Applied Mathematics and Nonlinear Sciences

<https://www.sciendo.com>

Application of Sobolev-Volterra projection and finite element numerical analysis of integral differential equations in modern art design

Zheng Tan^{1†}, Abdullah Albarakati²¹ Academy of Fine Arts, Qilu Normal University, Jinan 250200, China² Department of Information Systems, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

Submission Info

Communicated by Juan Luis García Guirao

Received June 17th 2021

Accepted September 24th 2021

Available online November 22th 2021

Abstract

The article uses the Spss statistical analysis software to establish a multiple linear regression model of short-term stock price changes in domestic agricultural listed companies. It uses a stable time series based on the ARMA model for stable agricultural value-added, fiscal expenditure and market interest rates. The regression method is used to study its impact on the stock price index. Compared with the existing stock forecasting methods, this method has simple data collection and no specific requirements for data selection, and the prediction results have a high degree of fit. Therefore, this method is suitable for most stocks.

Keywords: K-GA algorithm, illustration art, art design, operator optimisation, cluster optimisation**AMS 2010 codes:** I5J21

1 Introduction

In the traditional art design process, if designers have inspiration and ideas, they must use tools to transfer their design to reality. This requires human labour, so the quantity and quality of creation can be limited. living. Moreover, multiple works cannot be produced at the same time, which is a disadvantage of traditional creative methods. Now, because computers provide the level of technology needed in the field of art creation, they can help designers develop their inspirations well, and they can use less time to realise their inspirations [1].

The so-called evolutionary design refers to the use of a new evolutionary computing method in the concept of computer design, which can be fully used in the field of design. The first computer evolution program ‘Blind Watchmaker’ in history was proposed by Rudolph; it is mainly used to simulate tree clusters. Fogarty puts

[†]Corresponding author.Email address: zhengtsub121@163.com

forward the theory that genetic algorithms (GAs) can be used in the improvement and optimisation of design, and explains the theoretical framework of the entire design [2]. His design system can be used for the production of vehicles and seats. The use of computer-aided evolutionary design in the field has opened a new door for evolutionary art. In addition to this, there are people who have applied computer simulation technology in more fields, such as table lamps and sculptures in artworks, making the application of this technology more extensive. There are also people who use GAs in the design of building plans. Andrew Row bottom wrote a program called Form, which for the first time turned evolutionary design into a three-dimensional (3D) representation [3]. Matthew Lewis used the technology of evolutionary models in the fields of compositing colours, making cartoon characters, changing fonts and designing exterior shapes of cars, and successfully created interactive design systems. Domestic research and development scholars such as Liu Hong, Tang Mingxi and Liu Xiyu use computers to support innovative design of appearance modelling. Sun Shouqian, Zhang Lishan, Huang Qi and others used CAD to realise the problems of colour expression, design sketches and structural patterns [4].

In order to optimise the standard GA and improve some existing shortcomings, this paper uses a design model of illustration art, and the model is based on cluster optimisation and operator GA, and the model is simulated through more innovative experiments [5].

2 Volterra integral equation

2.1 Volterra integral equations of the first kind

2.1.1 Linear Volterra integral equations of the first kind

These equations are shaped as

$$\int_a^x k(x, y) \phi(y) dy = f(x) \quad (1)$$

2.1.2 A solution to the first type of Volterra integral equation

In some cases, Volterra integral equations of the first type can usually be transformed into solutions of the Volterra integral equations of the second type. Generally, the two sides of the equation are differentiated when the equations $k(x, y)$, $f(x)$ are differentiable, and $k(x, y) \neq 0$, the equations of the first type are transformed. The form of the second type is:

$$\phi(x) + \int_a^x \frac{k'_x(x, y)}{k(x, y)} \phi(y) dy = \frac{f'(x)}{k(x, y)} \quad (2)$$

In this way, it can be solved by the second type of integral equation [6]. A special type of Volterra integral equation is the Abel equation, whose form is:

$$\int_a^x \frac{\phi(y)}{(x-y)^a} dy = f(x) \quad (3)$$

When $x = y$, the equation appears weakly singular. The solution can be based on the theorem assuming that the free term $f(x)$ of the Abel integral equation $\int_a^x \frac{\phi(y)}{(x-y)^a} dy = f(x)$ is continuously differentiable, and $f(a) = 0$, then it has a unique solution.

$$\phi(x) = \frac{\sin \pi a}{\pi} - \frac{d}{dx} \int_a^x \frac{f(y)}{(x-y)^{1-a}} dy \quad (4)$$

2.2 Volterra integral equations of the second kind

2.2.1 The second type of linear Volterra integral equation

The form of the second type of equation is:

$$\phi(x) - \lambda \int_a^x k(x, y) \phi(y) dy = f(x) \quad (5)$$

where $\phi(x)$ is the required unknown function and λ is a known or need to be discussed parameter [7]. Like Fredholm's equation, $k(x,y)$ is a known function, called the kernel of Volterra equation. When $k(x,y) = 0$, Volterra equation can be regarded as a special form and the theory of Fredholm equation is suitable for the Volterra equation. The Volterra equation has its own characteristics. For example, it has no eigenvalues, and it has solutions to arbitrary free terms. The Volterra equation of the first type can be transformed into the second type under certain conditions. Like the linear Fredholm integral equations of the second type, the linear Volterra integral equations of the second type also have their own iterations and solutions [8]. The methods for deriving the iterations and solutions are the same as those of the second type of Fredholm [9].

Iteration: Hypothesis

$$k_2(x,y) = \int_y^x k(x,t)k(t,y)dt \quad (6)$$

Then:

$$\phi_2(x) = \int_a^x k_2(x,y)f(y)dy \quad (7)$$

$$\phi_n(x) = \int_a^x k_n(x,y)f(y)dy \quad (8)$$

We call $k_n(x,y)$ as the iterative kernel of the Volterra equation. We call $R(x,y;\lambda) = \sum_{n=1}^{\infty} \lambda^{n-1}k_n(x,y)$ a de-nuclear. If the overlapping kernel of the equation is known, the solution kernel and the solution of the equation can be obtained.

2.2.2 The solution of the second type of linear Volterra integral equation:

Suppose the equation has a solution of the form

$$\phi(x) = \phi_0(x) + \phi_1(x)\lambda + \cdots + \phi_n\lambda^n = \sum_{i=0}^n \phi_i(x)\lambda^i \quad (9)$$

If the equation has a solution for the successive method, solving the equation generally makes

$$\begin{aligned} \phi_0 &= f(x) \\ \phi_1 &= f(x) + \int_a^x k(x,y)\phi_0 dy \\ \phi_2 &= f(x) + \int_a^x k(x,y)\phi_1 dy \\ \phi_n &= f(x) + \int_a^x k(x,y)\phi_{n-1} dy \end{aligned} \quad (10)$$

Then, for the above series to converge, that is, for series $\sum_{i=0}^n \phi_i(x)\lambda^i$, it can be proved that there is a solution for any parameter λ equation, according to the theorem: if the kernel $k(x,y)$ and the free term $f(x)$ are continuous real functions [10]. Then the second type Linear Volterra equation is

$$\phi(x) - \lambda \int_a^x k(x,y)\phi(y)dy = f(x) \quad (11)$$

There is a unique continuous solution for any parameter λ , and the solution can be obtained by successive approximation.

Iteration: Hypothesis

$$k_2(x,y) = \int_y^x k(x,t)k(t,y)dt \quad (12)$$

Then

$$\begin{aligned}\varphi_2(x) &= \int_a^x k_2(x,y)f(t)dt \\ &\dots \\ \varphi_n(x) &= \int_a^x k_n(x,y)f(y)dy\end{aligned}\tag{13}$$

We call $k_n(x,y)$ the iterative kernel of the Volterra equation. We call $R(x,y;\lambda) = \sum_{n=1}^{\infty} \lambda^{n-1} k_n(x,y)$ the solution kernel. If the overlapping kernels of the equation are known, the solution kernel can be obtained and the solution of the equation can be obtained [11].

2.2.3 Nonlinear Volterra integral equations of the second kind

The form of the nonlinear Volterra integral equation of the second kind is as follows:

$$\phi(x) - \lambda \int_a^x k(x,y)F(\phi(y))dy = f(x)\tag{14}$$

The unknown function is $\phi(x)$, and $f(x)$, $k(x,y)$, $F(x)$, and are all known. When the equation meets certain conditions, it can be solved by successive approximation. For the first type of nonlinear Volterra integral equation, it can be solved by transforming into the second type of nonlinear Volterra integral equation [12]. For the specific conversion process, see the numerical solution of the nonlinear Volterra integral equation [13].

2.3 The solution of convolutional Volterra equation

2.3.1 Solutions to the convolution volterra integral equation of the second kind

It is shaped as:

$$\phi(x) = f(x) + \int_a^x k(x-y)\phi(y)dy\tag{15}$$

Called the second type of convolution Volterra integral equation, this type of equation is generally solved by Laplace transform. If $f(x)$, $k(x,y)$ in the equation is a sufficiently smooth, exponential order function, the solution of the equation is also exponential order, so that the equation can be solved by Laplace transform. Let $\varpi\{k(x)\} = K(p)$, $\varpi\{f(x)\} = F(p)$, $\varpi\{\phi(x)\} = \phi(p)$. and Laplace transform the two sides of the equation; we can get. $\phi(p) = F(p) + K(p)\phi(p)$ solve $\phi(p) = \frac{F(p)}{1-K(p)}$. and when $K(p) \neq 1$, $\phi(x) = \varpi^{-1}\left\{\frac{F(p)}{1-K(p)}\right\}$.

For the first type of Volterra integral equation, that is, Equation $\int_a^x k(x-t)\phi(t)dt = f(x)$ also performs Laplace transformation on both sides of the equation, and can be solved for $\phi(p) = \frac{F(p)}{K(p)}$, so the solution of the equation is $\phi(x) = \varpi^{-1}\left\{\frac{F(p)}{K(p)}\right\}$.

The solution of the nonlinear Volterra integral equation Laplace transform is

$$\phi(x) = f(x) + \lambda \int_0^x \phi(y)\phi(x-y)dy\tag{16}$$

Let $\varpi\{\phi(x)\} = \phi(p)$, $\varpi\{f(x)\} = F(p)$ and Laplace transform both sides of the equation, we can get

$$\phi(x) = \varpi^{-1}\left\{\frac{1 \pm \sqrt{1-4\lambda F(p)}}{2\lambda}\right\}\tag{17}$$

When $\frac{1 \pm \sqrt{1-4\lambda F(p)}}{2\lambda}$ exists, the solution is the Laplace transform of the nonlinear Volterra integral equation.

3 Knowledge of integral numerical solution

3.1 Newton-Cotes integral quadrature formula

Here we mainly discuss the numerical calculation of $\int_a^b f(x)dx$. It can be assumed that $f(x)$ is integrable on $[a, b]$. In some cases, the function $f(x)$ is not integrable and can be represented by an elementary function. Therefore, here we will study the numerical solution of function integration.

To calculate the integral $I(f) = \int f(x)W(x)dx$, where $W(x)$ is a weight function, you can assume that $f(x)$ is at $n+1$ different points: the values of E $a \leq x_1 < x_2 < \dots < x_{n+1} \leq b$ are: $f(x_1), f(x_2), \dots, f(x_{n+1})$ and you can use the linear combination of $f(x_1), f(x_2), \dots, f(x_{n+1})$ to get the integral An approximate solution of $I_n(f) \approx I(f)$, where $I_n = \sum_{i=1}^{n+1} A_i f(x_i)$ where $E_n(f)$ is the discrete error:

$$\begin{aligned} A_i &= \int_a^b l_i W(x) dx \\ l_i(x) &= \frac{w_{n+1}}{(x-x_i)w'_{n+1}(x_i)} \\ w_{n+1}(x) &= (x-x_1)(x-x_2)\dots(x-x_{n+1}) \\ i &= 1, 2, \dots, n+1 \end{aligned} \quad (18)$$

When we assume that $[a, b]$ is a finite interval, $W(x) = 1$ and divide the interval into n equal parts, take the equidistant base point as:

$$a = x_1 < x_2 < \dots < x_{n+1} = b,$$

the step length is $h = x_{i+1} - x_i = \frac{b-a}{n}$, and according to the difference quadrature formula obtained above, the Newton-Cotes integral quadrature formula can be obtained, that is, $I_n(f) = \sum_{i=1}^{n+1} A_i f(x_i)$, where

$$A_i = \int \frac{w_{n+1}(x)}{(x-x_i)w'_{n+1}(x_i)} dx = (-1)^{n+1-i} \frac{h}{(i-i)!(n+1-i)!} \int_0^n t(t-1)\dots(t-(i-2))(t-i)\dots(t-n) dt \quad (19)$$

In the Newton-Cotes integral quadrature formula, when $n = 1$, a trapezoidal formula can be obtained, let $x_1 = a, x_2 = b$ be a formula based on the Newton-Cotes integral quadrature formula:

$$I_1(f) = \frac{b-a}{2} (f(a) + f(b)),$$

where $A_1 = \frac{b-a}{2}, A_2 = \frac{b-a}{2}$. If you set $n = 2$, you can get the Simpson formula.

3.2 Compound trapezoidal formula

We assume that the localisation of the function in the integral in question is $[a, b]$, and take $n+1$ mutually distinct base points in this interval, that is, $a = x_1 < x_2 < \dots < x_{n+1} = b$, And take the step size $h = x_{i+1} - x_i = \frac{b-a}{n}$. Use the body size formula on the subinterval $[x_{i+1}, x_i]$, so

$$\int_a^b f(x)dx = \sum_{i=1}^n \int_{x_i}^{x_{i+1}} f(x)dx = \frac{h}{2} \sum_{i=1}^n [f(x_i) + f(x_{i+1})] - \frac{h}{12} \sum_{i=1}^n f''(\xi_i) \quad (20)$$

This leads to:

$$\int_a^b f(x)dx = \frac{h}{2} \left[f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(a+ih) \right] - \frac{h}{12} \sum_{i=1}^n f''(\xi_i) \quad (21)$$

Rounding out the $\frac{h}{12} \sum_{i=1}^n f''(\xi_i)$ term, we get the compound trapezoidal formula:

$$\int_a^b f(x)dx = \frac{h}{2} \left[f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(a+ih) \right] \quad (22)$$

4 Numerical solution method of integral equation

4.1 Unknown function expansion method

Here, we will discuss the role of a complete function system in $L_2(a, b)$ in solving approximate equations [14]. These function systems can be orthogonal or non-orthogonal. You can use the finite terms of a function system as an equation the approximate value of the solution. Let the function system be $\{\varphi_i\}_{i=1}^n$, where each function of the function system is linearly independent. You can make the solution of the integral equation $\phi(x) \approx \sum_{i=1}^n c_i \varphi_i$, and substitute the approximate function with the integral equation:

$$\phi(x) - \lambda \int_a^x k(x, y) \phi(y) dy = f(x) \quad (23)$$

So, you get

$$\sum_{i=1}^n c_i \varphi_i \approx \lambda \sum_{i=1}^n c_i \int_a^x k(x, y) \varphi_i dy + f(x) \quad (24)$$

Organise it into:

$$\sum_{i=1}^n c_i \varphi_i - \lambda \sum_{i=1}^n c_i \int_a^x k(x, y) \varphi_i dy - f(x) = R(x) \quad (25)$$

where $R(x)$ is the residual; if $R(x)$ can be equal to zero, then the approximate solution of the equation is equal to the exact solution of the equation, but in general, it is difficult to make $R(x)$ equal to zero [15]. Generally, it can be ignored when $R(x)$ is small. A numerical solution of the equation is obtained as:

$$\sum_{i=1}^n c_i \varphi_i = \lambda \sum_{i=1}^n c_i \int_a^x k(x, y) \varphi_i dy + f(x) \quad (26)$$

However, different requirements for residuals can lead to different solutions. Generally, these are the following solutions: collocation method [16], Galerkin method and least square method. If the residual is required to satisfy $R(x_k)$ equal to zero at the selected base point, where $\{x_k\}_{k=1}^n$ is some different base points, then we can get the following equations:

$$\sum_{i=1}^n c_i \varphi_i(x_k) - \lambda \sum_{i=1}^n c_i \int_a^{x_k} k(x_k, y) \varphi_i(y) dy = f(x_k) \quad k = 1, 2, \dots, n \quad (27)$$

Solving the equations can obtain the expansion coefficient $\{c_i\}_{i=1}^n$. For the moment method used in the Fredholm integral equation,

$$\phi(x) - \lambda \int_a^b k(x, y) \phi(y) dy = f(x) \quad (28)$$

The moment method requires that the moment of the residual from the origin to the n th order is zero, that is, $\int_a^b R(x) x^k dx = 0$, and the following equations can be obtained:

$$\sum_{i=1}^n c_i \int_a^b \varphi_i(x) x^k dx - \lambda \sum_{i=1}^n c_i \int_a^b \left[\int_a^b k(x, y) \varphi_i(y) dy \right] x^k dx = \int_a^b f(x) x^k dx \quad (29)$$

Solving this equation can get the expansion coefficient $\{c_i\}_{i=1}^n$. The Galerkin method requires that the residual function $R(x)$ in the square integrable space, that is, the inner product of space $L^2[a, b]$ and function φ_i is zero, that is, $\int_a^b R(x) \varphi_i dx = 0, i = 1, 2, \dots, n$. So the expansion coefficient can be used to determine $\{c_i\}_{i=1}^n$ by taking the first n functions $\varphi_i (i = 1, 2, \dots, n)$ in the function system and the integral equation on $[a, b]$.

$$\sum_{i=1}^n c_i \varphi_i(x) - \lambda \sum_{i=1}^n c_i \int_a^b k(x, y) \varphi_i(y) dy = f(x_k) \quad (30)$$

The two ends are orthogonal, so that $\phi^n = c_i \varphi_i$ expansion coefficient satisfies the following linear equations:

$$(\phi^n(x), \varphi_i(x)) = \lambda \left(\int_a^b k(x, y) \phi^n dy, \varphi_i(x) \right) + (f(x), \varphi_i(x)) a \quad i = 1, 2, \dots, n \quad (31)$$

where

$$((f(x), g(x)) = \int_a^b f(x) g(x) dx \quad (32)$$

Solve the system of equations to get the expansion coefficient $\{c_i\}_{i=1}^n$

4.2 Expansion of integral kernel series

The integral kernel series expansion method, also known as the degenerate kernel approximation method, uses a certain expansion method to expand the non-degraded kernel into an approximately degraded kernel. The general expansion methods include Taylor series and Fourier series expansions. A linearly independent function system for the approximate expansion of a known function in $L^2[a, b]$ space. If Taylor expansion is used, then care should be taken to retain the appropriate number of series terms. Generally, the series terms should be determined according to the size of the integration limit. There are also solutions to unknown functions that expand the unknown function to the integral equation.

The following estimation methods are used to approximate the error of the integral equation kernel with a degenerate kernel:

Theorem. [17] Let $k(x, y)$ be the approximate degenerate kernel of the integral equation kernel, and satisfy the conditions for the degenerate kernel

$$\int_a^b |k(x, y) - \tilde{k}(x, y)| dt < h \quad (33)$$

The solution kernel $R(x, y; \lambda)$ of the integral equation with the degenerate kernel $\tilde{k}(x, y)$ as the integral kernel holds that

$$\int_a^b |R(x, y; \lambda)| dt < R \quad (34)$$

Integral equation

$$\phi(x) = \lambda \int_a^b k(x, y) \phi(y) dy \quad (35)$$

Of solution $\phi(x)$ and solution $\tilde{\phi}(x, y)$ of the integral equation replaced with an approximate degenerate kernel, satisfying

$$|\tilde{\phi}(x) - \phi(x)| < \frac{B|\lambda|(1 + |\lambda|R)^2 h}{1 - |\lambda|h(1 + |\lambda|R)} \quad (36)$$

In the formula, B is an upper bound of $f(x)$.

5 Numerical solution of nonlinear Volterra integral equation

$$\phi(x) + \lambda \int_a^x k(x, y) F(\phi(y)) dy = f(x) \quad (37)$$

We assume that $f(x)$, $k(x, y)$, $F(x)$ are continuous functions in their domains, and use numerical evaluation formulas

$$\phi_i - \lambda \sum_{m=1}^i A_m K_{im} F(\phi_m) = f_i \quad i = 1, 2, \dots, n, \quad (38)$$

$\phi_i = \phi(i)$, A_m is the weight coefficient in the numerical integration formula, $K_{im} = k(x_i, x_m)$, $f_i = f(x_i)$. The system of equations is a system of lower triangular equations of order n , $\phi(a) = f(a)$. With this we can solve n numerical solutions along the top of the system of equations, so we get the nearsighted solution of the equation

$$\phi(x) = \lambda \sum_{m=1}^i A_m k(x_i, x_m) F(\phi_m) + f(x_i) \quad i = 1, 2, \dots, n \quad (39)$$

When n approaches infinity, the solution also tends to be exact.

We divide the integration interval $[a, b]$ into n equal small intervals $[x_i, x_{i+1}]$, $i = 1, 2, \dots, n$, where we take the step size as

$$h = \frac{b-a}{n} \quad (40)$$

Compound trapezoidal formula:

$$T_n(f) = \frac{1}{2} \left[f(a) + f(b) + 2 \sum_{i=1}^{n-1} f(a + ih) \right] \quad (41)$$

We apply this formula to the integral equation and we get

$$\phi(x_i) - \lambda \left[\frac{h}{2} (k(x_i, x_1) F(\phi(x_1)) + k(x_i, x_i) F(\phi(x_i)) + 2 \sum_{j=1}^{i-1} k(x_i, x_{j+1}) F(\phi(x_{j+1}))) \right] = f(x_i) \quad (42)$$

In this way, we can get a system of equations of the lower triangle row. By touching the solution of this system of equations, we can get n numerical solutions of the equation, and then fitting by interpolation, we can get an approximate solution.

First, in a n -dimensional space, assuming a linearly independent set of basis $\{e_i(x)\}_{i=1}^n$ can make the unknown approximate solution $\phi_n(x) = \sum_{i=1}^n c_i e_i(x)$, as long as the value of $\{c_i\}_{i=1}^n$ can be solved, the approximate solution of the equation can be obtained. Then the projection $V(x) \in [a, b]$ of $V(x)$ in the n -dimensional coordinate system is $p_n V(x) = \sum_{i=1}^n V(x_i) e_i(x)$, x_i which is the interpolation point, so

$$\phi_n(x) = p_n \phi(x) = \sum_{i=1}^n c_i e_i \quad (43)$$

Multiply both sides of the equation by p_n to get:

$$p_n \phi(x) + p_n \int_a^x k(x, y) F(\phi(x)) dy = p_n f(x) \quad (44)$$

Putting formula (41) into (42), we get

$$\begin{aligned}
 \sum_{i=1}^n c_i e_i(x) + \sum_{i=1}^n \int_a^x k(x_i, y) F(\varphi(x)) e_i(x) dy &= \sum_{i=1}^n f(x_i) e_i(x) \\
 \sum_{i=1}^n \left[c_i e_i(x) + e_i(x) \int_a^{x_i} k(x_i, y) F(\varphi(y)) dy - f(x_i) e_i(x) \right] &= 0 \\
 \sum_{i=1}^n e_i(x) \left[c_i + \int_a^{x_i} k(x_i, y) dy - f(x_i) \right] &= 0 \\
 c_i + \int_a^{x_i} k(x_i, y) F(\varphi(y)) dy - f(x_i) &= 0 \\
 c_i + \sum_{j=1}^i k(x_i, y) F(c_i e_i(y_j)) A_n - f(x_i) &= 0
 \end{aligned} \tag{45}$$

6 Case study

6.1 Algorithm simulation

In order to verify the feasibility of the algorithm, the unimodal and multimodal functions are used to improve the standard GA and the improvements proposed in this paper. The algorithm K-GA is tested. The unimodal function is as follows:

$$\min f_1(x, y) = 100 \cdot (y - x^2)^2 + (x - 1)^2 \tag{46}$$

The multimodal function is as follows:

$$\max f_2(x, y) = 21.5 + x \sin(4\pi x) + y \sin(20\pi y) \tag{47}$$

The average running algebraic test results of the standard GA and the improved algorithm proposed in this paper are shown in Table 1.

Table 1 Algorithm simulation results of test functions.

Function method		f_1		f_2	
		GA	K-GA	GA	K-GA
Cross probability	1	56.28	7.32	12.67	4.71
	0.9	57.03	7.84	13.61	5.38
	0.8	60.05	8.09	13.88	6.37
	0.7	61.44	8.67	15.58	6.94
	0.6	62.49	9.52	19.71	7.54
	0.5	63.58	10.49	20.85	8.06

It can be seen from Table 1 that for the two functions, when the crossover probability is set to 1, for any one test function, the average running generation number is the least, and the average running generation number decreases gradually as the crossover probability increases. Therefore, the greater the crossover probability, the greater the probability of producing outstanding individuals, which improves the performance of the GA.

6.2 Case analysis

Taking the illustration design of the hexagon fractal as an example, the design results using the standard GA and the improved algorithm proposed in this paper are shown in Figures 1 and 2:

Fig. 1 Fractal image illustration art design of standard algorithm.

Fig. 2 Artistic design of fractal image illustration with improved algorithm.

By comparing Picture 1 and Picture 2, we can find that the optimised algorithm mentioned in the article can be used in illustration art and is innovative.

7 Summary

Until now, the development of computer technology has continued to advance globally, and the use of evolutionary technology to help product innovation and design is an important way. How to improve the algorithm of this computer technology and how this technology can be better used and practiced in the field of design has been valued and studied by more and more people. In this article, I wrote out some shortcomings of some standard GAs, and conducted an illustration art design experiment on the operator and cluster optimisation GA. It was explained after the experiment that the algorithm after optimisation is more creative than design algorithms without optimisation.

Acknowledgements.

The authors acknowledge the study supported by ‘Anhui Quality Engineering Teaching Innovation Team Project’, China (Grant: 2019 cxt100); ‘2014 Anhui Provincial Department of Education Outstanding, by ‘Faculty projects for 2019’, China (Grant: 2019 cyxy 07); ‘Faculty projects for 2019’, China (Grant: 2019 jnjs 10); and ‘Faculty projects for 2019’, China (Grant: 2019 jngzs 03).

References

- [1] Urve Kangro. 2017. Cordial Volterra integral equations and singular fractional integro-differential equations in spaces of analytic functions. *Mathematical Modelling & Analysis*, 22(4), pp. 548-567.
- [2] Ermine Oganessovna Azizian, & Khachatur Aghavardovich Khachatryan. 2016. One-parametric family of positive solutions for a class of nonlinear discrete hammetstein-volterra equations. *Ufa Mathematical Journal*, 8(1), pp. 13-19.

- [3] Yunxia Wei, Yanping Chen, & Yunqing Huang. 2018. Legendre collocation method for Volterra integro-differential algebraic equation. *Journal of Scientific Computing*, 19(3), pp. 672-688.
- [4] Hounie, Jorge, & Picon, Tiago. 2016. L1 sobolev estimates for (pseudo)-differential operators and applications. *Mathematische Nachrichten*, 289(14-15), pp. 1838-1854.
- [5] Ibrahim, R. W., Ahmad, M. Z., & Mohammed, M. J. 2016. Periodicity and positivity of a class of fractional differential equations., 5(1), pp. 824.
- [6] M. A Abdou, & S. A Raad. 2016. Nonlocal solution of a nonlinear partial differential equation and its equivalent of nonlinear integral equation. *Journal of Computational & Theoretical Nanoscience*, 13(7), pp. 4580-4587.
- [7] GERT AUDRING. 2017. Isomorphism theorems for some parabolic initial-boundary value problems in hörmander spaces. *Open Mathematics*, 15(1), pp. 57-76.
- [8] Min Cai, & Changpin Li. 2019. Regularity of the solution to riesz-type fractional differential equation. *Integral Transforms and Special Functions*, 30(2), pp. 1-32.
- [9] Mohammad ZAREBNIA, Reza PARVAZ, & Amir SABOOR BAGHERZADEH. 2018. Deviation of the error estimation for Volterra integro-differential equations. *Acta Mathematica Scientia (English Series)*, 38(4), pp. 1322-1344.
- [10] Shaofei Wu, 2015. A Traffic Motion Object Extraction Algorithm, *International Journal of Bifurcation and Chaos*, 25(14), pp. 1540039.
- [11] Shaofei Wu, Mingqing Wang, Yuntao Zou. 2018. Research on internet information mining based on agent algorithm, *Future Generation Computer Systems*, 86, pp. 598-602.
- [12] Shaofei Wu, 2019. Nonlinear information data mining based on time series for fractional differential operators, *Chaos*, 29, pp. 013114.
- [13] M. Salai Mathi Selvi, L. Rajendran. 2019. Application of modified wavelet and homotopy perturbation methods to nonlinear oscillation problems, *Applied Mathematics and Nonlinear Sciences*, 4(2), pp. 351-364.
- [14] Faruk Dusunceli. 2019. New Exact Solutions for Generalized (3+1) Shallow Water-Like (SWL) Equation, *Applied Mathematics and Nonlinear Sciences*, 4(2), pp. 365-370.
- [15] Shailaja Shirakol, Manjula Kalyanshetti, Sunilkumar M. Hosamani. 2019. QSPR Analysis of certain Distance Based Topological Indices, *Applied Mathematics and Nonlinear Sciences*, 4(2), pp. 371-386.
- [16] P. F. Zhou, Q. Fan, J. Zhu, Empirical Analysis on Environmental Regulation Performance Measurement in Manufacturing Industry: A Case Study of Chongqing, *Applied Mathematics and Nonlinear Sciences*, 2020. 5(1):pp. 25-34.
- [17] D. K. Josheski, E. Karamazova, M. Apostolov, Shapley-Folkman-Lyapunov theorem and Asymmetric First price auctions, *Applied Mathematics and Nonlinear Sciences*, 2019. 4(2):pp. 331-350.

This page is intentionally left blank