Parameter Tuning of a PID Controller Based on the Cellular Genetic Algorithm

Lifeng Chen

***Abstract*—The PID controller is the most widely used controller, and the performance of each controller depends on its three gain parameters. The genetic algorithm (GA) is an optimization process that involves three types of operators: selection, crossover, and mutation. Inspired by natural evolution, a GA searches for the optimum solution automatically from a collection of potential solutions. The GA was employed for the optimization of PID controller parameters in this paper; this algorithm evolves from generation to generation until the stopping criterion is met or the optimal solution is found. The traditional GA has several shortcomings, such as the great influence of the best individual on the whole population, poor global search ability, and premature convergence; therefore, an improved cellular genetic algorithm (CGA) was proposed whose genetic operators occur between the individual and its neighbors. CGA avoids the adverse influence of the best individual and the possibility of falling into a local optimum. The simulation results show that CGA outperforms traditional GA. CGA is more efficient than GA in global search activities, and its optimum solution is better than that of GA.**

***Index Terms*—Genetic algorithm, Cellular genetic algorithm, Premature convergence, Global search ability**

I. INTRODUCTION

The PID controller is robust and simple in structure and is widely used in the chemical, electricity, automation, military, and aerospace fields, among others. The performance of the PID controller depends on the values of its three parameters, K_p, K_i, and K_d. The classic approaches used for PID parameter tuning include theoretical methods and engineering methods. In theoretical methods, the optimal controller parameters are obtained by establishing the system model and calculating its response. Engineering methods used include the Ziegler–Nichols method and Cohen–Coon method. Classic tuning methods require operators with expert experience, and their application process is complicated and time-consuming.

Artificial intelligence technology developed rapidly in the 1960s and was gradually introduced into many fields. Yan-e Hou proposed a hybrid particle swarm optimization algorithm to achieve efficient task scheduling for a remote sensing product production system. The experimental results demonstrated that the proposed algorithm was effective and more competitive than the existing scheduling algorithms [1]. Gonggui Chen proposed a modified hybrid flower pollination algorithm (MHFPA) to address conflicting objectives. The simulation results showed that the MHFPA had a competitive advantage over other algorithms in handling different scales and nonconvex optimization problems [2]. Jie Qian proposed an improved multigoal particle swarm optimization (IMPSO) algorithm to achieve the optimal operation of a power system. The experiments showed that the IMPSO algorithm was beneficial in realizing the optimal operation of the power system with improved economy and safety [3]. Gonggui Chen proposed an improved marine predator algorithm (IMPA) to solve the short-term hydrothermal scheduling problem. The experimental results showed that the IMPA achieved better results than other methods [4]. Gonggui Chen proposed a novel multiobjective tree seed algorithm (MONTSA) to overcome the deficiency of the tree seed algorithm (TSA) in solving environmental economic power dispatch (EED) problems with high-dimensional nonlinear and nonconvex characteristics. The experimental results showed that MONTSA was more effective at addressing the EED problem than TSA [5]. Inspired by natural selection and natural genetics, genetic algorithms (GAs) involves moving from one population of “chromosomes” to a new population by using a kind of “natural selection” together with the genetics-inspired operators of selection, crossover, and mutation [6]. GAs exhibits global search ability, simplicity, robustness, and adaptability to parallel processing, and it has been widely used in many fields. In [7], Chunlu Wang proposed a multiobjective genetic algorithm for dynamic bus vehicle scheduling under traffic congestion. Jaume Jordan implemented a GA with graph crossover for the localization of charging stations for electric vehicles; this algorithm was shown to outperform other algorithms [8]. Feng Chen implemented an adaptive GA to adjust the sensor acquisition frequency, which reduced the system energy consumption [9]. Olateju Samuel Olaniyi proposed a variant genetic algorithm (VGA) to solve vehicle routing problems with time windows and split delivery. The simulation results showed that VGA obtained better results than traditional methods [10]. Lina Wang proposed a fuzzy neural network temperature prediction method using a GA. The experimental results showed that this method achieved good prediction accuracy [11]. GAs have also been widely used in the field of PID controller parameter tuning. Md. Manjurul Gani applied a GA to tune the PID controller parameters of a furnace temperature control system, which improved the control accuracy and rapidity of the system [12]. A. Ruby Meena applied a GA to tune the fuzzy PID controller parameters of a two-zone reheating thermoelectric system, which improved the system response speed [13]. Guanzheng Tan proposed an optimal PID controller with incomplete derivation based on fuzzy inference and a GA; this controller was used to control the D.C. motor of the intelligent bionic artificial leg designed

Manuscript received July 31, 2023; revised February 26, 2024.

Lifeng Chen is a senior engineer of Experimental Center, Jining University, Qufu 273155, China (e-mail: columbus1980@126.com)

by the author [14]. Fulu Cao proposed a PID controller optimized by a GA for an electrohydraulic servo system directly driven by a permanent magnet synchronous motor, effectively improving the control accuracy and reducing the adjustment time and reduced steady-state error [15]. Mingjian Zhou used a GA to adjust the parameters of the PID controller of a surge compensation system and obtained better anti-interference ability and system robustness [16]. The fusion of GAs with other algorithms has also been widely researched. Xiaogen Shao combined a GA with an interval algorithm and applied it to PID controller parameter optimization, which improved the convergence speed, overcame the premature phenomenon, and reduced the influence of a random initial population on the model [17]. SP Ghoshal optimized the PID controller gains through a genetic algorithm (GA) and a genetic algorithm-simulated annealing algorithm (SA) to obtain an optimal PID controller. The simulation results proved that the GA-SA method was superior to the GA method [18].

When using a GA, excellent individuals have a greater chance of survival than weaker individuals, and they greatly influence the whole population, which will cause the GA to fall into a local optimum and lead to premature phenomenon. To address the premature phenomenon of GAs, in this paper, a genetic algorithm is designed based on cellular automata (CGA), which distributes individuals via the genetic algorithm in two dimensions, and introduces the concept of neighbors. Genetic operations such as selection and crossover of CGAs occur among neighbors; thus, each excellent individual can affect only a small, nearby area, which ensures the best individual cannot influence the whole population and avoids premature convergence. The simulation results show that the multipeak searching characteristic of the CGA overcomes the shortcoming that GAs may easily fall into a local optimum, and the optimum solution of the CGA is much better than that of the GA.

II. PREPARATORY KNOWLEDGE

A. PID controller

A PID controller, which is based on feedback theory, is the most widely used control method. The controller involves three types of control actions: proportional, integral, and derivative operations. The proportional term is proportional to the current control error, the integral term is proportional to the integral of the control error, and the derivative term is proportional to the derivative of the control error [19]. The working mechanism of the PID controller is shown in Fig. 1.

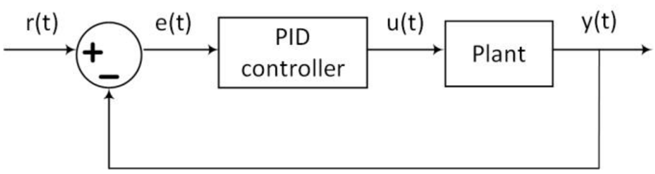

Fig. 1 Block diagram of the PID control system

The output of the PID controller u (t) can be expressed as Eq. 1. K_p, K_i=K_p/T_i, and K_d=K_p·T_d are the proportional gain, integral gain, and differential gain, respectively. A higher proportional gain will increase the speed of response; however, this increase typically comes at the cost of a sharp transient overshoot. Integral control aims to minimize the steady-state tracking error and minimize the steady-state output response to disturbances. The derivative control "knows" the slope of the error signal, so it is said to have an "anticipatory" behavior. Derivative control aims to speed up the transient, reduce the overshoot, and provide a sharp response to suddenly changing signals. Properly adjusting the proportional, integral, and derivative coefficients of the PID controller can lead to better control system performance in terms of the stability, rapidity, and accuracy [20].

$$u(t) = K_p[e(t) + \frac{1}{T_i} \cdot \int_0^t e(t)dt + T_d \cdot \frac{de(t)}{dt}] \tag{1}$$

B. Standard genetic algorithm

(1) Overview

The GA is an optimization method based on the mechanism of biological evolution. The simplest form of the GA involves three types of operators: selection, crossover, and mutation. The selection operator selects the fitter chromosomes in the population for reproduction; the crossover operator chooses a random locus and exchanges the subsequences before and after the locus between two chromosomes to create two offspring; and the mutation operator randomly flips some bits in a chromosome [6]. The fittest individual is obtained through evolution from generation to generation. The process of the general GA is as follows:

1) Start with a randomly generated population of N L-bit chromosomes.

2) Decode each chromosome, and calculate the fitness f(x) of each chromosome. The fitness f(x) represents the individual's ability to survive in the environment.

3) Execute the following GA operators: selection, crossover, and mutation.

4) Replace the current population with the new population generated in step 3.

5) Repeat steps 3 and 4 until the fittest chromosome is obtained.

(2) Encoding and decoding

The coding system is constructed to maintain a fixed structure within the chromosome that allows similar alleles to compete against each other at a locus during evolution [21]. The simple GA encoding includes binary encoding, real encoding, and other encoding. Binary encoding was adopted in this paper. Each chromosome is a 30-bit binary string, and the three parameters K_p, K_i, and K_d correspond to 10-bit binary strings. The decoding is constructed to obtain the values of encoded parameters such as K_p, K_i, and K_d. If the value range of K_p is [a, b] and the decoded value is S, then the value of K_p is as follows:

$$K_{\mathrm{p}} = a + \frac{S}{2^{10} - 1} \cdot (b - a) \tag{2}$$

For example, one chromosome's binary coding is 111110110110001110101110110111. The parameter K_p responds to the first 10-bit binary strings, the parameter K_i responds to the 10-bit binary strings in the middle, and the parameter K_d responds to the final 10-bit binary strings behind, in which the left bit is the low bit and the right bit is the high bit. The value range of K_p is [0, 15]. The decoding of K_p can be expressed as Eq. 3 and Eq. 4.

$$S = 1\times 2^0 + 1\times 2^1 + 1\times 2^2 + 1\times 2^3 + 1\times 2^4 + 0\times 2^5 + 1\times 2^6 + 1\times 2^7 + 0\times 2^8 + 1\times 2^9 = 735 \quad (3)$$

$$K_p = 0 + \frac{735}{2^{10}-1}\cdot(15-0) = 10.7771 \quad (4)$$

(3) GA operators

First, the ranges of parameters K_p, K_i, and K_d of the PID controller are roughly estimated using engineering experience; second, the initial population is randomly generated within this range. The random operation distributes individuals evenly to ensure the diversity of the population.

GA operators involve three types of operators: selection, crossover, and mutation. The selection operator selects individuals in the population for reproduction. The fitter the individual is, the more times it is to be selected for reproduction. The common selection methods of GAs are fitness-proportionate selection, the roulette-wheel sampling method, and the linear ranking method. The fitness-proportionate selection method was employed in this paper. In this selection method, the number of times an individual is expected to reproduce is in direct proportion to its fitness. The crossover operator chooses a random locus and exchanges the subsequences before and after the locus between two individuals to create two offspring. The crossover operator can conduct single-point crossover, two-point crossover, and multipoint crossover; single-point crossover was employed in this paper. The mutation operator randomly flips some of the bits in a chromosome. The mutation operator can destroy and create instances, thereby maintaining the diversity of the population [6].

(4) The process of PID controller parameter optimization based on GA

The process of PID controller parameter optimization based on the GA is as follows:

1) Determine the parameters of the GA. Parameters such as the number of individuals in the population *Size*, the maximum number of generations *N*, the form of GA encoding, and the encoding length *L* of the GA, as well as the ranges of parameters K_p, K_i, and K_d of the PID controller, are determined.

2) Generate an initial population of random binary strings of length *L*. The values of the individuals must be taken to ensure that the parameters K_p, K_i, and K_d of the PID controller are in the range determined in step 1.

3) Evaluate the fitness of each individual in the population. The parameters K_p, K_i, and K_d decoded from chromosomes are employed in the PID controller to obtain its system response. The fitness of each individual is calculated based on the system response, and the individuals are ranked in proportion to their fitness.

4) Employ GA operators. The GA operators of selection, crossover, and mutation are employed to obtain a new population.

5) Repeat steps 3 and 4 until the evolution termination condition is met. The termination condition of a GA is generally identifying the best individual or reaching the maximum number of generations.

The workflow chart of PID controller parameter tuning based on the GA is as follows:

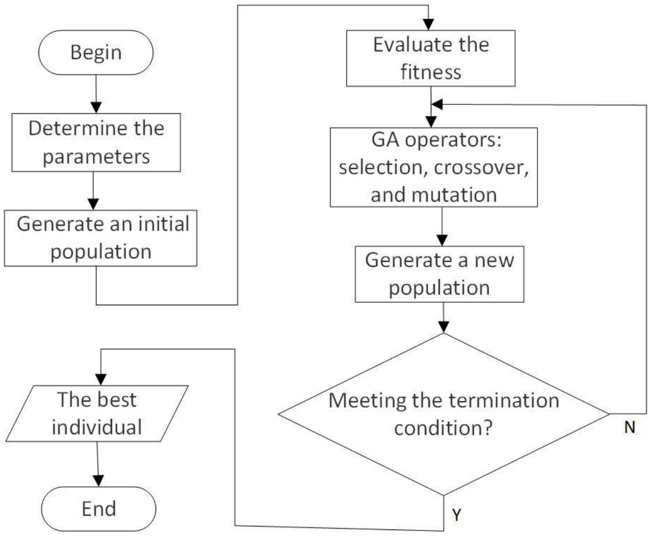

Fig. 2 Workflow of the GA

C. Cellular genetic algorithm

(1) Overview

Cellular automata were first introduced by von Neumann and Ulam for the purpose of modeling biological self-reproduction [22]. The cellular model simulates natural evolution from the point of the individual. It provides the population of a special structure, which is defined as a connected graph in which each vertex is an individual who can communicate only with his neighbors [23]. CGA assigns one individual per processor and limits reproduction within neighborhoods. This leads to a form of locality known as "isolation-by-distance". As a result, CGA has excellent characteristics, such as implicit parallelism and multimodality, allowing it to avoid falling into a local optimum.

(2) Cellular genetic automata

Cellular automata have four elements: cells, cellular space, neighborhoods, and evolution rules. Cells are the basic components of cellular automata distributed in cellular space. The cellular space is a special structure defined as a connected graph in which each vertex is an individual who communicates with its nearest neighbors. The structure can be one-dimensional, two-dimensional, or higher-dimensional; typically, the structure used is two-dimensional. There are triangular, square, and hexagonal grids for two-dimensional cellular automata. Neighborhoods are demes wherein individuals are allowed to reproduce with the central individual. The evolution rules refer to the mathematical function of an individual from the current state to the new state.

Two-dimensional cellular automata and square grids were employed in this paper. The square grids shown in Fig. 3 were constructed, in which each mesh represents an individual. The Von Neumann neighborhood was used, and the left, right, upper and lower cells are the neighbors of the cell C_{ij}.

The neighbors are shown in Eq. 5, in which parameters *i* and *j* represent the position of the cell in the cellular space and parameters *x* and *y* represent the position of their neighbors. If two individuals are neighbors, they are adjacent in logic. Logical adjacency includes positional adjacency, leftmost and rightmost adjacency, and uppermost and lowermost adjacency. For example, if the parameters x and i range from 1 to 5 and the parameters y and j range from 1 to 6, the individuals (1,:) are the uppermost individuals, the individuals (5,:) are the lowermost individuals, the individuals (:,1) are the leftmost individuals, and the individuals (:,6) are the rightmost individuals. In addition to the neighbors described in Eq. 5, the uppermost individuals are adjacent to the lowermost individuals, and the leftmost individuals are adjacent to the rightmost individuals. For example, individual (1,1) is adjacent to individual (5,1) and individual (1,6).

$$Neighbor = \left\{ C_{x,y} \| x-i | + | y-j | \leq 1 \right\} \tag{5}$$

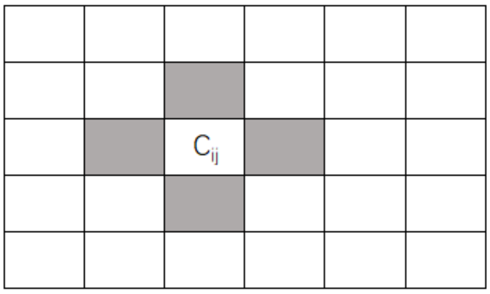

Fig. 3 Von Neumann Neighborhood

(3) The process of PID controller parameter optimization based on the CGA

The cellular genetic algorithm (CGA) is an evolutionary algorithm that combines the principles of cellular automata and genetic algorithms. The CGA employs the topology and evolutionary rules of cellular automata to improve the performance of the GA. The process of PID controller parameter optimization based on the CGA is described step by step as follows:

1) Determine the parameters of the CGA. Parameters such as the number of individuals in population *Size*, the maximum number of generations *N*, the form of CGA encoding, and the encoding length *L* in the CGA are determined, and the ranges of parameters K_p, K_i, and K_d of the PID controller are determined. The form of the neighborhood is also determined.

2) Generate an initial population of 3*L bits random binary strings. The values of the individuals must be chosen to ensure that the parameters K_p, K_i, and K_d of the PID controller are in the range determined in step 1. The individuals in the first population are distributed in the square grids of Von Neumann.

3) Evaluate the fitness of each individual in the population. The parameters K_p, K_i, and K_d of the PID controller decoded from chromosomes are employed in the PID controller to obtain its system response. The fitness of each individual is calculated based on the system response.

4) The GA operators of selection, crossover, and mutation are employed in this step. The selection and crossover operators choose the fittest neighbor for reproduction. The offspring of these neighbors will replace them if they are fitter, while other neighbors will remain as they are.

5) Repeat steps 3 and 4 until the evolution termination condition is met. The termination condition of a GA is generally identifying the best individual or reaching the maximum number of generations.

III. Computer Simulation Experiment

A. Building simulation models

The simulation model of parameter tuning of the PID controller based on the GA is presented in Fig. 4. The GA program generates K_p, K_i, and K_d randomly in the range determined in section II, and the three parameters are employed in the PID controller to obtain its step response and error signal *e*. The GA program reproduces a new generation through selection, crossover, and mutation operators based on the obtained data.

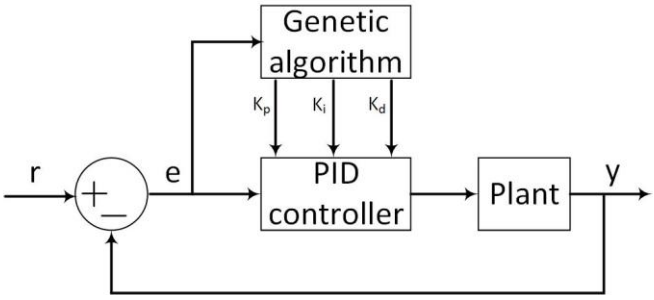

Fig. 4 Simulation model of parameter tuning of the PID controller based on the GA

The GA program judges the merits and pitfalls of individuals based on the fitness values and executes GA operators based on the results. Individuals with higher fitness have more chances to reproduce, and the others have fewer chances to reproduce or are eliminated. By considering the stability, rapidity, and accuracy of the control system, Eq. 6 was selected as the objective function in this paper, where *e* (t) is the system error, *u* (t) is the controller output, *ey* (t) represents the overshoot, as shown in Eq. 7, and t_u is the system rising time.

$$J = \int_0^t (w_1 \cdot | e(t) | + w_2 \cdot u^2(t) + w_3 \cdot | ey(t) |) dt + w_4 \cdot t_u \tag{6}$$

$ey(t) = y(t) - y(t-1)$ when $y(t) - y(t-1) > 0$; otherwise, it is 0. (7)

The system error, e (t), represents the ability to track the input signals of the control system and accounts for the largest weight in the objective function. The square term of *u(t)* is added to the objective function to prevent the system overshoot from being overly large, and the terms *ey(t)* and t_u are added to limit the response speed of the system. The smaller the objective function is, the better the system performs. A fitness function that is the reciprocal of the objective function is shown in Eq. 8 for convenient calculation. The larger the fitness function is, the better the system performance is.

$$F = \frac{1}{\int_0^t (w_1 \cdot | e(t) | + w_2 \cdot u^2(t) + w_3 \cdot | ey(t) |) dt + w_4 \cdot t_u} \tag{8}$$

B. Parameter tuning of the PID controller based on the GA

The parameter tuning of the PID controller based on the GA was simulated in MATLAB. The SIMULINK model of the control system was established as shown in Fig. 4 above, in which the controlled object is described as $\frac{133}{s^2+25s}$ and w_1=0.99, w_2=0.01, w_3=0.01, and w_4=0.01. The configuration of the GA for the control system is shown in TABLE 1. The termination condition of a GA is generally identifying the best individual or reaching the maximum number of generations. In this paper, reaching the maximum number of generations was adopted as the termination condition. The change curve of the objective function of the best individual in each population is shown in Fig. 5, and the evolutionary frequency of the GA is shown in TABLE 2. The objective function value of each individual in the last population is 23.3726, and each individual has the same gene sequence. As shown in Fig. 5 and TABLE 2, 12 evolutions occur in the 100 generations, with 11 evolutions occurring in the first 30 generations, and 1 evolution occurring in the 46th generation. This situation shows that the objective function value declines quickly during the GA process, and the population converges too early.

TABLE 1. THE CONFIGURATION OF THE GA

Parameters	Values
Population size *Size*	30
Evolution algebra *N*	100
Encoding	Binary encoding
Encoding length *L*	10

C. Parameter tuning of the PID controller based on the CGA

The SIMULINK model and common parameters are consistent with the parameter tuning of the PID controller based on the GA. A two-dimensional distribution and Von Neumann neighborhood are employed. In the selection operation, the cell with the highest fitness is selected and replicated; the cell with the lowest fitness in the four corners of the cell is replaced; and the other cells are retained, maintaining the diversity of the system. In the crossover operation, the cell with the highest fitness reproduces with one of its neighbors, and the neighbors with smaller fitness values remain as they are. In the mutation operation, all cells undergo the mutation operation except for the best individual. The curve of the objective function of the best individual is shown in Fig. 5. The evolutionary frequency of the CGA is shown in TABLE 2. A diagram of the distribution of the objective function values of the last population is shown in Fig. 6. The individuals in the last population are classified according to their distances from the best individual, and the average value of the objective function of the classifications is shown in Fig. 7.

As shown in Fig. 5, the horizontal ordinate is the number of generations in the population, and the ordinate is the objective function value of the best individual in each generation. As shown in TABLE 2, the frequency of GA is the number of changes in the best individual during the generations shown in the table of the GA, and the frequency of CGA is the number of changes in the best individual during the generations shown in the table of the CGA. As shown in Fig. 6, the row and column axes represent the coordinates of each individual in the last population of the CGA, and the BsJi axis represents the objective function value of the last population. The individual with the row and column coordinates (1,1) is the best. As shown in Fig. 7, the Distance axis is the distance of individuals from the best individual shown in Eq. 9,10,11,12, and 13, and the average of the Ji axis is the average of the objective function values of the individuals whose distances from the best individual are 1, 2, 3, 4, or 5.

$$Distance1 = |Row-1| + |Column-1| \tag{9}$$

$$Distance2 = |Row-5| + |Column-1| + 1 \tag{10}$$

$$Distance3 = |Row-1| + |Column-6| + 1 \tag{11}$$

$$Distance4 = |Row-5| + |Column-6| + 2 \tag{12}$$

$$Distance = \min(Distance1, Distance2, Distance3, Distance4) \tag{13}$$

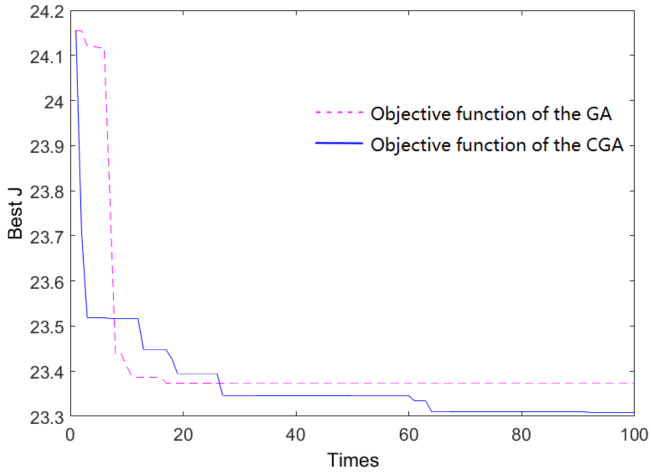

Fig. 5 Objective function of the best individual

TABLE 2. THE EVOLUTIONARY FREQUENCY

Generations	Frequency of GA	Frequency of CGA
1–10	7	3
10–20	2	3
20–30	2	1
30–40	0	1
40–50	1	1
50–60	0	0
60–70	0	3
70–80	0	0
80–90	0	0
90–100	0	1

As shown in Fig. 5 and TABLE 2, 13 evolutions occur in the 100 generations of the CGA. The distribution of the CGA evolutions is more average than that of the GA evolution. The process of the CGA's evolution is slow but continuous. As shown in Fig. 6 and Fig. 7, individual (1,1) is the best, and individuals farther from the best individual have a larger objective function as the distance increases. In CGA, each cell and its neighbors form a subecosystem, and each subecosystem evolves independently to a certain extent. Each subecosystem affects others through its boundaries, and individuals near the best individual exhibit better performance. As each subecosystem evolves independently, some individuals evolve more rapidly than others nearby, such as those whose distance is 5 from the best individual, as shown in Fig. 6 and Fig. 7.

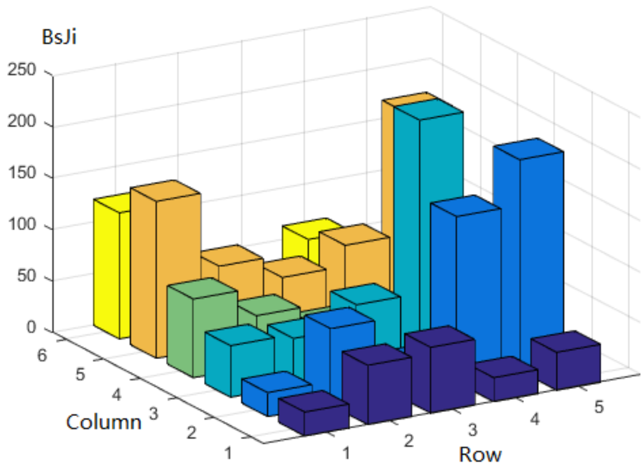

Fig. 6 Objective function values of the last population

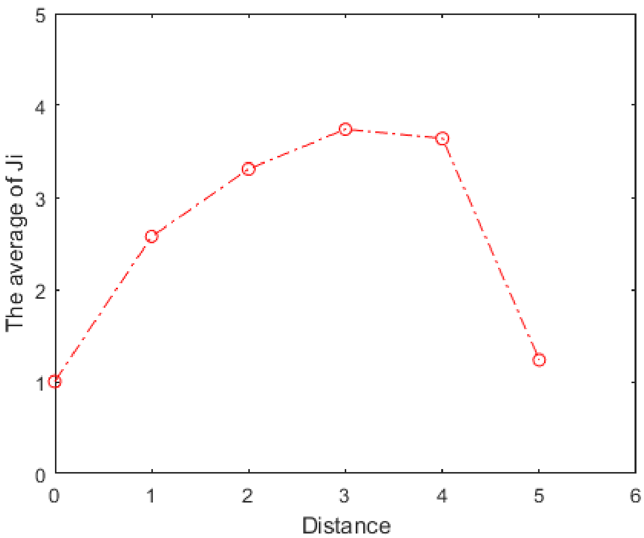

Fig. 7 The average value of the objective function

IV. CONCLUSIONS

During the GA process, the early objective function value declines quickly, and the population converges too early. This is due to the global influence of the best individual on the whole population, which can easily cause premature convergence and local optimum. This is verified in Fig. 5 and TABLE 2. In the CGA, each cell and its neighbors form a subecosystem, and each subecosystem evolves independently from the others; however, each sub-ecosystem affects others through its boundaries. This approach ensures the diversity of the population, the multioptimum, and the global search ability of the algorithm. The objective function value of the CGA gradually decreases, while new fittest individuals are constantly obtained; thus, a better final result was obtained with the CGA than with the GA. The last population objective function value of PID controller parameter tuning based on the CGA is shown in Fig. 6, which shows the diversity of the population retained in the last generation of the CGA.

REFERENCES

[1] Yan-e Hou, Wenwen He, Xianyu Zuo, Lanxue Dang, and Hongyu Han, "A Task Scheduling Approach based on Particle Swarm Optimization for the Production of Remote Sensing Products," *IAENG International Journal of Computer Science*, vol. 50, no.1, pp23-31, 2023.

[2] Gonggui Chen, Qilin Qin, Zhou Ping, Kang Peng, Xianjun Zeng, Hongyu Long, and Mi Zou, "A Novel Approach Based on Modified and Hybrid Flower Pollination Algorithm to Solve Multi-objective Optimal Power Flow," *IAENG International Journal of Applied Mathematics*, vol. 51, no.4, pp966-983, 2021.

[3] Jie Qian, and Gonggui Chen, "Improved Multi-goal Particle Swarm Optimization Algorithm and Multi-output BP Network for Optimal Operation of Power System," *IAENG International Journal of Applied Mathematics*, vol. 52, no.3, pp576-588, 2022.

[4] Gonggui Chen, Ying Xiao, Fangjia Long, Xiaorui Hu, and Hongyu Long, "An Improved Marine Predators Algorithm for Short-term Hydrothermal Scheduling," *IAENG International Journal of Applied Mathematics*, vol. 51, no.4, pp936-949, 2021.

[5] Gonggui Chen, Handu Zhuo, Xiaorui Hu, Fangjia Long, and Hongyu Long, "Environment Economic Power Dispatch from Power System Based on Multi-objective Novel Tree Seed Optimization Algorithm," *IAENG International Journal of Computer Science*, vol. 48, no.4, pp845-861, 2021.

[6] Mitchell Melanie, *An Introduction to Genetic Algorithms*. MIT Press, Cambridge. DBLP, 1998, pp. 128-130.

[7] Chunlu Wang, Hongyi Shi, and Xinquan Zuo, "A multi-objective genetic algorithm based approach for dynamical bus vehicles scheduling under traffic congestion," *Swarm and Evolutionary Computation*, vol. 54, 2020, Art. no. 100667.

[8] Jaume Jordán, Javier Palanca, Elena del Val, Vicente Julian, and Vicent Botti, "Localization of charging stations for electric vehicles using genetic algorithms," *Neurocomputing*, vol. 452, pp. 416-423, 2021.

[9] Feng Chen, Shouzhi Xu, Ying Zhao, and Hui Zhang, "An Adaptive Genetic Algorithm of Adjusting Sensor Acquisition Frequency," *Sensors*, vol. 20, no. 4, Feb 12 2020, Art. no. 990.

[10] Olateju Samuel Olaniyi, Adebayo Kayode James, Abdullahi Adinoyi Ibrahim, and Aderibigbe Felix Makanjuola, "On the Application of a Modified Genetic Algorithm for Solving Vehicle Routing Problems with Time Windows and Split Delivery," *IAENG International Journal of Applied Mathematics*, vol. 52, no.1, pp101-109, 2022.

[11] Lina Wang, Wenbin Dai, Jinjie Liu, Xiaohong Cui, and Binrui Wang, "Research on the Prediction Model of Greenhouse Temperature Based on Fuzzy Neural Network Optimized by Genetic Algorithm," *IAENG International Journal of Computer Science*, vol. 49, no.3, pp828-832, 2022.

[12] Md. Manjurul Gani, Md. Saiful Islam, and Muhammad Ahsan Ullah, "Optimal PID tuning for controlling the temperature of electric furnace by genetic algorithm," *SN Applied Sciences*, vol. 1, no. 8, 2019, Art. no. 880.

[13] A. Ruby Meena and S. Senthil Kumar, "Genetically tuned fuzzy PID controller in two area reheat thermal power system," *Russian Electrical Engineering*, vol. 87, no. 10, pp. 579-587, 2016.

[14] Guanzheng Tan and Anping Li, "Fuzzy-GA PID controller with incomplete derivation and its application to intelligent bionic artificial leg," *Journal of Central South University of Technology*, vol. 10, no. 3, pp. 237-243, 2003.

[15] Fulu Cao, "PID controller optimized by genetic algorithm for direct-drive servo system," *Neural Computing and Applications*, vol. 32, pp. 23-30, 2020.

[16] Mingjian Zhou, Yuqin Wang, and Haibing Wu, "Control Design of the Wave Compensation System Based on the Genetic PID Algorithm," *Advances in Materials Science and Engineering*, vol. 2019, pp. 1-13, 2019.

[17] Xiaogen Shao, Liqing Xiao, and Chengchun Han, "Optimization of PID Parameters Based on Genetic Algorithm and Interval Algorithm," *Chinese Control and Decision Conference*, vol. 1, pp. 741-745, 2009.

[18] S. P. Ghoshal, "Application of GA/GA-SA based fuzzy automatic generation control of a multi-area thermal generating system," *Electric Power Systems Research*, vol. 70, no. 2, pp. 115-127, 2004.

[19] Yongdaun Song, *Control of Nonlinear Systems via PI, PD and PID: Stability and Performance.* Taylor & Francis Group, 2018, pp. 5-8.

[20] Gene F. Franklin, J. David Powell, and Abbas Emami-N, *Feed back Control of Dynamic System.* Pearson, 2018, pp. 550-596.

[21] Wolfgang Banzhaf, Peter Nordin, Robert E.Keller, and Frank D.Francone, *Genetic Programming- an introduction: on the automatic*

evolution of computer programs and its applications. Morgan Kaufmann Publishers Inc. 1998, pp. 264-266.
[22] Stephen Wolfram, *Cellular automata and complexity (collected papers)*. CRC Press, 2018, pp. 3-7.
[23] Enrique Alba and Bernabe Dorronsoro, *Cellular Genetic Algorithms.* Springer, 2008, pp. 13-17.