Hindawi
Mathematical Problems in Engineering
Volume 2019, Article ID 1378783, 11 pages
https://doi.org/10.1155/2019/1378783

Research Article

An MEA-Tuning Method for Design of the PID Controller

Yongli Zhang,[1] **Lijun Zhang**,[2] **and Zhiliang Dong**[1]

[1] *School of Management Science and Engineering, Hebei GEO University, Shijiazhuang, Hebei, China*
[2] *Earth Sciences Museum, Hebei GEO University, Shijiazhuang, Hebei, China*

Correspondence should be addressed to Yongli Zhang; zhangyongli086@163.com

Received 1 November 2018; Revised 3 December 2018; Accepted 8 January 2019; Published 20 January 2019

Guest Editor: Anders E. W. Jarfors

The optimization and tuning of parameters is very important for the performance of the PID controller. In this paper, a novel parameter tuning method based on the mind evolutionary algorithm (MEA) was presented. The MEA firstly transformed the problem solutions into the population individuals embodied by code and then divided the population into superior subpopulations and temporary subpopulations and used the similar taxis and dissimilation operations for searching the global optimal solution. In order to verify the control performance of the MEA, three classical functions and five typical industrial process control models were adopted for testing experiments. Experimental results indicated that the proposed approach was feasible and valid: the MEA with the superior design feature and parallel structure could memorize more evolutionary information, generate superior genes, and enhance the efficiency and effectiveness for searching global optimal parameters. In addition, the MEA-tuning method can be easily applied to real industrial practices and provides a novel and convenient solution for the optimization and tuning of the PID controller.

1. Introduction

The proportional integral derivative (PID) controller is the most widely used and most mature in industrial production [1, 2]. So far, the vast majority of industrial controllers are PID controllers or their second generations owing to the advantages of simple structure, strong robustness, and easy realization [3–5]. However, once the controller characteristics, control scheme, interference form, and size are basically fixed, the quality of the control system depends on the setting of the controller parameters. Therefore, PID parameter tuning and optimizing is one core issue for PID controller design [6, 7].

The parameter tuning methods for PID controller design fall into two basic categories: conventional parameter tuning methods and intelligent optimization algorithms. Conventional parameter tuning methods include the empirical method, upwards curve method, critical ratio method, damping oscillatory method, and relay feedback method [8]. The Z-N method as an empirical parameter tuning method had a very strong impact on the actual control system; practically all vendors and users of the PID controller apply it or its simple modifications in controller tuning. The Z-N method was developed by Ziegler and Nichols in seminal paper [9], which was based on two ideas: to characterize process dynamics by two parameters, which are easily determined experimentally, and to calculate controller parameters from the process characteristics by a simple formula. Although widely used in practice, conventional parameter tuning methods have large overshoot, poor correction accuracy, and time-consuming tuning process [10–13].

With the development of the intelligent control theory, intelligent optimization algorithms started to be applied in PID parameter tuning and optimizing and achieved incomparable results that the conventional parameter tuning methods cannot obtain. For example, the genetic algorithm (GA) [14–16], particle swarm optimization (PSO) [17–19], tabu search algorithm (TSA) [20–22], bacterial foraging algorithm (BFA) [23–25], ant colony algorithm (ACA) [26], artificial bee colony (ABC) algorithm [27], and BAT search algorithm [28] were adopted to optimize PID controller parameters and had achieved much better performances.

Although intelligent optimization algorithms possess merits of strong robustness, good universality, and few limited conditions for use, some intelligent algorithms still have defects such as premature convergence and slow convergence

rate. For example, the GA, PSO, and TSA are all stochastic search algorithms. Their search processes are nondeterministic, suffer from slow convergence, and easily fall into the local optimal solution [29, 30]. The parameter settings of the GA and ACA have not clear theoretical basis, and most parameters still need to be determined by experience and experiment. The population individual of the PSO and BAT search algorithm lacks the mutation mechanism; once trapped in the local extreme, it is difficult to get rid of [31]. The complexity of the ABC algorithm programming limits its application to a certain extent. In addition, there are few controlled process models that adopt the ABC algorithm for PID parameter tuning, and its generality is not high [32, 33]. To further improve the optimization performance and overcome the defects of traditional algorithms, Chinese scholar Sun Cheng-Yi et al. raised the mind evolutionary algorithm (MEA) based on the genetic algorithm in 1998 [34]. The MEA is an evolutionary algorithm simulating the progress of human mind and has the positive and negative feedback mechanism, wherein the positive feedback mechanism improves toward being more beneficial to the population survival, so as to consolidate and develop the evolution achievement. The negative feedback mechanism prevents the premature convergence of the algorithm, so as to avoid the situation in which the algorithm is caught in the local optimal solution. The structural parallelism of the MEA guarantees the high search efficiency of the algorithm, overcomes the defects of the GA such as time-consuming computation and premature convergence, and also has extremely strong robustness on interference [35, 36].

While theory application and engineering practices have proved that the MEA has very high search efficiency and convergence performance [37–39], few researches apply it to the PID parameter tuning and optimizing. Therefore, this paper is undertaken to establish a parameter tuning method using the MEA for PID controller design to fill the research gap.

The rest of this article is organized as follows. The principle of the MEA and MEA-tuning process for PID controller design are described in Section 2. Three classical functions are used to test the performance of the MEA and compare it with the GA in Section 3. The MEA is applied to the parameters tuning of five classical PID controllers and compared with the GA and traditional Z-N method in optimization performance, convergence time, and robustness in Section 4. Finally, conclusions are drawn and future researches are suggested in Section 5.

2. Mind Evolutionary Algorithm

2.1. Algorithm Principle. The mind evolutionary algorithm (MEA) is a kind of evolutionary algorithm simulating the progress of the human mind, developed on the basis of the GA. The MEA uses two types of operation, similar taxis and dissimilation, and uses population optimizing instead of individual optimizing, which avoids the defects of the GA. The similar taxis operation is a process in which the individual competes to be a winner, which happens within the scope of a subpopulation. The dissimilation operation is the process in which a subpopulation competes to be a winner and continuously explores the new point of solution space, which happens within the whole solution space. In the course of the algorithm running, the similar taxis and dissimilation operation is executed repeatedly until the condition of terminating the running of algorithm is satisfied.

Compared with the GA, the MEA has following advantages: the crossover and mutation operations of the GA generate not only superior genes, but also inferior destructive genes, and those operations have duality, but the MEA uses the similar taxis and dissimilation operations, which amend the defects of the GA; the similar taxis and dissimilation operations of the MEA are coordinated mutually but also independent mutually, and any improvement on any aspect will raise the algorithm's prediction accuracy; the similar taxis and dissimilation operations have parallelism on structure, which raises the algorithm's search efficiency and computation speed; the MEA divides the populations into superior subpopulations and temporary subpopulations, which can memorize evolutionary information more than one generation.

2.2. Basic Concepts. The MEA follows some basic concepts of the GA such as "population," "individual," and "environment," but meanwhile it also adds some new concepts.

2.2.1. Population and Subpopulation. The MEA is a kind of learning method making optimization through iteration, and all individuals in every generation of the evolutionary process gather into one population. A population is divided into several subpopulations. The subpopulation contains two classes: superior subpopulation and temporary subpopulation. The superior subpopulation records the information of the winners in the global competition, and the temporary subpopulation records the process of the global competition.

2.2.2. Billboard. The billboard is equivalent to an information platform, which provides chances of information communication between the individuals or the subpopulations. The billboard records three types of effective information: the serial number of individual or subpopulation, the action, and the score. By utilizing the serial number of the individual or subpopulation, it is convenient to distinguish different individuals or subpopulations; the description of action varies from different research fields, and since this article is researching the problem of parameter optimization, the action is used to record the exact position of the individual and subpopulation; the score is the evaluation of environment on the individual action, and, in the optimization process by utilizing the MEA, it can rapidly find out the optimized individuals and populations only if the scores of every individual and subpopulation are recorded all the time. The individuals in the subpopulation post up their own information on the local billboard, and the information of each subpopulation is posted up on the global billboard.

2.2.3. Similar Taxis. Within the scope of a subpopulation, the process in which an individual competes to be a winner is called similar taxis. In the process of a subpopulation's

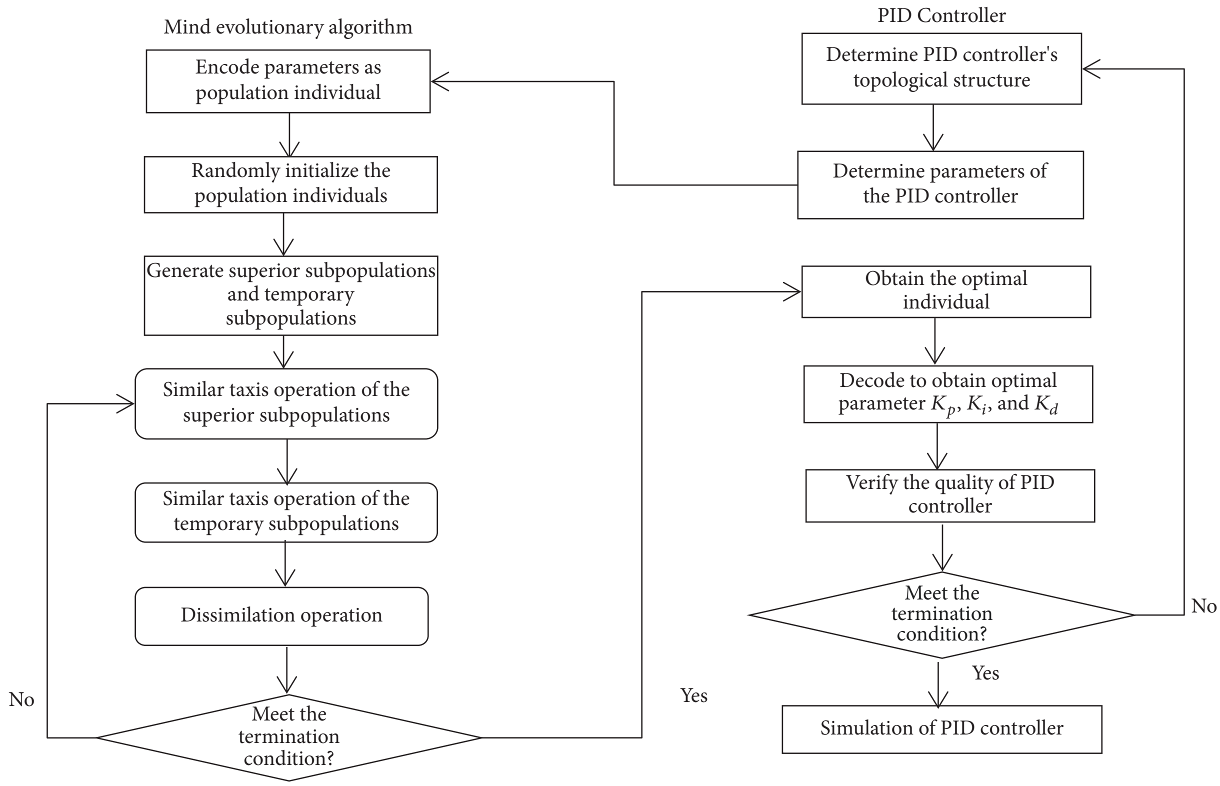

Figure 1: Parameter tuning process of the PID controller by the MEA.

similar taxis, if a new winner cannot be generated, this means that such subpopulation has matured. When a subpopulation matures, the similar taxis process of such subpopulation comes to an end. The period of a subpopulation from its birth to maturity is called as lifetime.

2.2.4. Dissimilation. In the whole solution space, each subpopulation competes to be a winner and continuously explores a new solution space point; this process is known as dissimilation. Dissimilation has two definitions: each subpopulation makes global competition, and if the score of a temporary subpopulation is higher than the score of a certain matured superior subpopulation, such superior subpopulation will be replaced by the winning temporary subpopulation, and the individuals of the original superior subpopulation will be released; if the score of a matured temporary subpopulation is lower than the score of any superior subpopulation, such temporary subpopulation will be abandoned, and the individuals therein will be released; the released individuals will re-search and form a new temporary subpopulation.

2.3. Calculation Procedure. The quality of the PID controller depends on the tuning and optimizing of parameters K_p, K_i, and K_d. To obtain the optimal PID controller, the parameters K_p, K_i, and K_d are encoded and taken as the population individual, the integral of time absolute errors (*ITAE*) obtained from the PID controller is regarded as the individual fitness value, and then the MEA begins to evolve. Through continuous iterative evolution of the similar taxis and dissimilation operation, the optimal individual of the population is finally obtained, which is taken as the parameter of the PID controller after being decoded; the optimal PID controller is established.

The tuning process of the PID controller by using the MEA is shown as Figure 1.

Step 1 (initialization of the population's individual). The parameters K_p, K_i, and K_d of PID controller coding in the real number are taken as the population individual and the integral of time absolute errors (*ITAE*) is regarded as the individual fitness value, and then N initial population's individuals are generated randomly.

Step 2 (generation of the superior subpopulation and temporary subpopulation). According to the individual fitness value, the N_s superior individuals with the highest scores and N_t temporary individuals with the next highest scores are picked out. Taking each superior individual or temporary individual as center, new individuals are produced and N_s superior subpopulations and N_t temporary subpopulations are formed, named as G_s and N_t, respectively. The number of individuals in the subpopulation is $N_p = N/(N_s + N_t)$.

Step 3 (similar taxis). To compute the fitness values of all individuals in the superior subpopulation and temporary subpopulation, the optimal population $g_s{}^*$ or $g_t{}^*$ (namely, the center) in the subpopulation is taken as the center of the subpopulation, and the score of the optimal individual is

taken as the score of such subpopulation. The fitness values of $g_s{}^*$ and $g_t{}^*$ are described as below:

$$f_s{}^* = \min\{(f(g) : g \in G_s),\ 1 \le s \le N_s\} \tag{1}$$

$$f_t{}^* = \min\{(f(g) : g \in G_t),\ 1 \le t \le N_t\}. \tag{2}$$

Step 4 (judgment of subpopulation's maturity). In the process of subpopulation similar taxis, when the population center is the optimal individual, and no new superior individual is generated any more, then such subpopulation matures and turns into the next step; otherwise it turns into the previous step and executes the similar taxis operation again.

Step 5 (dissimilation). After the subpopulation matures, the score of each subpopulation will be posted up on the global billboard, and if the score of any temporary subpopulation G_t is higher than the score of any matured superior subpopulation G_s, such superior subpopulation G_s will be replaced by the winning temporary subpopulation G_t, the individuals in the original superior subpopulation G_s will be released, the originally winning temporary subpopulation G_t will be replaced by the new temporary subpopulation $G_t{}'$, and the individuals in $G_t{}'$ will be evenly distributed in the solution space.

Step 6 (iterative evolution). The superior subpopulation with the highest fitness value in all superior subpopulations or temporary subpopulations is selected to judge whether the termination condition is satisfied. If yes, the evolution will be terminated, and if not, the operation of Steps 3~5 will be repeated.

Step 7 (decoding of the optimal individual and establishing of the PID controller optimized by the MEA). When the termination condition of the MEA is met, the superior subpopulation center as the optimal population individual will be decoded to obtain the parameters K_p, K_i, and K_d of the PID controller, and the optimized PID controller is established.

3. Classical Test Functions Verification

In order to verify the validity of the MEA, 3 classical functions are selected for testing and compared with the GA. The test functions are the Sphere function, Rastrigin function, and Rosenbrock function.

(1) The Sphere Function

$$f(x, y) = x^2 + y^2. \tag{3}$$

In formula (3), x, $y \in [-10, 10]$; when (x,y)=(0,0), the Sphere function reaches its minimum 0 (Figure 2).

(2) The Rastrigin Function

$$f(x, y) = 20 + x^2 - 10\cos(2\pi x) + y^2 - 10\cos(2\pi y). \tag{4}$$

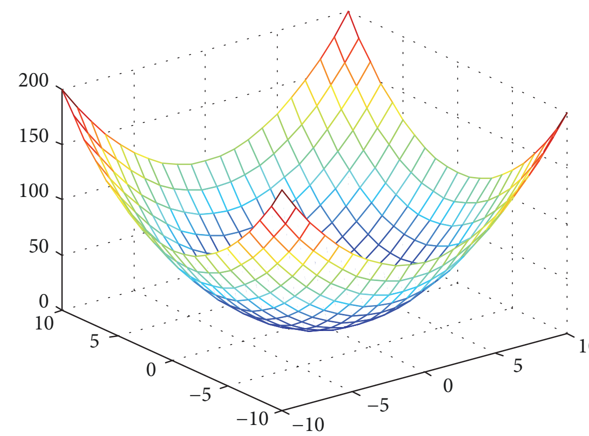

Figure 2: Sphere function graph.

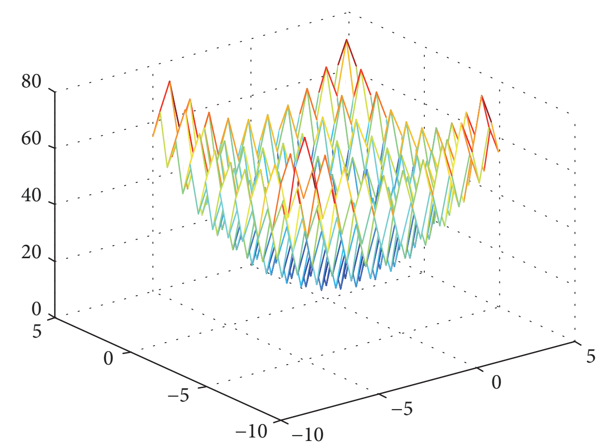

Figure 3: Rastrigin function graph.

In formula (4), x, y∈[-5.12,5.12]; the global optimal solution of the Rastrigin function is 0, distributed at (0,0) (Figure 3).

(3) The Rosenbrock Function

$$f(x) = \sum_{i=1}^{N-1} \left(x_i^2 - x_{i+1}\right)^2 + \left(1 - x_i\right)^2. \tag{5}$$

In formula (5), x, $y \in [-10, 10]$; the Rosenbrock function is a pathological function that is hard to minimize, and the theoretical global minimum is 0 (Figure 4).

The MEA is established due to the defect of the GA, so this paper makes a comparative test between the MEA and GA to verify the superiority of the MEA. In the test, parameters of the MEA and GA are set as follows: evolution generations $iter_{max}$=10, population size N=200, number of superior subpopulation N_s=5, number of temporary subpopulation N_t=5, subpopulation size N_p=20, crossover probability C=0.2, and mutation probability M=0.1. The testing results are displayed in Table 1.

As shown in Table 1, the test results of three classical functions prove that the optimization performance of the MEA is better than the GA. In Figure 5, the fitness evolution

Table 1: Optimization results and comparison for the MEA and GA.

Function	Theoretical extreme	MEA			GA		
		Mean value	Optimal value	Worst value	Mean value	Optimal value	Worst value
Sphere	0	1.3396	0.0004	4.7535	2.1702	0.1306	2.2877
Rastrigin	0	5.9375	0.1557	10.2223	5.9935	2.0612	17.6934
Rosenbrock	0	1.0562	0.0010	9.2253	2.5015	6.5593	5.6020

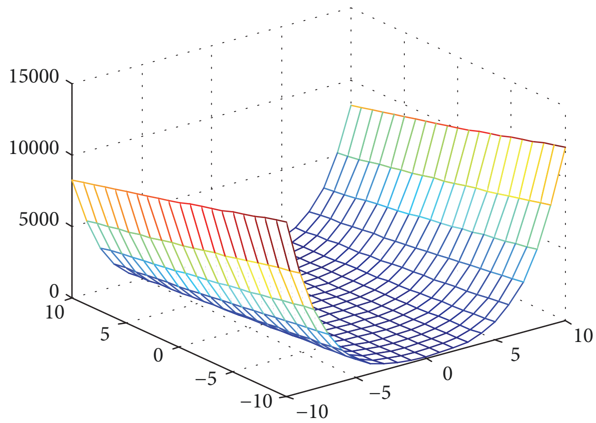

Figure 4: Rosenbrock function graph.

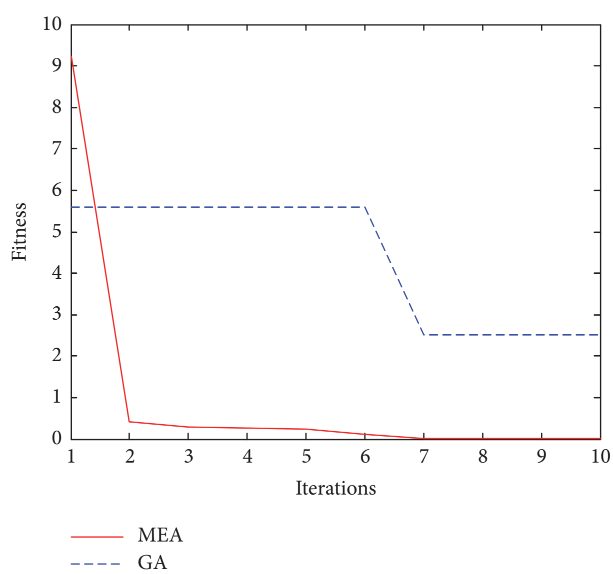

Figure 5: Evolutionary curves of optimal fitness for the Rosenbrock function.

curves of the Rosenbrock function show that, compared with the GA, the MEA has faster convergence speed and higher convergence precision. The experimental results indicated that, due to its superior design feature and parallel structure, the MEA could memorize more evolutionary information and search the optimal solution more efficiently and effectively, overcoming the defects of the GA, such as precociousness, easiness of falling into the local extremum, and time-consuming computation.

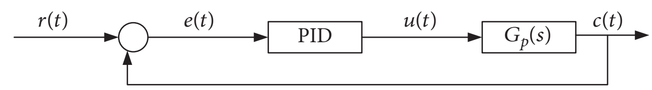

Figure 6: PID control system.

4. Simulation Studies

4.1. PID Controller and Performance Index. The proportional integral derivative (PID) controller is widely applied in industrial processes. The system structure of the PID controller is shown in Figure 6.

The general form of one PID controller is shown as below.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)\, d\tau + K_d \frac{de(t)}{dt}. \qquad (6)$$

In formula (6), $u(t)$ and $e(t)$ represent the control signal and error signal and K_p, K_i, and K_d are the proportional gain, integral gain, and derivative gain, respectively. By entering parameters K_p, K_i, and K_d, the control signal is calculated and then sent to the controlling module; the controlled object is driven. If the controller is designed properly, the control signal will make the output error converge to a small neighborhood of the origin to achieve the control requirement. Therefore, the optimization objective of the PID controller is to obtain an optimal set of parameters (K_p, K_i, K_d), which will minimize the performance function by searching the given controller parameters space.

In general, integration error indexes include the integral of squared errors (*ISE*), integrated absolute errors (*IAE*), and integral of time absolute errors (*ITAE*) [40]. In this paper, the *ITAE* is chosen as the performance indicator of the PID controller. The formula for calculation is as follows:

$$ITAE = \int_0^T t\,|e(t)|\, dt. \qquad (7)$$

If the functional model on error is regarded as a loss function, the integral index of formula (7) can be regarded as the control goal of a control system when it is transferred from one state to another; then it is considered that the system has the optimal control law with a minimal performance index. Generally, the adjustment time t_s corresponding to 5% or 2% error is taken as the upper bound T parameter of the integral.

Due to the existence of absolute error $|e(t)|$, formula (7) is difficult to solve by the analytic method, and the numerical methods are usually chosen to solve it; the specific steps are as follows: the continuous time is discretized with time step Δt; meanwhile, when the time $T = m\Delta t$ and m are

Table 2: Models for general controlled processes.

Type	Mathematical model	Model features
I	$\dfrac{e^{-\tau s}}{1+s}$ τ=0.5,1,2	First-order time-delay process τ=0.5 in this study
II	$\dfrac{e^{-\tau s}}{(1+s)^2}$ τ=0.5,1,2	Second-order time-delay process τ=1 in this study
III	$\dfrac{e^{-s}}{s(1+s)}$	Nonoscillating process
IV	$\dfrac{1-as}{(1+s)^2}$ a=0.5,1	Nonminimum phase process a=0.5 in this study
V	$\dfrac{1}{(1+s)^n}$ n=4,6,8	High-order process n=6 in this study

the integers of the appropriate size, the discrete time vector $t[n](n$=1,2,. . .,$m)$ is obtained. When t is small enough, the $ITAE$ can be calculated according to

$$ITAE = \sum_{i=0}^{m} t\,[i]\,|e\,[i]|\,\Delta t. \tag{8}$$

The process of PID parameter tuning using the MEA is actually an optimization problem. The parameter set (K_p, K_i, K_d) is regarded as the particle position vector $P(K_p,K_i,K_d)$ in the optimization space, K_p, K_i, and K_d are restricted to [0, K_{pm}], [0, K_{im}], and [0, K_{dm}], respectively, and the $ITAE$ performance index is chosen as the fitness function of the algorithm. Through the iteration of the MEA, a set of parameters that minimize the $ITAE$ value of the system response is found.

4.2. Simulation Experiment. In order to verify the control performance of the MEA, 5 typical industrial process control models $G_p(s)$ are selected as the research models, as shown in Table 2. Five typical industrial process control models are self-balanced and nonoscillating process with time lag of the first order, self-balanced and nonoscillating process with time lag of the second order, nonoscillating process without self-balance, nonminimum phase process with reverse characteristics, and high-order process.

According to the structural characteristics of the PID controller, the parameters to be optimized are K_p, K_i, and K_d. Then the parameters K_p, K_i, and K_d are encoded as the population individual; the individual length is 3, equal to parameter number. In order to compare algorithm performance, different algorithms need to set the same experimental conditions, so the population size is set to 200, the number of superior population or temporary population size is 5, the subpopulation size is 20, the evolution generations are set to 10, the crossover probability is set to 0.2, and the mutation probability is set to 0.1.

(1) Industrial Process Control Model I. The MEA, GA, and Z-N methods were adopted to search the optimal PID parameters of industrial process control model I, respectively; the PID parameters (K_p, K_i, K_d) and performance indexes ($ITAE$, t_s) obtained were listed in Table 3.

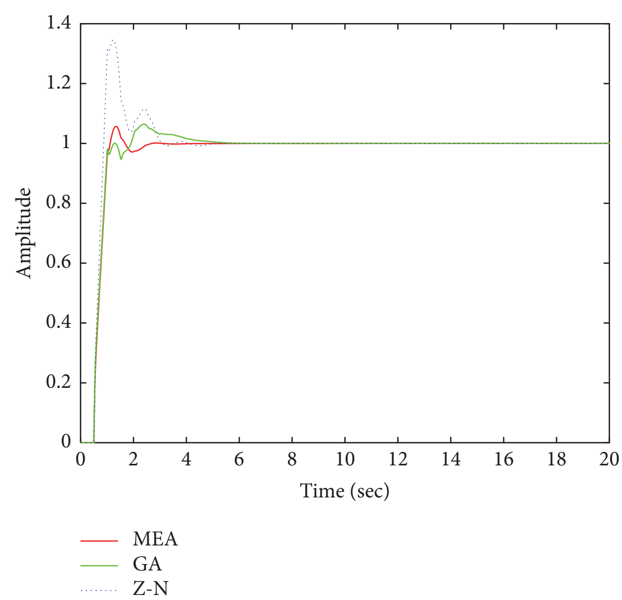

Figure 7: Closed-loop step response with process I.

Table 3 shows that the MEA-tuning method has the best fitting performance ($ITAE$=0.3463) and the fastest convergence speed (t_s=2.6910), followed by the GA-tuning method ($ITAE$=0.5758, t_s=5.6440) and the Z-N-tuning method ($ITAE$=0.7332, t_s=6.2987).

The obtained PID parameters (K_p, K_i, K_d) were brought into process model I, and the unit step responses of the MEA, GA, and Z-N methods are obtained, respectively, through Simulink (Figure 7). Figure 7 also proves that the MEA-tuning method controls process control model I more stably and faster.

(2) Industrial Process Control Model II. The optimal PID parameters (K_p, K_i, K_d) and performance indexes ($ITAE$,

Table 3: PID parameters and performances of industrial process control model I.

Parameter	MEA	GA	Z-N
K_p	1.7898	1.7449	2.2800
K_i	1.3770	1.5492	2.6576
K_d	0.2788	0.4228	0.4890
t_s	2.6910	5.6440	6.2987
ITAE	0.3463	0.5758	0.7332

Table 4: PID parameters and performances of industrial process control model II.

Parameter	MEA	GA	Z-N
K_p	1.3525	1.5441	1.6200
K_i	0.4982	0.5752	0.6707
K_d	0.8345	1.0517	0.9782
t_s	9.0798	9.5050	13.5319
ITAE	2.8671	3.2592	3.9535

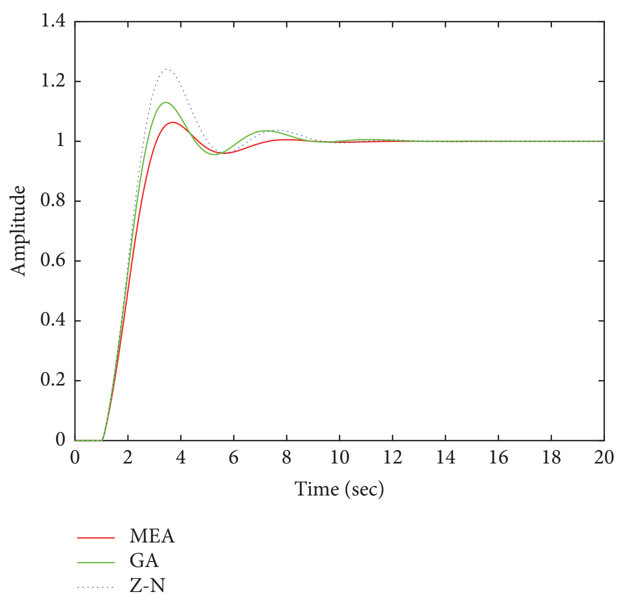

Figure 8: Closed-loop step response with process II.

t_s) tuned by the MEA, GA, and Z-N methods are obtained, as shown in Table 4. The performance indexes (*ITAE*, t_s) obtained by the MEA, GA, and Z-N methods are (*ITAE*=2.8671, t_s=9.0798), (*ITAE*=3.2592, t_s=9.5050), and (*ITAE*=3.9535, t_s=13.5319), respectively. Obviously, the MEA-tuning method has the least *ITAE* and the fastest search speed.

In order to further verify the effectiveness of the MEA, Simulink was used to investigate the unit step response of typical model control system II. The results show that, compared with the GA and Z-N methods, the MEA can make process control model II reach a stable state more quickly (Figure 8).

(3) Industrial Process Control Model III. The optimal PID parameters (K_p, K_i, K_d) and performance indexes (*ITAE*, t_s) of industrial process control model III were obtained by the MEA, GA, and Z-N methods and listed in Table 5. Compared with the GA method (*ITAE*=23.7167) and Z-N method (*ITAE*=23.7657), the MEA method can greatly improve the control performance (*ITAE*=11.2763), and the search speed of the MEA method is also improved slightly (t_s=15.2005). The unit step responses also show that the MEA method tunes process control model III with a smaller fluctuation range and faster convergence speed (Figure 9).

(4) Industrial Process Control Model V. The optimal PID parameters (K_p, K_i, K_d) and performance indexes (*ITAE*, t_s) of industrial process control model V were obtained by the MEA, GA, and Z-N methods and listed in Table 6. Compared with the GA method (*ITAE*=2.4037, t_s=7.2594) and Z-N method (*ITAE*=2.4623, t_s=8.7442), the MEA method has less *ITAE* and shorter calculation time, greatly improving the control performance (*ITAE*=0.7536) and search speed (t_s=3.3309). The GA has a slight advantage over the Z-N method, but it is not significant. The unit step responses also show that the MEA method tunes process control model V with a smaller fluctuation range and faster convergence speed (Figure 10).

(5) Industrial Process Control Model IV. The MEA, GA, and Z-N methods were used to search the optimal PID parameters of industrial process control model IV, respectively; then the PID parameters (K_p, K_i, K_d) were obtained and put into process control model IV through Simulink; the performance indexes (*ITAE*, t_s) were obtained and listed in Table 7.

Compared with the GA (*ITAE*=19.4274, t_s=18.1772) and Z-N method (*ITAE*=21.3847, t_s=20.0000), the MEA improves the control performance and computing speed, but not by much (*ITAE*=15.7474, t_s=16.1596). The unit step responses also show that none of the three methods achieve complete control of process control model IV within 20 seconds, but the MEA method has the smallest fluctuation range and the best performance (Figure 11).

The above simulation results show that, during the optimization process, the MEA constantly searches for better parameters, and the *ITAE* index decreases continuously;

Table 5: PID parameters and performances of industrial process control model III.

Parameter	MEA	GA	Z-N
K_p	0.9941	0.7477	0.6780
K_i	0.2601	0.2558	0.1855
K_d	0.9224	0.8571	0.6169
t_s	15.2005	18.4473	20.0000
ITAE	11.2763	23.7167	23.7657

Table 6: PID parameters and performances of industrial process control model V.

Parameter	MEA	GA	Z-N
K_p	2.6303	3.1253	2.3880
K_i	0.9459	1.829	1.6959
K_d	0.9984	1.7222	0.8406
t_s	3.3309	7.2594	8.7442
ITAE	0.7536	2.4037	2.4623

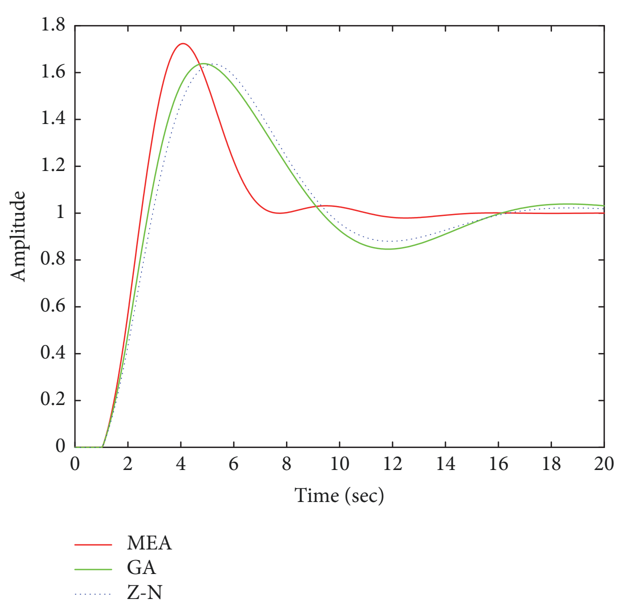

Figure 9: Closed-loop step response with process III.

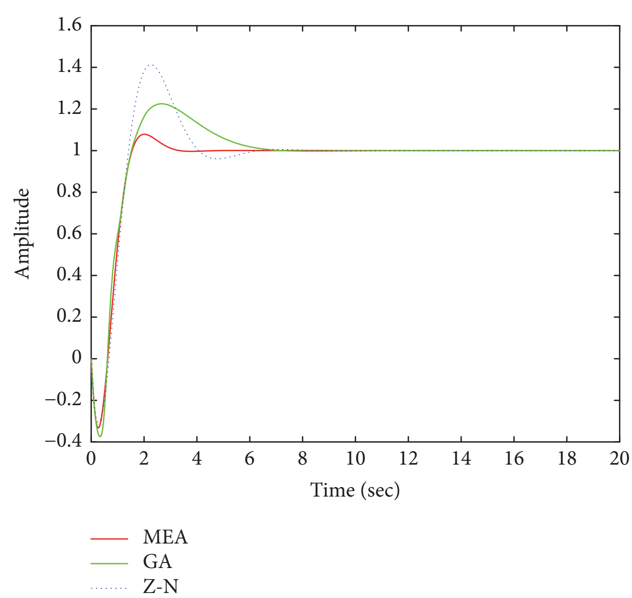

Figure 10: Closed-loop step response with process V.

finally, the unstable controlled object gradually reaches a steady state; this proves that the MEA-tuning PID controller well realizes the tuning and optimization of the controlled object. Compared with the GA and Z-N methods, the MEA can obtain satisfactory PID parameters for control systems within a few generations, the *ITAE* obtained is lower, and the convergence speed is faster. Therefore, the MEA is superior to the GA and Z-N methods.

5. Conclusion

The performance of the traditional PID control depends on the set of the parameters; therefore, an MEA-tuning method is proposed to search globally the optimal controller parameters. The MEA approach has superior design features and parallelism on structure, which raises the efficiency and effectiveness for searching global optimal parameters. In order to verify the performance of the MEA, three classical functions and five typical industrial process control models are employed for simulation and testing; the simulation results were compared with those of the GA and Z-N methods. The main results and conclusions are summarized as follows.

The experimental results of three classical functions (Sphere function, Rastrigin function, and Rosenbrock function) indicate that the MEA has faster convergence speed and higher convergence precision than the GA; due to the superior design feature and parallel structure, the MEA can memorize more evolutionary information and search the optimal solution more efficiently and effectively, overcoming the defects of the GA, such as precociousness, easiness of falling into the local extremum, and time-consuming computation.

Table 7: PID parameters and performances of industrial process control model IV.

Parameter	MEA	GA	Z-N
K_p	1.0968	1.2785	1.4220
K_i	0.2103	0.2769	0.2613
K_d	1.7656	1.9265	1.9346
t_s	16.1596	18.1772	20.0000
$ITAE$	15.7474	19.4274	21.3847

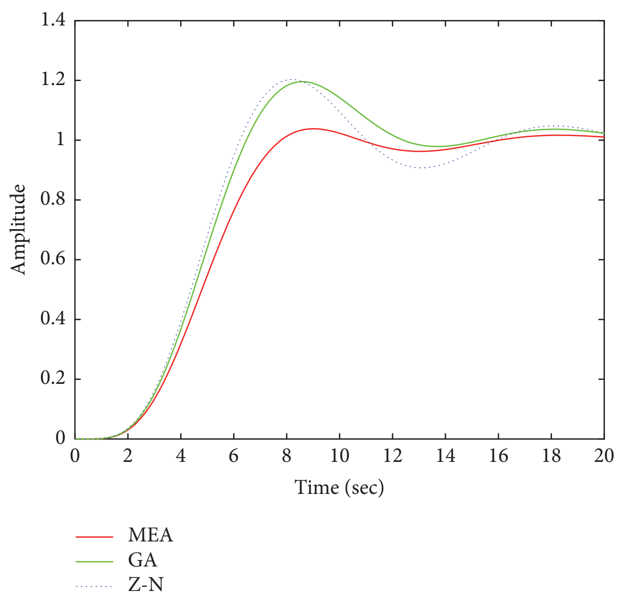

Figure 11: Closed-loop step response with process IV.

The simulation experiments of five typical industrial process control models also show that the MEA can capture better PID parameters (K_p, K_i, K_d) and lower $ITAE$; the MEA-tuning PID controller well realizes the tuning and optimization of the controlled object; the unstable controlled object gradually reaches a steady state; meanwhile, the MEA is superior to the GA and Z-N methods in speed and optimization performance.

Experiments indicated that the MEA-tuning method proposed in this study is feasible and valid, which offers a practical and novel approach for the design of the traditional PID controller. However, future work should focus on the comparison researches between the MEA and other intelligent algorithms; other PID controllers such as the PSO-PID controller, TSA-PID controller, BFA-PID controller, ACA-PID controller, and ABC-PID controller should be introduced into the tuning and optimization of the traditional PID controller and compared with the MEA-PID controller to verify its effectiveness and advantage. Moreover, the MEA approach should be further tested and developed through other practical application fields such as regression fitting, pattern recognition, and job planning and scheduling.

Abbreviations

PID: Proportional integral derivative
MEA: Mind evolutionary algorithm
Z-N: Ziegler and Nichols
GA: Genetic algorithm
PSO: Particle swarm optimization
TSA: Tabu search algorithm
BFA: Bacterial foraging algorithm
ACA: Ant colony algorithm
ITAE: Integral of time absolute errors
ISE: Integral of squared errors.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This research was funded by the National Social Science Fund of China (Grant No. 17BGL202).

References

[1] S. Bennett, "The past of PID controllers," *Annual Reviews in Control*, vol. 25, pp. 43–53, 2001.

[2] V. Chopra, S. K. Singla, and L. Dewan, "Comparative analysis of tuning a PID controller using intelligent methods," *Acta Polytechnica Hungarica*, vol. 11, no. 8, pp. 235–249, 2014.

[3] M. J. Er and Y. L. Sun, "Hybrid fuzzy proportional-integral plus conventional derivative control of linear and nonlinear systems," *IEEE Transactions on Industrial Electronics*, vol. 48, no. 6, pp. 1109–1117, 2001.

[4] M. Kushwah and A. Patra, "Tuning PID controller for speed control of DC motor using soft computing techniques-A review," *Advance in Electronic and Electric Engineering*, vol. 4, no. 2, pp. 141–148, 2014.

[5] C. Zou, C. Manzie, and D. Nesic, "A Framework for Simplification of PDE-Based Lithium-Ion Battery Models," *IEEE Transactions on Control Systems Technology*, vol. 24, no. 5, pp. 1594–1609, 2016.

[6] J. B. Odili, M. N. M. Kahar, and A. Noraziah, "Parameters-Tuning of PID controller for automatic voltage regulators using the African buffalo optimization," *PLoS ONE*, vol. 12, no. 4, p. e0175901, 2017.

[7] Y. Luo and Z. Chen, "Optimization for PID control parameters on hydraulic servo control system based on the novel compound evolutionary algorithm," in *Proceedings of the 2nd International Conference on Computer Modeling and Simulation (ICCMS '10)*, pp. 40–43, IEEE, January 2010.

[8] T. Hagglund and K. J. Astrom, "Revisiting the Ziegler-Nichols tuning rules for PI control," *Asian Journal of Control*, vol. 4, no. 4, pp. 364–380, 2002.

[9] J. G. Ziegler and N. B. Nichols, "Optimum settings for automatic controllers," *Journal of Dynamic Systems, Measurement, and Control*, vol. 46, no. 2B, pp. 220–222, 1993.

[10] A. Giwa and S. Karacan, "Decoupling PID control of a reactive packed distillation column," *International Journal of Engineering & Technical Research*, vol. 1, no. 6, 2012.

[11] M. Korkmaz, O. Aydogdu, and H. Dogan, "Design and performance comparison of variable parameter nonlinear PID controller and genetic algorithm based PID controller," in *Proceedings of the 2012 International Symposium on Innovations in Intelligent Systems and Applications (INISTA)*, pp. 1–5, IEEE, Trabzon, Turkey, July 2012.

[12] S. Y. S. Hussien, H. I. Jaafar, N. A. Selamat, F. S. Daud, and A. F. Z. Abidin, "PID control tuning via Particle Swarm Optimization for coupled tank system," *International Journal of Soft Computing & Engineering*, vol. 4, no. 2, pp. 202–206, 2014.

[13] M. W. Foley, R. H. Julien, and B. R. Copeland, "A comparison of PID controller tuning methods," *The Canadian Journal of Chemical Engineering*, vol. 83, no. 4, pp. 712–722, 2010.

[14] A. Harrag and S. Messalti, "Variable step size modified P&O MPPT algorithm using GA-based hybrid offline/online PID controller," *Renewable & Sustainable Energy Reviews*, vol. 49, pp. 1247–1260, 2015.

[15] X. H. Liu, X. H. Chen, X. H. Zheng, S. P. Li, and Z. B. Wang, "Development of a GA-Fuzzy-Immune PID Controller with Incomplete Derivation for Robot Dexterous Hand," *The Scientific World Journal*, vol. 2014, Article ID 564137, 10 pages, 2014.

[16] C.-H. Cheng, P.-J. Cheng, and M.-J. Xie, "Current sharing of paralleled DC-DC converters using GA-based PID controllers," *Expert Systems with Applications*, vol. 37, no. 1, pp. 733–740, 2010.

[17] S. Ahmadi, S. Abdi, and M. Kakavand, "Maximum power point tracking of a proton exchange membrane fuel cell system using PSO-PID controller," *International Journal of Hydrogen Energy*, vol. 42, no. 32, pp. 20430–20443, 2017.

[18] I. Atacak and B. Küçük, "PSO-based PID controller design for an energy conversion system using compressed air," *Tehnički vjesnik*, vol. 24, no. 3, pp. 671–679, 2017.

[19] Y. Ye, C.-B. Yin, Y. Gong, and J.-J. Zhou, "Position control of nonlinear hydraulic system using an improved PSO based PID controller," *Mechanical Systems and Signal Processing*, vol. 83, pp. 241–259, 2017.

[20] A. Ateş and C. Yeroglu, "Optimal fractional order PID design via Tabu Search based algorithm," *ISA Transactions*, no. 60, pp. 109–118, 2016.

[21] O. Roeva and T. Slavov, "PID controller tuning based on metaheuristic algorithms for bioprocess control," *Biotechnology & Biotechnological Equipment*, vol. 26, no. 5, pp. 3267–3277, 2012.

[22] A. Bagis, "Tabu search algorithm based PID controller tuning for desired system specifications," *Journal of The Franklin Institute*, vol. 348, no. 10, pp. 2795–2812, 2011.

[23] D. H. Kim, "Hybrid GA-BF based intelligent PID controller tuning for AVR system," *Applied Soft Computing*, vol. 11, no. 1, pp. 11–22, 2011.

[24] A. Shamel and N. Ghadimi, "Hybrid PSOTVAC/BFA technique for tuning of robust PID controller of fuel cell voltage," *Indian Journal of Chemical Technology*, no. 23, pp. 171–178, 2016.

[25] E. S. Ali and S. M. Abd-Elazim, "BFOA based design of PID controller for two area Load Frequency Control with nonlinearities," *International Journal of Electrical Power & Energy Systems*, no. 51, pp. 224–231, 2013.

[26] M. Ünal, H. Erdal, and V. Topuz, "Trajectory tracking performance comparison between genetic algorithm and ant colony optimization for PID controller tuning on pressure process," *Computer Applications in Engineering Education*, vol. 20, no. 3, pp. 518–528, 2012.

[27] A. S. Oshaba, E. S. Ali, and S. M. Abd Elazim, "PI controller design using ABC algorithm for MPPT of PV system supplying DC motor pump load," *Neural Computing and Applications*, vol. 28, no. 2, pp. 353–364, 2017.

[28] A. S. Oshaba, E. S. Ali, and S. M. Abd Elazim, "PI controller design for MPPT of photovoltaic system supplying SRM via BAT search algorithm," *Neural Computing and Applications*, vol. 28, no. 4, pp. 651–667, 2017.

[29] N. Yusup, A. M. Zain, and S. Z. M. Hashim, "Evolutionary techniques in optimizing machining parameters: review and recent applications (2007–2011)," *Expert Systems with Applications*, vol. 39, no. 10, pp. 9909–9927, 2012.

[30] A. K. Dwivedi, S. Ghosh, and N. D. Londhe, "Review and Analysis of Evolutionary Optimization-Based Techniques for FIR Filter Design," *Circuits, Systems and Signal Processing*, pp. 1–22, 2018.

[31] B. J. Saharia, H. Brahma, and N. Sarmah, "A review of algorithms for control and optimization for energy management of hybrid renewable energy systems," *Journal of Renewable and Sustainable Energy*, vol. 10, no. 5, p. 053502, 2018.

[32] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, pp. 21–57, 2014.

[33] Z. Li, W. Wang, Y. Yan, and Z. Li, "PS-ABC: A hybrid algorithm based on particle swarm and artificial bee colony for high-dimensional optimization problems," *Expert Systems with Applications*, vol. 42, no. 22, pp. 8881–8895, 2015.

[34] C. Sun, Y. Sun, and L. Wei, "Mind-evolution-based machine learning: framework and the implementation of optimization," in *Proceedings of the IEEE International Conference on Intelligent Engineering Systems (INES'98)*, pp. 355–359, 1998.

[35] C. Sun, X. Zhou, and W. Wang, "Mind evolutionary computation and applications," *Journal of Communication and Computer*, vol. 1, no. 1, pp. 13–21, 2004.

[36] X. Zhou and C. Sun, "Pareto-MEC and its convergence analysis," *Jisuanji Gongcheng/Computer Engineering*, vol. 33, no. 10, pp. 233–236, 2007.

[37] H. Liu, H. Tian, X. Liang, and Y. Li, "New wind speed forecasting approaches using fast ensemble empirical model decomposition, genetic algorithm, Mind Evolutionary Algorithm and Artificial Neural Networks," *Journal of Renewable Energy*, vol. 83, pp. 1066–1075, 2015.

[38] G. Yan, G. Xie, Y. Qiu, and Z. Chen, "MEA based nonlinearity correction algorithm for the VCO of LFMCW radar level gauge," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, Springer, Berlin, Germany, 2005.

[39] G. Li and W. H. Li, “Face occlusion recognition based on MEBML,” *Journal of Jilin University*, vol. 44, no. 5, pp. 1410–1416, 2014.

[40] K. J. Åström and T. Hägglund, *PID Controllers: Theory, Design And Tuning*, Instrument society of America, Research Triangle Park, NC, USA, 1995.