selection, adaptive mutation, and periodic elite migration mechanisms, and designing a diversity reward function, EPGA effectively balances exploration and convergence capabilities. Simulation experiments show that for a typical first-order inertial system with time delay, the PID controller optimized by EPGA can achieve no overshoot dynamic response, fast stability, and precise steady-state control performance. Compared with traditional genetic algorithms and empirical tuning methods, this optimization strategy significantly improves control quality indicators. The study indicates that EPGA suppresses premature convergence by constructing a parallel population topology network and accelerates the propagation efficiency of excellent parameters by combining dynamic gene selection mechanisms, demonstrating excellent global search capability and convergence characteristics in control system parameter optimization. This method provides a new solution for high-precision control requirements in complex industrial scenarios, balancing dynamic response quality and system robustness.

Keywords: PID control; parameter optimization; genetic algorithm; intelligent control

1. Introduction

In the industrial automation control system, the stable and reliable operation of the control system is fundamental to ensuring production efficiency and product quality. As the most widely used core control strategy in industrial process control, the Proportional-Integral-Derivative (PID) controller's parameter tuning quality directly determines the dynamic response characteristics and robustness of the control system. Although the classic Ziegler-Nichols (Z-N) method can quickly generate initial parameters, its empirical tuning process, relying on critical gain measurement, often leads to significant system overshoot and difficult elimination of steady-state errors, exposing inherent limitations in precision control scenarios [1]. With the increase in industrial system complexity and the upgrade of control precision requirements, traditional trial-and-error parameter tuning methods can no longer meet multi-objective optimization needs. How to break through the constraints of experience dependence and local optimality to achieve high-precision adaptive optimization of PID parameters has become a key scientific issue in improving the performance of industrial control systems [2].

In recent years, intelligent optimization technologies represented by genetic algorithms have provided new ideas for PID parameter tuning. The Simple Genetic Algorithm (SGA) realizes automatic parameter optimization by simulating biological evolution mechanisms [3]. However, its single-population architecture is prone to falling into local optimality, and the problem of limited global search capability is particularly prominent in complex system optimization. The Parallel Genetic Algorithm (PGA) adopts a multi-subgroup collaborative evolution strategy, and the migration mechanism enhances population diversity, providing possibilities for breaking through the premature convergence bottleneck [4]. However, existing PGA algorithms still have shortcomings such as single information interaction mechanisms between subgroups, lack of dynamic adaptability in elite migration strategies, insufficient consideration of the multi-objective optimization characteristics of control systems in the design of selection and mutation operators, and the lack of a diversity maintenance mechanism for the PID parameter space. These defects make it difficult to balance the algorithm's convergence speed and solution set quality, restricting its application potential in real-time control systems. Therefore, constructing an enhanced parallel genetic algorithm that integrates dynamic evolution strategies is of great theoretical value and engineering significance for achieving global optimization of PID parameters.

This study proposes an Enhanced Parallel Genetic Algorithm (EPGA), which uses an island model to construct a parallel architecture and balances global exploration and local development through periodic elite migration and dynamic tournament selection. A hybrid crossover mutation operator is designed, and the parameter search capability is enhanced by combining uniform crossover and adaptive three-mode mutation strategies. A weighted evaluation system including overshoot, setting time, and other parameters is established to guide the efficient tuning of PID parameters.

2. Methods

2.1 Genetic Algorithm (GA) Algorithm Framework Design

2.1.1 PID control principle

The PID controller is a classic feedback controller, and its name comes from the three regulatory actions of Proportional, Integral, and Derivative. The proportional term generates a control amount in proportion to the current error size, helping to accelerate the system response; the integral term accumulates historical errors to eliminate steady-state deviations; the derivative term generates a control amount according to the error change rate, which can suppress overshoot and improve system stability. The typical PID control law can be expressed as:

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt} \tag{1}$$

Where $u(t)$ is the controller output control amount, K_p is the proportional coefficient, $e(\tau)$ is the control error at time, K_i is the integral coefficient, K_d is the derivative coefficient, and $\frac{de(t)}{dt}$ is the rate of change of the error with respect to time t.

The PID control has a simple structure and strong robustness, and is widely used in industrial processes, robotics, and other fields. By reasonably tuning the three parameters, the system can be balanced to achieve a balance between its rapidity, stability, and steady-state accuracy (as shown in Fig. 1).

ISSN 2959-6157

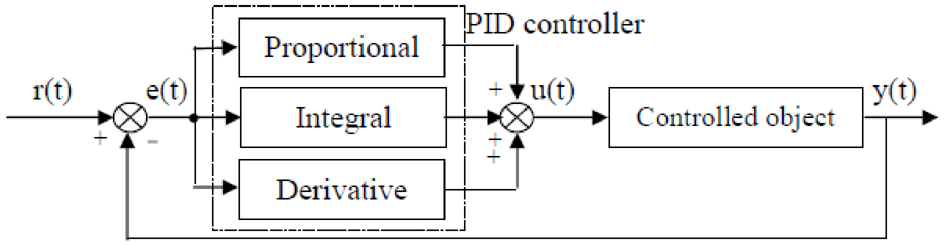

Fig. 1. The structure of the PID controller [5].

2.1.2 Theoretical basis of genetic algorithm

The Genetic Algorithm (GA) was first proposed by J. Holland and others in the 1970s. It is a global optimization method that simulates the natural evolution process. The algorithm maps the problem's solution to a chromosome-type individual and searches for the optimal solution through population evolution [6]. Its basic process is as follows: first, initialize the population, randomly generate an initial population of a certain scale, and each individual is encoded to represent a candidate solution; then carry out fitness evaluation, calculate the fitness of each individual according to the preset objective function, and measure the quality of the candidate solution. To solve the multi-objective optimization problem in engineering, a fitness function containing multiple variables is generally constructed [7]; then set the selection operator, carry out survival of the fittest on the population according to the fitness, and select individuals with high fitness to enter the next generation; then set the crossover operator, perform gene recombination operations on the selected parent individuals to generate new individuals to mix the genetic information of the parents; then set the mutation operator, perform random gene mutations on individuals with a certain probability to introduce diversity and prevent the population from falling into local optimality; finally, carry out evolutionary iteration. After generating the offspring, form a new generation of the population, and repeat the above selection, crossover, and mutation processes until the termination criterion is met to end the search.

Through simulating the survival of the fittest and gene mutation mechanisms of biology, the genetic algorithm realizes the global search capability for high-dimensional solution spaces with few parameter assumptions. It has been widely applied to control system parameter optimization, machine learning, engineering optimization, and other fields, and is particularly common in PID parameter tuning, which can automatically optimize controller parameters to meet multi-objective performance requirements [8].

2.2 Enhanced Parallel Genetic Algorithm

To improve the search efficiency and global performance of PID parameter optimization, this study proposes an Enhanced Parallel Genetic Algorithm (its algorithm framework is shown in Fig. 2).

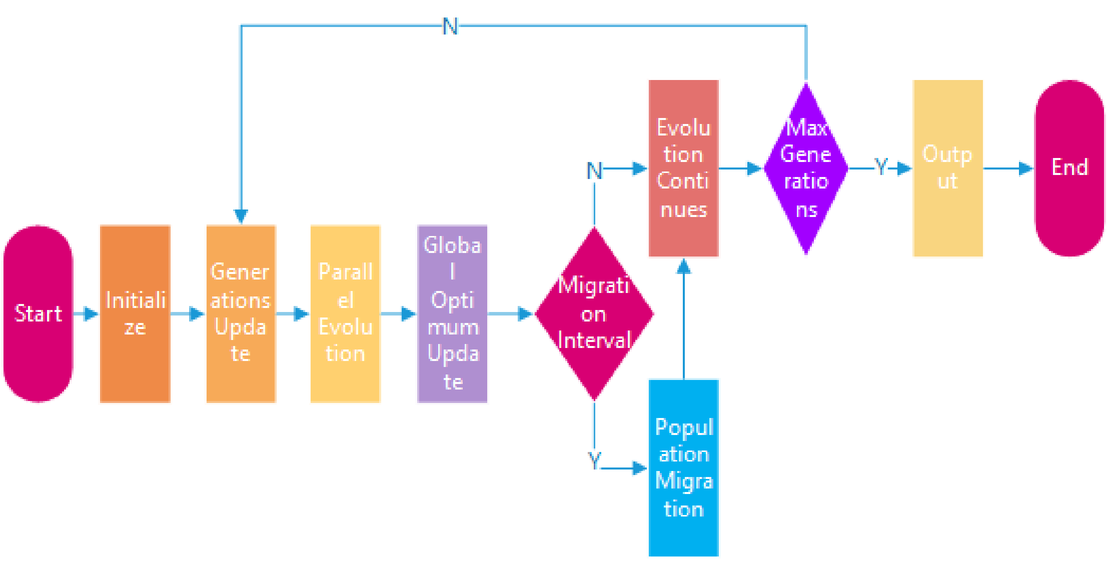

Fig. 2. The structure of the autoencoder (Photo/Picture credit: Original).

2.2.1 Parallel evolutionary architecture

The entire population is divided into multiple subpopulations (islands), and each subpopulation evolves in parallel on independent computing units. The number and scale of subpopulations are specified by parameters. Each subpopulation performs selection, crossover, and mutation operations similar to traditional GA internally, and significantly improves evolution efficiency through parallel computing. Each subpopulation regularly exchanges excellent individuals according to the strategy after each generation to achieve global information sharing and collaborative search.

2.2.2 Dynamic tournament selection mechanism

This paper designs an enhanced tournament selection operator. As the number of generations increases, the tournament scale dynamically increases to enhance the selection pressure in the later stage.

$$tournamentSize = basesize + \left[\left(\frac{gen}{\max Gen} \right)^{1.5} \times \left(\max Size - baseSize \right) \right]$$

Where $basesize$ is the base tournament scale, $\max Size$ is the maximum tournament scale, gen is the current number of evolutionary generations, and $\max Gen$ is the maximum number of evolutionary generations.

In addition, the diversity is quantified by calculating the average value of the population parameter standard deviation, and the diversity reward value multiplied by the coefficient is added to or subtracted from the individual fitness to encourage the retention of individuals with high diversity. In each round of selection, the individual with the highest fitness in the current subpopulation is always forced into the candidate pool to ensure that excellent genes are not lost (as shown in Fig. 3). Finally, a tournament is carried out based on the adjusted fitness, and the winner is selected from the candidate pool to enter the next generation. This mechanism effectively maintains population diversity while maintaining convergence speed.

Fig. 3. Construction of Candidate Pool (Photo/Picture credit: Original).

2.2.3 Uniform crossover operator and its bit operation logic

EPGA uses a binary uniform crossover operator to generate offspring, and each parameter undergoes gene crossover independently. For each pair of parents, a crossover mask is randomly generated for the binary encoding bits of each parameter, and the gene bits corresponding to the position of 1 are exchanged between parents and offspring. In implementation, uniform crossover is performed for each parameter separately: first, the parent parameter values are binary encoded, then the encoding bits are exchanged using a random mask, and finally, the exchanged binary sequence is decoded into a real number to obtain a new individual. Compared with single-point crossover, uniform crossover does not require a fixed split point and can perform more fine-grained mixing of chromosomes, thereby improving the diversity of offspring (as shown in Fig. 4).

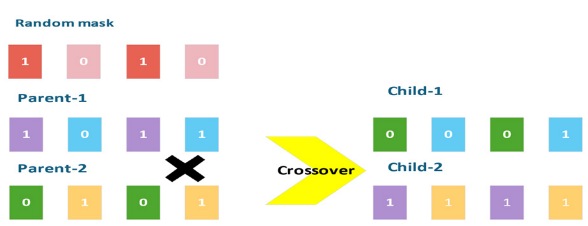

Fig. 4. Uniform Crossover (Photo/Picture credit: Original).

2.2.4 Adaptive multi-mode mutation strategy

EPGA sets an adaptive mutation probability for each individual's parameters and randomly switches between three mutation modes. Specifically, the mutation probability is dynamically adjusted according to the current number of evolution generations and the parameter value range. When actual mutation occurs, it is determined whether to trigger the mutation with an adaptive probability, and then a mode is randomly selected to execute. The parameter

ISSN 2959-6157

value can be encoded as a binary string and randomly flip the encoding bits, which is the binary bit flip mode. In this mode, there is a 30% probability of performing multiple random flips (selecting 1 to 1/4 of the total number of bits to flip), and in other cases, only one random position is flipped; a new value can also be randomly generated in a uniform distribution within the parameter allowable range, that is, the uniform replacement mode. This mode allows the algorithm to adaptively switch between global search (large-scale perturbation) and local fine-tuning (small-scale perturbation), helping to improve search flexibility and global optimization capabilities; Gaussian noise with a mean of 0 and a standard deviation proportional to the parameter range can also be superimposed on the parameter value, that is, the Gaussian perturbation mode.

2.2.5 Elite migration mechanism and diversity maintenance

In the multi-subgroup parallel framework, a periodic elite migration mechanism is introduced to enhance global search. Migration is performed every few generations, and each subpopulation selects the top 15% of individuals with the highest fitness as elite immigrants. A fully connected topology is adopted, each subpopulation transmits elites to all other subpopulations, and each connection migrates at most 2 individuals. At the receiving end, according to the number of received immigrants, the individuals with the worst fitness in the current subpopulation are replaced with these elite immigrants, thereby realizing the cross-group diffusion of excellent genes. After migration, to prevent the decline of population diversity, about 5% of new random individuals are injected into each subpopulation to replace the remaining worst individuals (as shown in Fig. 5). The elite migration and diversity maintenance mechanism can effectively suppress the premature convergence phenomenon while accelerating global convergence [9].

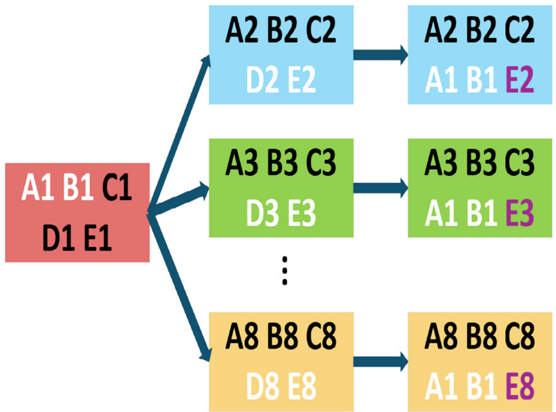

Fig. 5. Fully-Connected Topology (Photo/Picture credit: Original).

2.3 Control Object Modeling and Simulation Configuration

2.3.1 Mathematical model of the controlled system

The controlled object used in the simulation is a first-order inertial system with time delay, and its transfer function model is $\frac{2e^{-2s}}{5s+1}$. This model has typical first-order inertia and 2-second pure time delay characteristics, which is representative of PID parameter tuning.

2.3.2 Simulation environment and evaluation system

This paper uses the MATLAB simulation platform to construct a closed-loop system and evaluate control performance. The specific approach is to use the pid function and feedback function in the Control System Toolbox to connect the candidate PID parameters with the controlled object model to form a unit negative feedback closed-loop system. A unit step input is applied to the closed-loop system, and the system response curve is obtained through the step function (simulation time is 0–30 seconds, step size 0.01s).

In terms of performance evaluation, key time-domain indicators of the closed-loop step response are extracted: overshoot, settling time, rise time, etc.; at the same time, error integral indicators are calculated: Integral Squared Error (ISE), Integral Absolute Error (IAE), and Integral Time Weighted Absolute Error (ITAE). The above indicators are weighted and combined according to preset weights to form a comprehensive performance evaluation value (fitness). In the algorithm iteration, the search is carried out with the principle that the smaller the comprehensive index, the better, to ensure that the PID parameters with good response speed and stability are obtained.

2.4 Experimental Design and Comparison Scheme

To verify the performance advantages of EPGA, this study designed a comparative experiment, comparing EPGA with the standard SGA and the classic Ziegler–Nichols (Z–N) empirical tuning method. The experimental settings are that all genetic algorithms use the same parameter configuration (total population size 400, number of subpopulations 8, maximum number of iterations 100, migration interval 5 generations, parameter binary encoding length 20 bits, PID parameter range ($Kp \in [0,10]$, $Ki \in [0,1]$, $Kd \in [0,1]$). The EPGA and SGA algorithms are independently run multiple times to eliminate the influence of randomness; each run records the change trend of the global optimal fitness value during the iteration, the finally obtained optimal PID parameters, and the optimization time consumption. In the control scheme, the SGA runs in a serial mode, and the Z–N method uses the MATLAB built-in tuning tool for parameter setting.

The evaluation indicators include convergence speed (the change curve of fitness with the number of generations), total optimization time, and the performance indicators of the final closed-loop system. A self-compiled comparison function is used to simulate the PID parameters obtained by the three methods, draw the response curve, and calculate the response index. Finally, the improvement effect of EPGA relative to SGA and Z–N methods in control performance and search efficiency is evaluated through numerical comparison and graphic analysis.

3. Experimental Results

In the step response diagram (Fig. 6), the system optimized by EPGA (blue solid line) has almost no overshoot, can quickly track the target and stably converge; in contrast, the SGA (red dashed line) response has obvious overshoot and stabilizes slowly, while the classic Z-N tuning (green dotted line) has the largest overshoot.

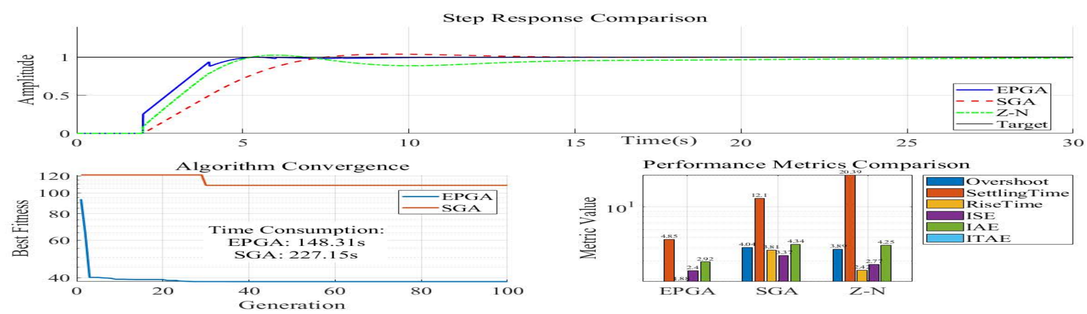

Fig. 6. Response Curves of EPGA/SGA/Z-N Methods (Photo/Picture credit: Original).

The convergence curve (Fig. 7) shows that the optimal fitness value of EPGA decreases rapidly and tends to be stable within the first several dozen generations, and finally reaches a level significantly lower than that of SGA, while the total optimization time is also significantly reduced.

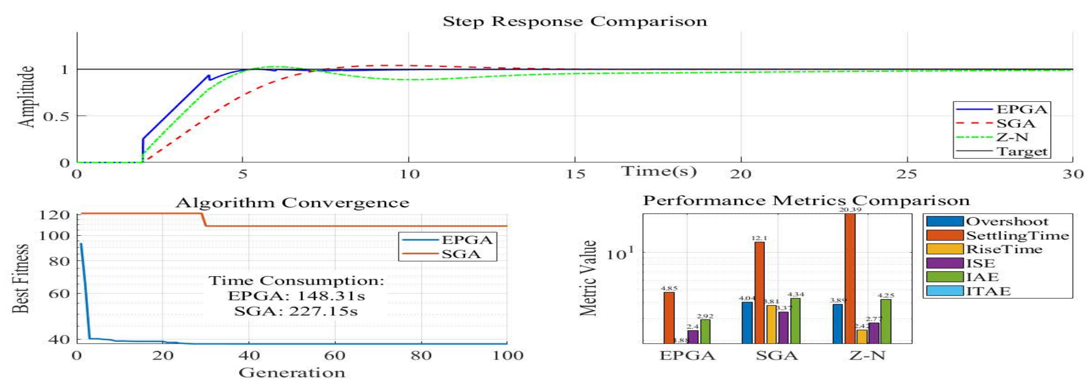

Fig. 7. Changes in Fitness Function and Optimization Time Consumption of EPGA/SGA (Photo/Picture credit: Original).

The performance index histogram (Fig. 8) shows the overshoot, setting time, rise time, and ISE/IAE/ITAE errors of EPGA, SGA, and Z-N together. It can be seen that EPGA is better than the control scheme in all indicators: its overshoot is only about 0.02%, significantly smaller than SGA (about 7.6%) and Z-N (about 20.4%), the setting time and rise time are also the shortest, and the ISE/IAE/ITAE values are significantly lower, indicating that the steady-state accuracy and robustness are greatly improved (the specific results are shown in Table 1). The comprehensive simulation results show that the EPGA optimization strategy effectively improves the dynamic response quality of the control system and realizes the control performance of no overshoot and fast stability.

ISSN 2959-6157

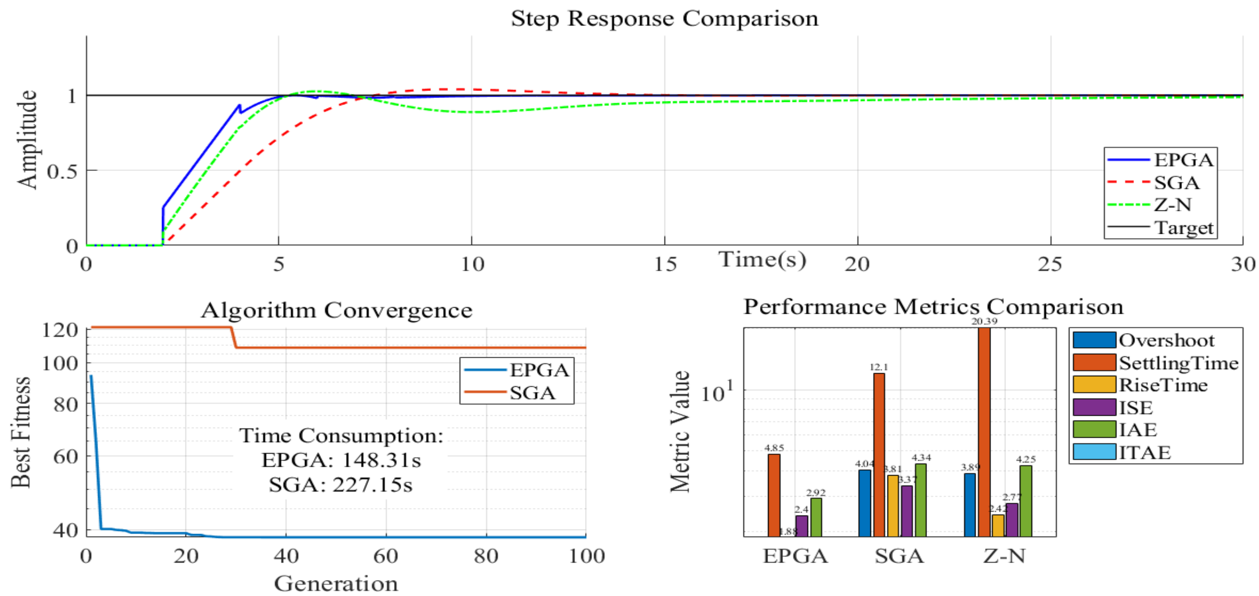

Fig. 8. Comparison of Performance Bar Charts (Photo/Picture credit: Original).

Table 1 shows the performance parameter comparison of EPGA/SGA/Z-N methods.

Table 1. Performance Parameter Comparison of EPGA/SGA/Z-N Methods

Performance Indicators	EPGA	SGA	Z-N
Overshoot σ(%)	0.02	0.84	3.89
Setting Time $t_s(s)$	5.99	7.63	20.39
Rise Time $t_r(s)$	1.84	3.94	2.42
ISE	2.38	2.89	3.77
IAE	2.88	3.79	4.25
ITAE	0	0	0

4. Discussion

The comprehensive experimental results analysis shows that EPGA has significant advantages over the traditional SGA and the empirical Z-N method. Its performance improvement mainly comes from the enhancement of global search capability by the multi-subpopulation parallel architecture and elite migration mechanism in the algorithm, the maintenance of population diversity by the dynamic tournament selection and adaptive multi-mode mutation strategy, the improvement of gene combination richness by the uniform crossover operator, and the guidance of the optimization direction by the objective function integrating multi-dimensional performance indicators. However, the algorithm still has limitations such as high computational resource requirements (the parallel architecture and fully connected migration have strict requirements on hardware computing power), the dynamic adjustment of hyperparameters relying on experience (such as the mutation probability decay coefficient needs to be manually set), and the adaptability to high-order nonlinear systems to be verified. In the future, the engineering applicability and complex system optimization capabilities of the algorithm can be further improved by introducing a hierarchical migration topology to reduce communication overhead, constructing a meta-learning-based hyperparameter automatic adjustment module, and incorporating anti-interference performance and parameter robustness into the multi-objective optimization framework [10, 11].

5. Conclusion

This paper proposes an Enhanced Parallel Genetic Algorithm (EPGA) applied to PID parameter tuning and verifies it through simulation experiments on a typical first-order inertial system with time delay. The designed

EPGA combines multiple innovative mechanisms such as dynamic tournament selection, adaptive multi-mode mutation, uniform crossover, and periodic elite migration, effectively improving the search capability and population diversity of genetic algorithms. The experimental results show that the PID controller optimized by EPGA has dynamic response performance of no overshoot, fast stability, and excellent steady-state accuracy, and its performance indicators (overshoot, settling time, integral error, etc.) are significantly better than the traditional SGA optimization scheme and the classic Ziegler-Nichols method. The study proves that EPGA, through the combination of parallel subgroup evolution and adaptive operators, effectively avoids premature algorithm convergence and can provide an efficient solution for parameter tuning of precision control systems that balances response quality and robustness. Future work can further verify the applicability of EPGA in higher-order and nonlinear systems and explore the algorithm's extended application in real-time online tuning and multi-objective optimization problems.

References

[1] Ziegler J. B., Nichols N. B. Optimum settings for automatic controllers. InTech, 1995, 64(6): 759-768.
[2] Yuan X. Achieving stable trajectory tracking in complex environments using an adaptive PID control strategy-based quadcopter drone. Applied and Computational Engineering, 2024, 1.
[3] Moravec J. Using simple genetic algorithm for a hand contour classification: an experimental study. 2020.
[4] Tang J., Liu D., Wang Q., Li J., Sun J. Probabilistic chain-enhanced parallel genetic algorithm for UAV reconnaissance task assignment. Drones, 2024.
[5] Zhang S. S., Wang X. B., Wang W. H. Implementation of multi-core parallel computation for real-time tuning of PID parameters based on genetic algorithm. Applied Mechanics and Materials, 2012, 159: 132-136.
[6] Holland J. H. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. Beijing: Higher Education Press, 2008.
[7] Liu T. K., Chang H. Y., Wu W. P., Chen C. H. Evaluated preference genetic algorithm and its engineering applications. Key Engineering Materials, 2011.
[8] Mitsukura Y., Yamamoto T., Kaneda M. A genetic tuning algorithm of PID parameters. Proceedings of 1997 IEEE International Conference on Systems, Man, and Cybernetics, Orlando, FL, USA, 1997.
[9] Zhu P., Sheng Z., Bao J., Li J. Antenna selection for full-duplex distributed massive MIMO via the elite preservation genetic algorithm. IEEE Communications Letters, 2022.
[10] Wu M., Sun D., Cui Y., Gao S., Liu X., Buyya R. A state lossless scheduling strategy in distributed stream computing systems. Journal of Network and Computer Applications, 2022, 206: 103462.
[11] Vettoruzzo A., Bouguelia M. R., Vanschoren J., Rgnvaldsson T., Santosh K. C. Advances and challenges in meta-learning: a technical review. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024, 46(7): 17.