

UNIVERSITY OF WALES TRINITY SAINT DAVID

Gravitational Search Algorithm applied in Grey Model

by

Hailian Fang

A project submitted in partial fulfillment for the
degree of M.Sc.Applied Computing

in the
School of Applied Computing
Faculty of Architecture, Computing and Engineering

Supervisor: Gordon Dickers

September 2017

Declaration of Authorship

I, [Hailian Fang](#) declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed:

Date:

Form of Consent

I [Hailian Fang](#) hereby consent that my Project, submitted in candidature for the M.Sc.Applied Computing degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed:

Date:

Contents

Declaration of Authorship	i
Form of Consent	ii
List of Figures	v
1 Introduction	1
1.1 Aims and Objectives	2
2 literature Review	3
2.1 Grey Model	4
2.1.1 Mean absolute percentage error	7
2.2 Particle Swarm Optimization Algorithm	8
2.3 Gravitational Search Algorithm	10
3 Research Methods	13
3.1 Philosophy	13
3.2 Experiment design	13
3.2.1 Experiment hypothesis	13
3.2.2 Experiment apparatus	13
3.3 Any other software methodology	14
4 Development Methods	15
4.1 The coding achievement of GSA itself	15
4.2 Experiment	18
4.2.1 The first grey model is chosen in this paper	18
4.2.1.1 Testing data	19
4.2.1.2 Fitting and forecasting with EXCEL	19
4.2.1.3 Fitting and forecasting used classical GM(1,1)	20
4.2.1.4 Fitting and forecasting used GSA-GM(1,1, λ)	26
4.2.1.5 Fitting and forecasting used rolling GM(1,1)	27
4.2.1.6 Fitting and forecasting used GSA-rolling GM(1,1, λ)	28
4.2.2 The second grey model is chosen in this paper	30
4.2.2.1 Testing data	31
4.2.2.2 Fitting with EXCEL	31
4.3 Data Evaluation	36

5	Summary	39
6	AppendixA	41
7	AppendixB	45
8	AppendixC	47
9	AppendixD	49
	Bibliography	51

List of Figures

2.1	Every mass accelerate toward the result force that act it from the other masses.After [1]	10
2.2	GSA minization principle flow After [2]	11
4.1	The original figure in paper [1]	16
4.2	Sphere model optimization with GSA	16
4.3	The original figure in paper [1]	17
4.4	Quartic model optimization with GSA	17
4.5	The original figure in paper [1]	18
4.6	Ackley function optimization with GSA	18
4.7	X-Y scatter chart of original data	22
4.8	X-Y scatter chart of original data with exponential function	23
4.9	X-Y scatter chart of original data with polynomial function	24
4.10	The fitting results in original papre	25
4.11	The schematic diagram of GSA-GM(1,1)	26
4.12	The results in original paper	28
4.13	The fitting result of Actual1 data	32
4.14	The fitting result of Actual2 data	33
4.15	The fitting result of Actual3 data	34
4.16	The original result in paper [3] of Actual1 data	34
4.17	The original result in paper [3] of Actual2 data	34
4.18	The original result in paper[3] of Actual3 data	34
6.1	massCalculation.m	42
6.2	Gconstant.m	42
6.3	test_functions.m	43
6.4	test_functions_range.m	43
6.5	main.m	44
7.1	The main coding of GM(1,1)	45
7.2	The main running coding of GM(1,1)	45
7.3	MAPE function coding of GM(1,1)	46
8.1	MAPE function of FGM(q,1).m	47
8.2	test_functions.m in FGM(q,1)	48
8.3	test_functions_range.m in FGM(q,1)	48
9.1	test_function with 3-dimensional.m	49
9.2	test_functions_range with 3-dimensional.m	50

9.3 MAPE in the new grey model.m	50
--	----

Chapter 1

Introduction

With the rapid development of science and technology, people can get the efficient control of the occurrence likelihood in the future. For example, people can observe the temperature, humidity, wind and speed in this week to predict the weather today or tomorrow. According to the size and complexity of the dataset, it can select the appropriate methods to fit and forecast data. If the dataset size is very big such as thousands of records, it is hard to analyze it manually. There are a great deal of data mining tools now such as Weka and Tableau software. In Weka software, there are predicting tools such as "Classify", "Cluster", "Associate", etc. In each tool, there are different algorithms such as J48, Logistic, LinearRegression, XMeans, Tree, etc. However, before using Weka tools to analyze data, it need give a large amount of preprocessing. Sometimes, it is complexed. Compared to it, The Tableau software is easy to use. EXCEL file can be imported directly into the Tableau. It also can curve and fit the data. However, they are all tools for handling the "big data". If the dataset size is small, there are also available tools to analyze such as EXCEL software. In EXCEL software, there are ready-made fitting and forecasting tools such as exponential function, polynomial function, power function, etc. However, if the dataset nature is not obvious, complexed and limited. The conventional regression functions cannot fit with high accuracy. Scholars are seeking for ways to solve it. Among them, Deng et al proposed grey theory model to solve the fitting and forecasting of this kind of dataset in 1982[4]. In grey model, there are some parameters which need to be set to get minimum error. Therefore, researchers used different optimization algorithms [5–8] to estimate parameters to achieve it. Most of them [3, 9–16] prefer to use Particle Swarm Algorithm(PSO), which is focused to search the optimal of the objective functions to find the corresponding parameters. In grey model problems, the error function is set as the objective function.

Compared the PSO, Gravitational Search Algorithm(GSA) is also a new deterministic population-based heuristic optimization algorithm based on the law of gravity, which

applied in various fields for parameters controlling. Moreover, it can see Rashedi et al[1] showed the GSA convergence and the optimize value is better than PSO based on some particular benchmark function such as Sphere model, Schwefel's problem 2.22,1.2,2.21, Generalized Rosenbrock function,etc. However, as yet no one has used GSA to optimize parameters in grey model.

In this paper, considering the similarity of PSO and GSA and the result in reference [1], it will firstly introduced GSA in grey model for parameters controlling. Moreover, for existing dataset which used PSO, it will use the EXCEL regression tools to fit and forecast the data and use GSA to replace PSO to fit and forecast based the same conditions. Then, it will compare these three results for the same data and give the corresponding description. In addition, it will combine the characteristic of GSA and grey model theory to improve the fitting and forecasting result of GSA.

Firstly, it will check the GSA coding itself. Secondly, it will choose the China's iron dataset in [14] to fit and forecast with the EXCEL tool and GSA. Thirdly, it will fit the three datasets in [3] with EXCEL and GSA. Fourth, it will combine these two grey models in these two papers above [3, 14] to improve the accuracy.

1.1 Aims and Objectives

Aim:

Try to use Gravitational Search Algorithm (GSA) to replace the Particle Swarm Optimization (PSO) in grey model. Moreover, it will combine the GSA characteristic to improve the accuracy of the grey model.

Objectives:

- To achieve the implementation of GSA at first and compare the existing result in [1].
- List different kinds of testing data and use traditional regression methods and grey model with GSA respectively.
- Compare the existing results with PSO and new results with GSA.
- In order to solve high dimensional optimization problems, people proposed GSA. Therefore, in this paper, it will expand grey model dimensions to get better use of GSA to improve the fitting accuracy.

Chapter 2

literature Review

In the classical optimization theory, scholars always use local search to find the optimal. In the local search, set the initial solution and iteration maximum. Then it can search the optimal around the initial solution. However, it can see that the optimal is related to the initial solution, if the initial solution is far away from the optimal, it is hard to search the optimal in the limiting iterations. Therefore, scholars proposed the heuristic optimization algorithms to solve global optimal problem instead of setting the initial solution in local search. However, in optimization algorithms, it is necessary to choose suitable parameters and number of parameters. If there are a great deal of parameters, it can fit the model well while it is only useful to this model which called "overfitting"; if there are a small deal of parameters, it cannot fit the model well, the accuracy is low. In addition, people prefer to use classical regression methods to find the nature of the data so that it can fit and predict the data to get the efficient control of the occurrence likelihood in the future. There are some popular and simple classical regression methods such as using existing excel or MATLAB regression tools to get linear fitting and predicting. If the data curve is not similar as the linear trends, it can choose other functions as fitting functions such as exponential function, logarithmic function, power function, etc. In this process, it need to use some particular mathematical transformation methods such as differential or logarithm or variable replacement to transfer nonlinear to be the linear function. However, it need to analyze the data nature at first so that it can choose function with the similar trends to fit. Moreover, if the data size is very limited and the nature is hard to judge. Obviously, the fitting or predicting error will be bad. Therefore, scholars proposed the grey model[4] to solve this problem as follow.

2.1 Grey Model

Grey model theory was proposed by Deng in 1982 [4] for prediction, where "grey" means partial data information is known and partial data information is unknown. As far as the data information is concerned, if all information is known, it called "white"; if all information is unknown, it called "black". In the grey system, it is difficult to find some regular patterns in the raw sequence to predict due to nature of grey information. There are some methods to make the data nature more clearly such as using accumulated generating operator [17] and differential equation [18] to handle the initial sequence. That handling process is called the process of whitening raw sequence. The classical grey model GM(1,1), it means the differential order is 1 which is the first "1" and the other "1" is the number of variable. Since its inception, it has been widely used in technology, engineering, economy, industrial and many other fields for forecasting[19–23]. For example, deaths numbers in road traffic accidents predicting in China[24], tourism demand predicting in United States[25], lifetime prediction of electronic equipment forecasting example[26], etc. There are three main forms of expression [27] for GM(1,1) as follow:

For an raw sequence $X^{(0)} = (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n))$, where $x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n)$ are arranged in chronological order which means $x^{(0)}(1)$ is the oldest information and $x^{(0)}(n)$ is the latest information. A 1th-order accumulated generating sequence (1-AGO) can be defined as follow:

Definition 2.1. Let sequence $X^{(1)} = (x^{(1)}(1), x^{(1)}(2), \dots, x^{(1)}(n))$ be the 1-AGO accumulated generating sequence of $X^{(0)}$, where $x^{(1)}(k) = \sum_{i=1}^k x^{(0)}(i), k = 1, 2, \dots, n$. ([4]). Obviously, $x^{(0)}(1) = x^{(1)}(1)$.

Form 1

$$x^{(0)}(k) + ax^{(1)}(k) = b \quad (2.1)$$

where a is the development coefficient of the model, b is the grey input of the model. However, some scholars found using $X^{(1)}$ directly to model can grow too fast to get more error. Therefore, it introduced another sequence $Z^{(1)}$ to replace $X^{(1)}$ to avoid grow too fast as follow:

Definition 2.2. [28] Set $z^{(1)}(k) = 0.5[x^{(1)}(k) + x^{(1)}(k - 1)]$ is the mean generating sequence of 1-AGO sequence, $k = 2, 3, \dots, n$.

Form 2

$$x^{(0)}(k) + az^{(1)}(k) = b \quad (2.2)$$

Liu et al [28] showed the example result to proof the Form 2 is better than Form 1. In order to use and caculate conveniently, it gave its whitenization formula of the GM(1,1)model is

Form 3

$$\frac{dx^{(1)}}{dt} + ax^{(1)} = b \quad (2.3)$$

The corresponding solution equation is $\hat{x}^{(1)}(t) = (x^{(0)}(1) - b/a)e^{-at} + b/a$. If $t \leq n$, it called GM(1,1) fitting model, if $t > n$, it called GM(1,1) predictting model. After that, it get the fitting and forecasting of $X^{(1)}$. Then, it need give the inverse of $X^{(1)}$ to get the fitting and forecasting of $\hat{X}^{(0)}$ which $\hat{x}^{(0)}(k) = \hat{x}^{(1)}(k) - \hat{x}^{(1)}(k-1)$, where $k = 2, 3, \dots, n$ and $\hat{x}^{(0)}(1) = \hat{x}^{(1)}(1)$.

Based on this classical grey model, scholars proposed different grey model to solve different problems. Xie et al [29–33] and Wang et al [34] used discrete grey model DGM(1,1) applied in exponential law sequence and focused on its optimization as follow.

Form 4

$$x^{(1)}(k+1) = \beta_1 x^{(1)}(k) + \beta_2 \quad k = 1, 2, \dots \quad (2.4)$$

where $\beta_1 = \frac{1-0.5a}{1+0.5a}$, $\beta_2 = \frac{b}{1+0.5a}$. If $k \leq n$, it called DGM(1,1) fitting model, if $k > n$, it called DGM(1,1) predictting model.

Deng et al [27] had given the equivalence proof between form 3 and form 4 by the McLaughlin formula. Here, it will not be repeated again.

As Definition 2.1 mentioned, one order AGO is using $X^{(0)}$ as the original sequence to get. Therefore, two order AGO is using $X^{(1)}$ as the original sequence to get so that GM(n,1) means n order AGO sequence Grey Model. Kayacan et al [35] used GM(n,m) which is n order equation and m variables, and Tseng et al [36] applied it in time series prediction, Hsu et al [37] and Wu et al [38] developed grey models with multiple variables to influence the final output within different factors. In addition, Lin et al [39] based on one order generating accumulated operation proposed second-order grey differential

equation to identify DNA binding proteins. Wang et al [40] transformed the one order generating accumulated operation to two orders generating accumulated operation to solve oil-dissolved problems.

Obviously, in definition 2.1, it showed that in each $x^{(1)}(k)$ calculation process, $x^{(0)}(1)$ which is the oldest data play a greater weight than $x^{(0)}(n)$ which is the latest data. However, in the case of predicting with less records data, the latest data should play a greater weight than the old data which called the principle of new information priority [17]. Che et al [41] used generating accumulated operation in opposite-direction which meets new information priority. A huge number of scholars have applied in practical examples and showed its accuracy [23, 42–44]. It can see the AGO in opposite-direction definition as follow:

Definition 2.3. Let sequence $X_{(1)} = (x_{(1)}(1), x_{(1)}(2), \dots, x_{(1)}(n))$ be the 1-AGO accumulated generating sequence in opposite direction of $X^{(0)}$, where $x_{(1)}(k) = \sum_{i=k}^n x^{(0)}(i)$, $k = 1, 2, \dots, n$. ([41]). Obviously, $x^{(0)}(n) = x_{(1)}(n)$.

In addition, Guo [45] et al added the self-memory principle to GM(1,n) model on engineering fields. Tseng et al [46] used Hybrid Grey model to predict seasonal time series. Here, it will not give detailed explanation.

Considered the existing the error by integer grey model is not perfect as well, scholars have been researched on fractional order grey model. Fractional grey model is based on classical grey model, which means changing integer-order accumulated generating operation to fractional order accumulated generating operation. As the first researchers to propose the fractional grey model, Wu et al [47–51] has been achieved the less error with the specific fractional generating accumulated order which called FAGM(1,1) in a great deal of practical problems although the fractional order is not the optimal order to get the minimum error. It can see the FAGM(1,1) grey model as follow:

As definition 2.1 mentioned, It can include that r-order AGO sequence $X^{(r)}$ of $X^{(r-1)}$ is $x^{(r)}(k) = \sum_{i=1}^k x^{(r-1)}(i)$. Therefore,

$$\begin{aligned} (x^{(r)}(1), x^{(r)}(2), \dots, x^{(r)}(n)) &= \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \dots & 1 \end{pmatrix}_{n \times n} \cdot (x^{(r-1)}(1), x^{(r-1)}(2), \dots, x^{(r-1)}(n))^{\top} \\ &= \begin{pmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \\ 1 & 1 & \dots & 1 \end{pmatrix}_{n \times n}^r \cdot (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n))^{\top} \end{aligned}$$

$$= \begin{pmatrix} 1 & 0 & 0 & \dots & 0 \\ r & 1 & 0 & \dots & 0 \\ \frac{r(r+1)}{2!} & r & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \frac{r(r+1)\dots(r+n-2)}{(n-1)!} & \frac{r(r+1)\dots r(r+1)\dots(r+n-3)}{(n-2)!} & \frac{r(r+1)\dots r(r+1)\dots(r+n-4)}{(n-3)!} & \dots & 1 \end{pmatrix}_{n \times n} \cdot (x^{(0)}(1), x^{(0)}(2), \dots, x^{(0)}(n))^T$$

Here, in order to narrate conveniently, it set coefficient matrix is A matrix. Therefore, r -order AGO matrix is $A^r = (a_{ij}^r)_{n \times n}$, where $j \leq i$, $a_{ij}^r = \frac{(i-j+r-1)!}{(r-1)!(i-j)!}$, otherwise it equals 0. It can be proved by mathematical induction [3, 9, 26, 47, 50, 52]. Noticed in the original grey theory, r is a integer number. Now, it has been expanded to a fractional number like 0.2, 1.2, 2.3, ... based on fractional calculus theory [53, 54].

It not only transformed the accumulated generating order [47–50] but also expanding the accumulated generating order in opposite-direction which meet the new information priority. Obviously, as Definition 2.3 mentioned, it can conclude that 1-order AGO matrix A is the transpose of 1-order AGO matrix in opposite-direction [55].

Moreover, Wu et al [51, 56] also first combined the Caputo fractional order derivative into classical grey model GM(1,1) which can decrease the accumulated generating error by using original data modeling directly as follow.

$$x^{(r)}(k) - x^{(r)}(k-1) + az^{(r)} = b \quad (2.5)$$

It can see in paper [47–50] that scholars only tried different orders to compare the error value to identify the most suitable fractional order.

2.1.1 Mean absolute percentage error

As the researchers in depth study [46, 57–61], they prefer to choose the Mean Absolute Percentage Error (MAPE) function as the measure of the error as follow:

$$\text{MAPE}(r) = \frac{1}{n} \sum_{k=1}^n \frac{|\hat{x}^{(0)}(k) - x^{(0)}(k)|}{x^{(0)}(k)} \times 100\% \quad (2.6)$$

where $x^{(0)}(k)$ is the actual value, $\hat{x}^{(0)}(k)$ is fitting values or forecasting values. There are some other measure function as follow:

Least absolute deviations(LAD)

$$S = \sum_{i=1}^n |x^{(0)}(k) - \hat{x}^{(0)}(k)| \quad (2.7)$$

Mean absolute error(MAE)

$$\text{MAE} = \frac{\sum_{i=1}^n |\hat{x}^{(0)}(k) - x^{(0)}(k)|}{n} \quad (2.8)$$

Mean percentage error(MPE)

$$\text{MPE} = \frac{1}{n} \sum_{k=1}^n \frac{\hat{x}^{(0)}(k) - x^{(0)}(k)}{x^{(0)}(k)} \times 100\% \quad (2.9)$$

Obviously, it can see if the records of data are in a huge amount, the LAD value is huge as well so that the convergence speed will be slow. Compare the MAE, MAPE value seems more intuitive. Meanwhile, it is hard to judge the positive or negative of $\hat{x}^{(0)}(k) - x^{(0)}(k)$ in MPE so that when use this to measure is not reasonable.

Tofallis et al has showed "MAPE is the most widely used measure of prediction accuracy" in different fields [62]. When $MAPE < 10\%$, it can conclude this model has higher accuracy. In MAPE function, there is the operation of the absolute value so it cannot obtain the minimum by calculating the derivative directly. Therefore, based on the optimization algorithm and choosing the error function MAPE as the objective function, it searches the most optimal fractional order to get the minimum error. It included Yang et al [5] used Quantum Genetic Algorithm (QGA), Zhao et al [6] used differential evolution algorithm, Chen et al [7] used fruit fly optimization algorithm and Hsu et al [8] used genetic algorithm. Most scholars [3, 9–16] used Particle Swarm Algorithm to optimize parameters included orders in grey model to improve the accuracy.

As yet no one has used Gravitational Search Algorithm (GSA) to optimize parameters in grey model.

Through the study of scholars [3, 10–12, 14], it found optimal parameters grey models by the optimization algorithm have less error in forecasting problems than original grey models. Moreover, different systems or problems need different parameters to get minimum error. Therefore the use of efficient optimization algorithms to select parameters is worth investigating and so we first review PSO before considering the use of GSA which to we believe has not been used in this context before.

2.2 Particle Swarm Optimization Algorithm

The basic principle as follow: In PSO, every agent can be called one particle, which represents a implied solution. For example, in a D dimension as the objective search space. Set there are m particles. Here, if m is over large, it will influence the search speed and convergence.

set $z_i = (z_{i1}, z_{i2}, \dots, z_{iD})$ is the vector location of the i th agent in D dimension, where $i = 1, 2, \dots, m$. It need set fitness function at first, which is related to objective solution problems. Then algorithm will caculate the fitness function. This value can measure the position (solution) is good or bad. $v_i = (v_{i1}, v_{i2}, \dots, v_{i1}, \dots, v_{id})$ is the particle flying speed, $p_i = (p_{i1}, p_{i2}, \dots, p_{id}, \dots, p_{iD})$ is the best position (solution) untill now by the particle searching, $p_g = (p_{g1}, p_{g2}, \dots, p_{gl}, \dots, p_{gD})$ it the global position(solution) for whole particles. In each iteration, the new speed and location updated as follow:

$$v_{id}^{k+1} = v_{id}^k + c_1 r_1 (p_{id} - z_{id}^k) + c_2 r_2 p_{gd} - z_{id}^k \quad (2.10)$$

$$z_{id}^{k+1} = z_{id}^k + v_{id}^k + 1 \quad (2.11)$$

where $i = 1, 2, \dots, m, d = 1, 2, \dots, D, k$ is the iteration numbers, r_1 and r_2 are the rand number in $[0,1]$. c_1 and c_2 are the memory factor, which can memory past self positions and best position to make itself closer to the best values.

The simplify process as follow:

step1: Caculate the fitness value;

step2: Update the v_{id} and z_{id} ;

step2: Whether the number of iterations is maximum and the objective function condition is satisfied or not; if yes, exit; else go back step 1.

PSO is based on birds organizational behaviour to solve the optimizatoin of nonlinear functions, which proposed by Eberhart and Kennedy in 1995 [63], where due to basic parameters is fixed, the optimal accuracy is bad for some specific function. Based on the original PSO, Shi et al [64] expanded the inertia factor as the linear decreasing function which can balance the relation between the exploration and exploration and the exploitation as follow:

$$v_{id}^{k+1} = \omega v_{id}^k + c_1 r_1 (p_{id} - z_{id}^k) + c_2 r_2 p_{gd} - z_{id}^k \quad (2.12)$$

where $\omega = \omega_{max} - \frac{\omega_{max} - \omega_{min}}{iter_{max}} \times k$.

It improved the PSO performance significantly so that scholars called it "Standard PSO". In addition, a great deal of researchers [65–69] developed PSO to solve the different complex systems. Since it incepted, Hami et al used improved PSO to apply in "case of casablanca stock exchange" [70], Liu et al [71] used it to solve dynamic problems such as adjustment of power system, Singh et al [72] focused the integral and derivate parameters problems to optimize the eletrical loading frequency control, etc.

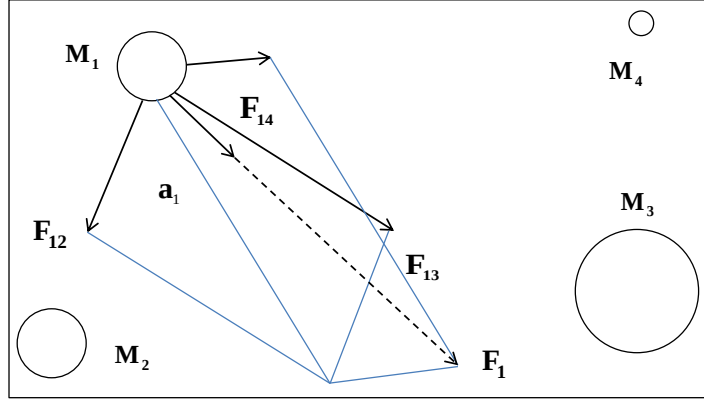


FIGURE 2.1: Every mass accelerate toward the result force that act it from the other masses. After [1]

2.3 Gravitational Search Algorithm

In 2009, Rashedi [1] et al proposed the new deterministic population-based heuristic optimization algorithm based on the law of Newton gravity which can be explored and called it Gravitational Search Algorithm (GSA). In GSA, every object can be regarded as a mass. In the search space, every mass with acceleration attracted each other by gravity as in Figure 2.1. The object's mass which can represent their performance is the only factor to decide the speed and acceleration. Their mass is computed by fitness function which is set by the optimum problems in advance. Iteration process will be stopped by the lighter mass moved the best position. It means the fitness become the global fitness. The location of the mass represents the solution of the system within the fitness function condition. Here, the principle flow as follow based on minimize optimization as an example in Figure 2.2 [2]: where $fit_i(t)$ is the fitness function of i th at time t ; $M_i(t)$ is the mass of i th at time t ; $G(t)$ is the gravity at time t ; R_{ij} is the distance between i th agent and j th agent; $F_i^d(t)$, a_i^d , V_i^d and x_i^d represent force, acceleration, velocity and position of the i th in the d th dimension.

Compared to the other optimization methods such as Harmony Search[73], Particle Swarm Optimization[74], the Bayesian Optimization Algorithm[75], Artificial Bee Colony Optimization Algorithm[76], Ant Colony Optimization Algorithm[77], Genetic Algorithm[78], etc, GSA is the new one, it can be explored more and the optimal values is better for some particular benchmark functions as seen in [1]. Here 23 classical benchmark functions are testing to compare the Real Genetic Algorithm (RGA), PSO and GSA, where all seven unimodal testing functions and three multimodal benchmark functions showed the GSA is the best one to find the global optimal. Furthermore, in that paper, it compared Central Force Optimization and GSA as well with ten benchmark functions, there are seven functions got best values using GSA. In 2010, Rashedi proposed the binary GSA (BGSA) which use some binary criteria to evaluate the performance [79] and it

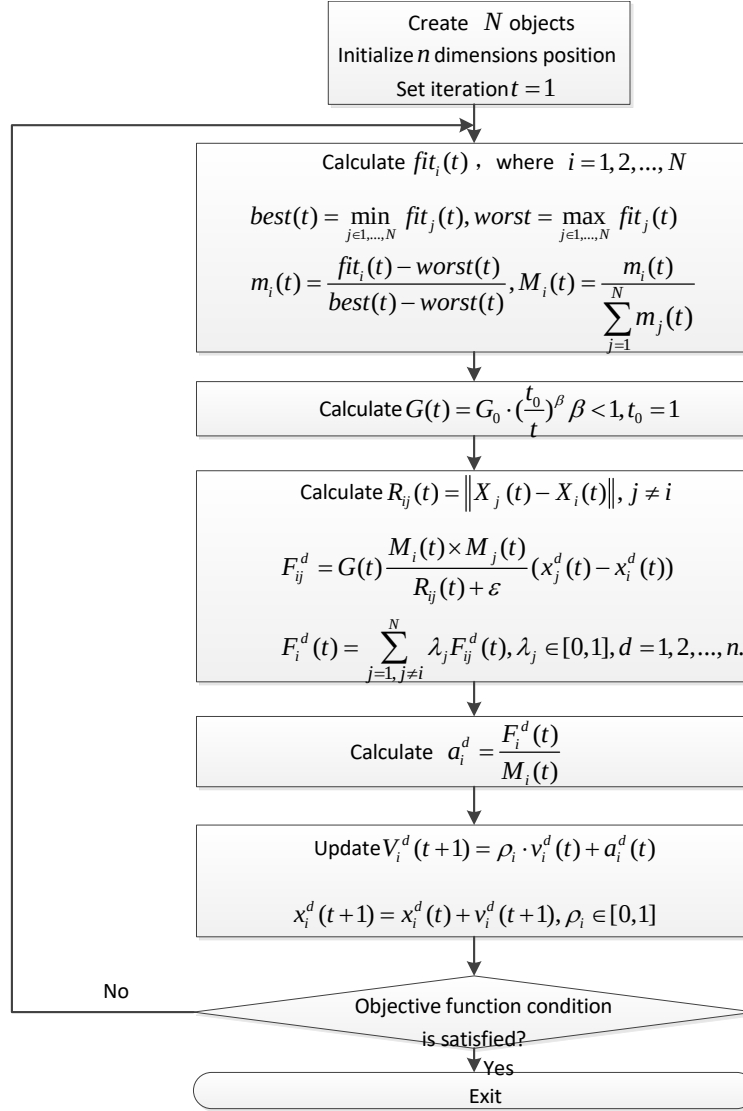


FIGURE 2.2: GSA minimization principle flow After [2]

was applied in feature selection [80]. The difference between GSA and BGSA is the update of x_i^d (It represents position of i th agent in d th dimension) which only equals 0 or 1. Moreover, Precup et al [81] using the coding to assign the proportion of exploration and exploitation as 15% and 45% so that it can improve the accuracy and accelerate the convergence speed focused on the fuzzy and PI Controlled Servo system [82]. Furthermore, Mirjalili et al [83] proposed the hybrid PSOGSA to combine the strength of PSO and GSA. Under this idea, it solved a great deal of problems such as antenna array system, fuzzy-VQ image compression, environmental dispatch, power flow system, T-S fuzzy neural network prediction, etc [84–88]. In addition, scholars proposed and used chaotic GSA, discrete GSA, GSA with wavelet mutation, Fuzzy GSA, for particular problems such as system identification in T-S fuzzy model [89], combinatorial optimization problems [90], economic load dispatch [91], multimachine power system stabilizers [92], etc.

Since inception, scholars prefer to compare PSO and GSA at the same problem such as Wei et al [93] expanded improved GSA to optimize the result based on the PID control by the Zhang et al [94] using PSO, Hassanzadeh et al [95] based the multi- objective PSO [96] proposed multi-objective GSA to solve MOPSO convergence premutually to local optimal. In addition, it can be found the applications with GSA and its variants in multi-hop wireless network[97],the interconnected power system [98], maximum optimization of arrival direction estimation [99], linear time-delayed system [100], human finger vein pattern identification [101],etc.

Chapter 3

Research Methods

3.1 Philosophy

This is the scientific experiment paper. It can expand the scope of GSA application. Moreover, the grey system theory can be more perfect. In the optimization theory of fitting and forecasting data, it provided a guide of solving the complexed problems with limiting and messy data.

3.2 Experiment design

3.2.1 Experiment hypothesis

When applied in the same grey model, GSA can get higher accuracy than PSO.

3.2.2 Experiment apparatus

Here, it will use MATLAB software to achieve this process.

MATLAB (Matrix Laboratory) is the commercial programming software invented by MathWorks Corporation in America. It is an interactive environment for advanced technical computing languages such as algorithm exploration, data visualization and analysis. Besides, it also can be used in numerical simulation. It mainly includes MATLAB software and Simulink software.

MATLAB language includes condition functions, data structures, statements, input and output, and object-oriented programming characters. You can enter the "input statement" and the "execution command" in the command window at the same time, or

you can write a large complex programming file which called ".file" at first and then run it together. It is more convenient to the user with limited professional programming skills. Moreover, this language can be transplanted and highly expandable, this is why MATLAB can be obtained this depth in scientific research and engineering calculation fields.

More and more scholars use MATLAB to analyze and compute the model to solve the practical problem in life in the world. For example, data analysis which includes loading the data and plotting the data, curve fitting, mapping surface curvature to colour [102]. It not only used widely in mathematical discipline, but also applied successfully in engineering fields such as handling strong movement records and estimating earthquake parameters[103]. It also applied in fractional grey model prediction of electricity prediction research [104]. In addition, there are a great deal of ready-made toolboxes, it is convenient for beginner to use it.

3.3 Any other software methodology

SciPy

SciPy is the open-source software in Python environment which functions are the similar with MATLAB. Python is the new programming based on the classical "ABC" language. Moreover, it combined the C language and other coding language so that it solve the nonopen-source problems. Furthermore, as Python incepted, it has been used in Google, Youtube, Yahoo,etc. However, it has been used in Linux operation. In other words, it is more suitable for business and companys instead of students, scholars, individuals, etc.

Octave

Octave is a tool which designed to solve scientific computing and numerical analysis problems. Moreover, it is free and open source as well. Most codings can be running directly on Matlab. In addition, it takes up less memory and the functions are also very similar with Matlab. Therefore, it is a strong competitor to MATLAB. However, The advantage of the MATLAB software is its excellent service due to its expensive fees. It is easy to find a great deal of useful coding source on its official website.

In conclusion, taking into account the advantages of MATLAB software modeling and I learned MATLAB software when I was a college student, this paper selects the MATLAB software for grey model fitting and predicting. Moreover, My computer operation system is windows 7. In addition, the original coding of GSA from official website is MATLAB coding.

Chapter 4

Development Methods

4.1 The coding achievement of GSA itself

Considering the significance of GSA in this paper, it will check the GSA coding without grey model at first. Here, it chose the GSA coding from the MATLAB official website which is published by the author who proposed the GSA [1]. In [1], Rashedi et al were inspired by the law of motion and gravity as mentioned in Figure 2.1 to propose this new algorithm concept. It also transformed the concept into the calculation formulas as in Figure 2.2, where the $fit_i(t)$ (fitness function) was compiled named "*test_functions.m*"; $m_i(t)$ (mass function) was compiled named "*massCalculation.m*"; $G(t)$ (Gravity constant) was compiled named "*Gconstant.m*"; $F_i^d(t)$ (Force function) and a_i^d (acceleration) were compiled named "*Gfield.m*"; $V_i^d(t+1)$ (velocity) and $x_i^d(t+1)$ (position) were compiled named "*move.m*" in advance. In addition, the author also compiled the "*test_functions_range.m*" to input the fitness functions dimension and its range. The purpose of "*space_bound.m*" is to control the moving of agents in the search space. In order to use acceleration, Gravity constant, position, velocity more convenient, the author compiled "*GSA.m*" to put them together. At last, it compiled "*main.m*" to call the function has been mentioned above to output the "Fbest" which is the fitness optimum. Considering the use of the following, a few codings such as "*massCalculation.m*", "*Gconstant.m*", "*test_functions.m*", "*test_functions_range.m*" and "*main.m*" were listed in the appendix A.

In order to test GSA, Rashedi et al[1] chose 23 benchmark function as testing functions. Moreover, it compared the results of GSA, PSO and Real genetic Algorithm (RGA) based the same benchmark function. There are nearly two-thirds benchmark functions showed GSA is better than PSO when search the optimum. About four-fifths benchmark functiond showed GSA is better than RGA. In addition, considering the number of

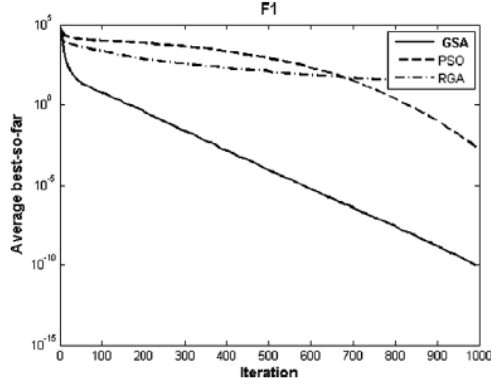


FIGURE 4.1: The original figure in paper [1]

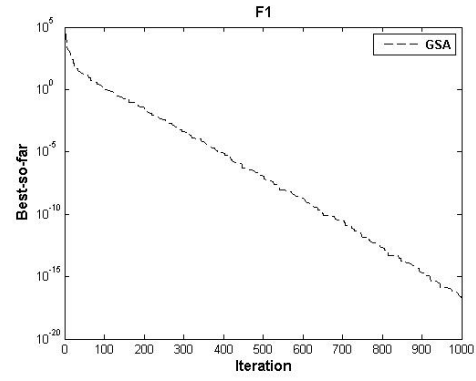


FIGURE 4.2: Sphere model optimization with GSA

benchmark function is large, it will choose some special benchmark functions to explain the performance of GSA. The results as follow:

1. Sphere Model

Function expression:

$$f_1(x) = \sum_{i=1}^{30} x_i^2 \quad (4.1)$$

where the searching range is $-100 \leq x_i \leq 100$. This is the nonlinear symmetric unimodal function. Obviously, the global minimum of this function is $f_1(0, \dots, 0) = 0$. Compared other test function, this is relatively simple, most algorithms can easily to get optimum so that it main used in testing the optimization accuracy. It can see the running coding in the Figure 5.5 in Appendix.A. The optimization results as follow in Figure 4.1 and Figure 4.2: where dimension is $n = 30$; population size is $N = 50$; iteration number is 1000.

It can see that the GSA optimization trends is the same. Obviously, the minimum in Figure 4.2 is smaller than the result in the Figure 4.1. It means the result has been improved now. In Figure 4.1, the convergence speed is much faster than PSO and RGA and the minimum is smaller as well.

2. Quartic function i.e. noise

Function expression:

$$f_7(x) = \sum_{i=1}^{30} ix_i^4 + \text{random}[0, 1) \quad (4.2)$$

where the searching range is $-1.28 \leq x_i \leq 1.28$. This is the function with noise where $\text{random}[0, 1)$ is the random variable between $[0, 1)$. Obviously, the global minimum of this function is $f_7(0, \dots, 0) = 0$. Due to this function include the random noise variable,

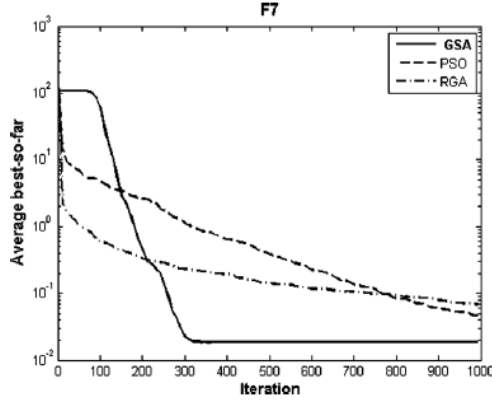


FIGURE 4.3: The original figure in paper [1]

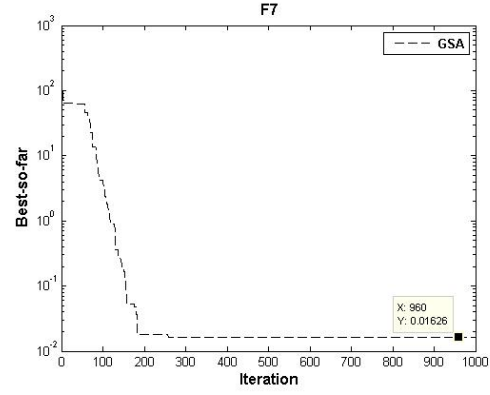


FIGURE 4.4: Quartic model optimization with GSA

it usually used to measure the performance of unimodal function with lots of noise. It only need modify "F_index=1" to the "F_index=7" in "main.m" file as Figure 5.5 in Appendix A. The optimization results as follow in Figure 4.3 and Figure 4.4: where dimension is $n = 30$; population size is $N = 50$; iteration number is 1000.

It can see that the GSA optimization trends are the same. Moreover, the minimum in original figure of paper [1] is 0.018. Now, it can see the minimum in the Figure 4.4 is 0.01626. The result has been improved as well. In Figure 4.3, the convergence speed is a little faster than PSO and RGA and the minimum is smaller as well.

3. Ackley Function

Function expression:

$$f_{10}(x) = -20 \exp \left\{ -0.2 \sqrt{\frac{1}{30} \sum_{i=1}^3 0x_i^2} \right\} - \exp \left(\frac{1}{30} \sum_{i=1}^{30} \cos 2\pi x_i \right) + 20 + e \quad (4.3)$$

where the searching range is $-32 \leq x_i \leq 32$. This is the continuous and rotating multimodal function. It adjusted the exponential function by the cosine function. This multimodal function has a great deal of local optimal. Obviously, the global minimum of this function is $f_{10}(0, \dots, 0) = 0$. In addition, the optimal is located at the edge. Therefore, if the initial value of algorithms falls on the edge, it will be easy to find the optimal rapidly. It only need modify "F_index=1" to the "F_index=10" in "main.m" file as Figure 5.5 in Appendix A. The optimization results as follow in Figure 4.5 and Figure 4.6: where dimension is $n = 30$; population size is $N = 50$; iteration number is 1000.

It can see that the GSA optimization trends is the same. Obviously, the minimum in Figure 4.6 is smaller than the minimum in original paper. The result has been improved

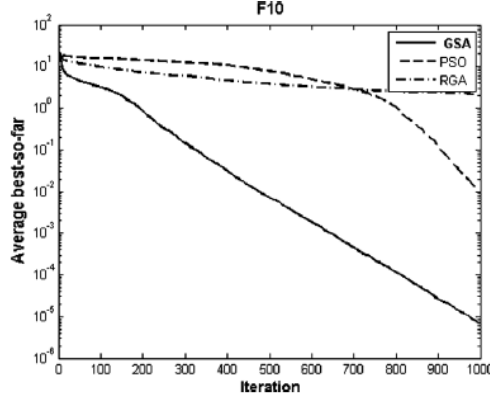


FIGURE 4.5: The original figure in paper [1]

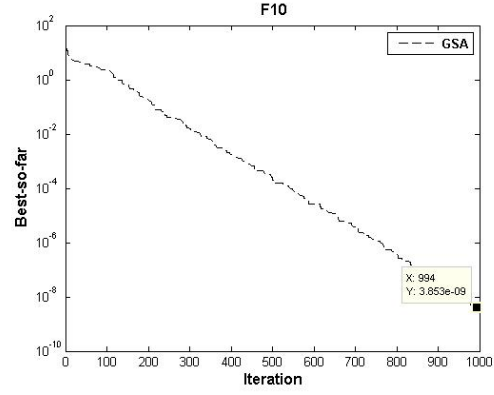


FIGURE 4.6: Ackley function optimization with GSA

as well. In Figure 4.5, the convergence speed is much faster than PSO and RGA and the minimum is smaller as well.

In conclusion, the GSA coding from the MATLAB official website can be showed the better result than the original paper. There may be some reasons which can explain that. Firstly, this code is generated by the author in 2010. However, the paper was finished in 2009. Maybe some coding steps has been simplified. Secondly, the MATLAB edition has been improved. Here, it use MATLAB R2014a instead of R2009b in original paper. Thirdly, the computer system may influence the result. It cannot know the system when the author was writing. Here, it use Windows 7 64-bit system. Moreover, in paper [1], there are almost two thirds benchmark functions which can be showed the GSA optimal is better than PSO optimal. It is hard to say which algorithm is better than the other algorithm due to the diversity of the problems. Therefore, in this paper, it is the first time to use GSA to optimal grey model based the existing paper which used PSO.

4.2 Experiment

4.2.1 The first grey model is chosen in this paper

Classical grey model GM(1,1) has been introduced in the section 2.1. Based on this one, Wen et al [105] expanded the mean generating sequence $z^{(1)}(k)$ of 1-AGO sequence in the definition 2.2 as follow:

Definition 4.1. Set $z^{(1)}(k) = \lambda x^{(1)}(k) + (1 - \lambda)x^{(1)}(k - 1)$ is the new generating sequence of 1-AGO sequence, $k = 2, 3, \dots, n$, where λ is between 0 and 1.

In the classical GM(1,1), the λ equals 0.5. In [105], the author showed different λ can influence the final forecasting error. However, it cannot find the most suitable λ to get the smallest error value. Therefore, Ma et al [14] used PSO to search the smallest error by setting the λ as the parameter as follow where $k = 1, 2, \dots$:

Step 1: Input actual value $x^{(0)}(k)$;

Step 2: Calculate the forecasting value $\hat{x}^{(0)}(k)$ with parameter λ due to the GM(1,1) as definition 4.1;

Step 3: Calculate the MAPE function with parameters λ as equation 2.6;

Step 4: Input MAPE function from step 3 in PSO algorithm to output the λ ;

Step 5: Update the GM(1,1) with λ from step 4 to output the fitting values $\hat{x}^{(0)}(k)$;

Step 6: Repeat Step 3 to step 5 until get the minimum MAPE.

In [14], there are two testing data which one is China's ore import from 1995 to 2010 and another is its consumption where the data source is from China statistical yearbook. Ma et al used four different grey models which are GM(1,1), PSO-GM(1,1, λ), Rolling GM(1,1) and PSO-Rolling GM(1,1, λ) to fit data from 1995 to 2010 and forecast data from 2011 to 2015. However, the date of this paper published is 2013. The forecasting data from 2011 to 2015 cannot calculate the error without the actual value. Here, it will choose China's ore import from 1995 to 2010 as testing data to fit and download the data between 2011 to 2015 from the China statistic yearbook as the actual value to forecast. Moreover, it will use classical regression methods such as Excel to fit and forecasting the data. Furthermore, we will give GM(1,1), GSA-GM(1,1, λ), Rolling GM(1,1) and GSA-Rolling GM(1,1, λ) results to compare and analyze it.

4.2.1.1 Testing data

In Table 4.1, there are 21 records from 1995 to 2015, where the unit of the import data is 10000 tons.

4.2.1.2 Fitting and forecasting with EXCEL

The main steps as follow:

Step 1: Input the testing data from 1995 to 2010 in the EXCEL sheet;

Step 2: Use EXCEL existing tools to draw scatter charts as follow in Figure 4.7;

Step 3: According to the graphical trend, the exponential function and the polynomial function are used to fit as follow in Figure 4.8 and Figure 4.9:

Step 4: Calculate the fit data and the forecast data based on the formula in the Figure 4.8 and Figure 4.9. The result as follow in Table 4.2 and Table 4.3, where the first MAPE

Year	Number	Actual
1995	1	4115
1996	2	4387
1997	3	5510
1998	4	5177
1999	5	5527
2000	6	6997
2001	7	9231
2002	8	11149
2003	9	14812
2004	10	20808
2005	11	27523
2006	12	32629
2007	13	38309
2008	14	44356
2009	15	62778
2010	16	61863
2011	17	68608
2012	18	74360
2013	19	81910
2014	20	93234.18
2015	21	95272.29

TABLE 4.1: China's iron ore import

is fitting data error with first 16 values and another is predicting data error with last 5 values;

It can see the polynomial function can fit and forecast the original data better. Moreover, the fitting MAPE is about $9.924\% < 10\%$. It can conclude that the fitting results have higher accuracy.

4.2.1.3 Fitting and forecasting used classical GM(1,1)

The main steps as follow:

Step 1:Input raw sequence $X^{(0)}=(4115, 4387, 5510, 5177, 5527, 6997, 9231, 11149, 14812, 20808, 27523, 32629, 38309, 44356, 62778, 61863)$ into MATLAB software named "GM11.m".

Step 2:Use 1-AGO which mentioned in definition 2.1 to get $X^{(1)}=(4115, 8502, 14012, 19189, 24716, 31713, 41034, 52183, 66995, 87803, 115326, 147955, 186264, 230620, 293398, 355261)$, where the $X^{(1)} = A * X^{(0)}$ and A is the coefficient matrix as defined in Chapter 2.

Exponential Function			
Actual	Forecast	$\text{abs}(\text{Forecast} - \text{Actual}) / \text{Actual}(\%)$	MAPE(%)
4115	3153.7222	23.3603	13.15353635
4387	3864.311025	11.9145	
5510	4735.007952	14.0652	
5177	5801.888141	12.0705	
5527	7109.155114	28.6259	
6997	8710.972223	24.4958	
9231	10673.70677	15.6289	
11149	13078.67977	17.3081	
14812	16025.5353	8.1929	
20808	19636.36896	5.6307	
27523	24060.78667	12.5793	
32629	29482.10314	9.6445	
38309	36124.93713	5.7012	
44356	44264.51791	0.2062	
62778	54238.08875	13.6034	
61863	66458.8797	7.4291	
68608	81433.22881	18.6935	
74360	99781.56094	34.1871	
81910	122264.0935	49.2664	
93234.18	149812.3343	60.6839	
95272.29	183567.676	92.6769	51.10156298

TABLE 4.2: China's iron ore import with exponential function

Polynomial Function			
Actual	Forecast	$\text{abs}(\text{Forecast} - \text{Actual}) / \text{Actual}(\%)$	MAPE(%)
4115	5783.43	40.5451	9.92410181
4387	4462.42	1.7192	
5510	3893.47	29.3381	
5177	4076.58	21.2559	
5527	5011.75	9.3224	
6997	6698.98	4.2593	
9231	9138.27	1.0045	
11149	12329.62	10.5895	
14812	16273.03	9.8638	
20808	20968.5	0.7713	
27523	26416.03	4.0220	
32629	32615.62	0.0410	
38309	39567.27	3.2845	
44356	47270.98	6.5718	
62778	55726.75	11.2320	
61863	64934.58	4.9651	
68608	74894.47	9.1629	
74360	85606.42	15.1243	
81910	97070.43	18.5086	
93234.18	109286.5	17.2172	
95272.29	122254.63	28.3213	17.66686125

TABLE 4.3: China's iron ore import with polynomial function

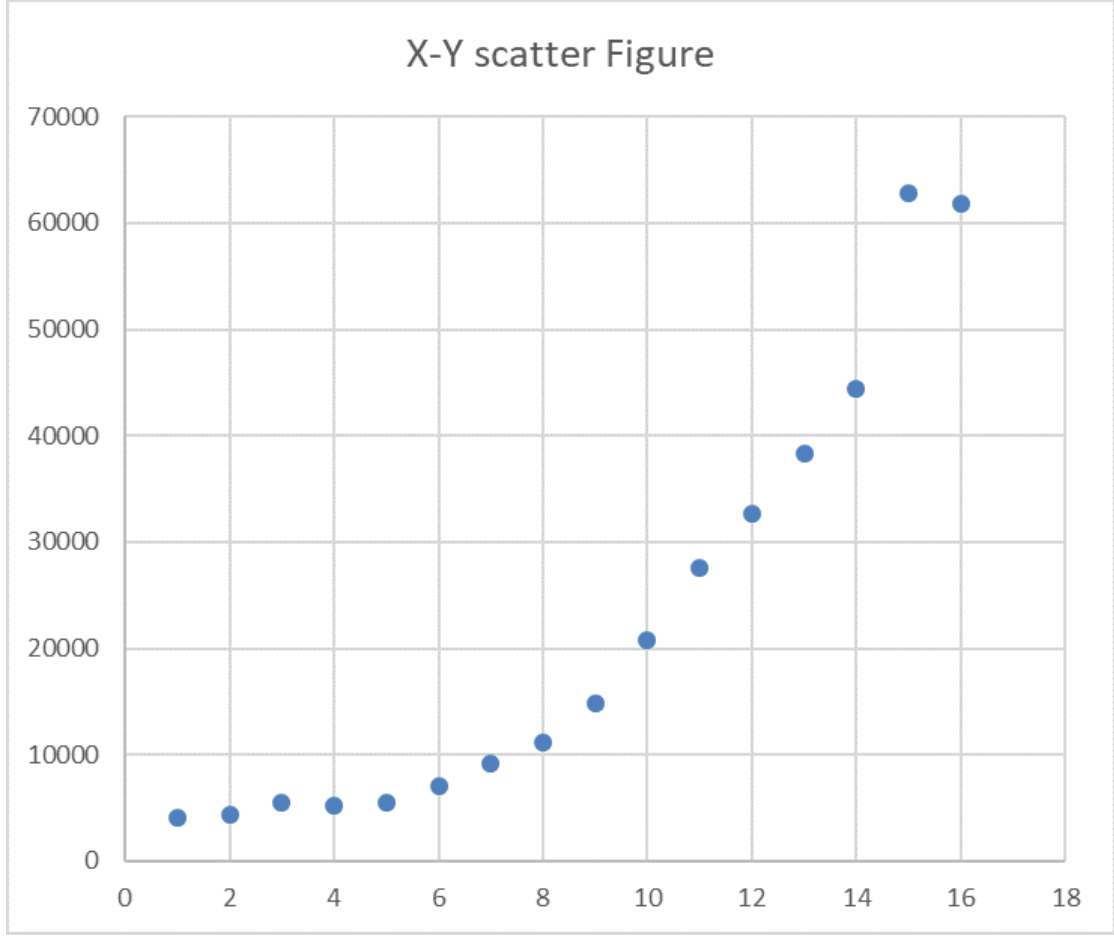


FIGURE 4.7: X-Y scatter chart of original data

Step 3:As mentioned in definition 2.2, the mean generating sequence is $z^{(1)}(k) = 0.5[x^{(1)}(k) + x^{(1)}(k-1)]$, where $k = 2, 3, \dots, n$. Therefore, $Z^{(1)} = (6308.5, 11257, 16600.5, 21952.5, 28214.5, 36373.5, 46608.5, 59589, 77399, 101564.5, 131640.5, 167109.5, 208442, 262009, 324329.5)$.

Step 4:Now, substitute $X^{(0)}$ and $Z^{(1)}$ into equation 2.2. It will get 15 equations with different k . Considering the equation 2.2 is the linear equation, it can use the least squares method to solve the development coefficient a and grey input b , which means to find a and b to make the δ as follow is the minimum.

the principle as follow where $k = 2, 3, \dots, 15$:

- Set $\delta = \sum_{k=2}^{15} [x^{(0)}(k) + az^{(1)}(k) - b]^2$;
- Differentiate δ with respect to a and b respectively and set this two formulas equals 0 (extreme value principle) as follow:

$$\begin{cases} 2 \sum_{k=2}^{15} (x^{(0)}(k) + az^{(1)}(k) - b)z^{(1)}(k) &= 0 \\ 2 \sum_{k=2}^{15} (x^{(0)}(k) + az^{(1)}(k) - b)(-1) &= 0 \end{cases} \quad (4.4)$$

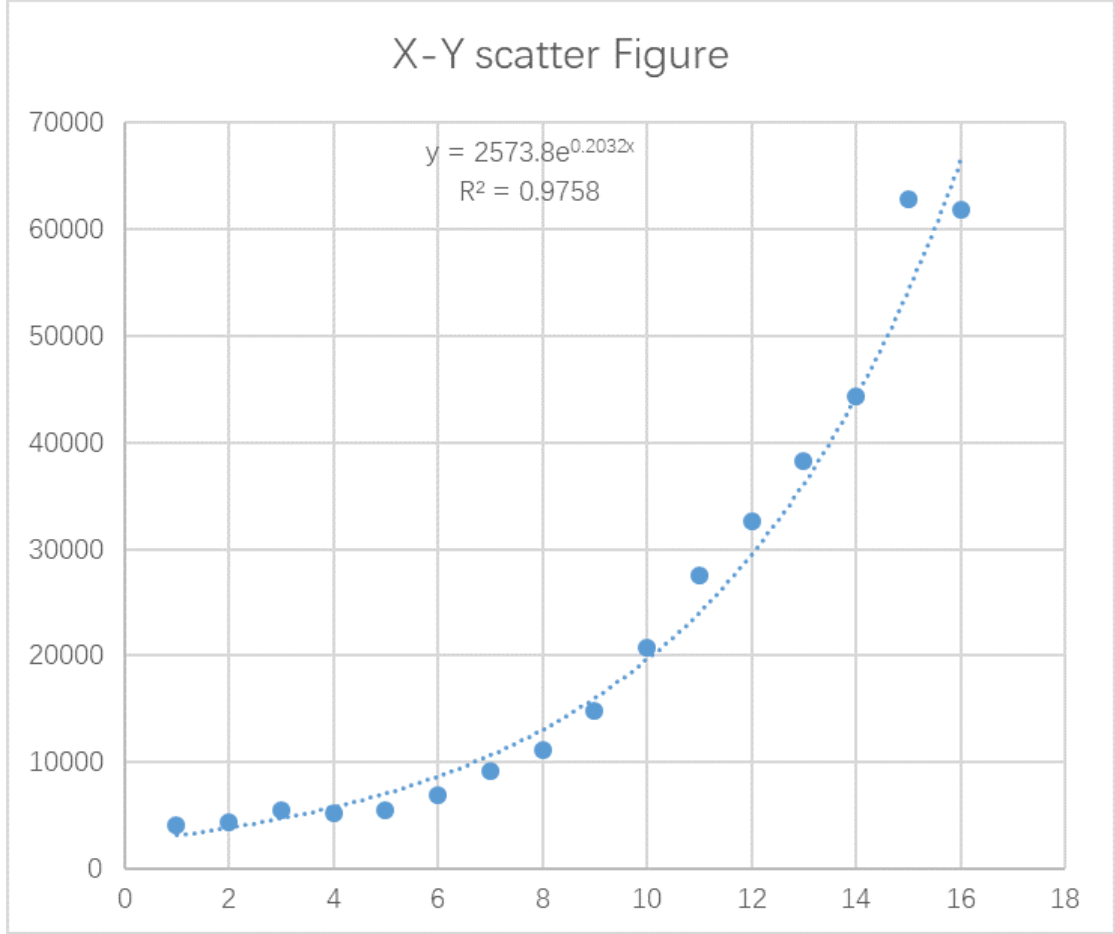


FIGURE 4.8: X-Y scatter chart of original data with exponential function

- Set the $B = \begin{pmatrix} -z^{(1)}(2) & 1 \\ -z^{(1)}(3) & 1 \\ \vdots & \vdots \\ -z^{(1)}(15) & 1 \end{pmatrix}$, $Y = \begin{pmatrix} -x^{(0)}(2) \\ -x^{(0)}(3) \\ \vdots \\ -x^{(0)}(15) \end{pmatrix}$, $P = \begin{pmatrix} a \\ b \end{pmatrix}$;

- Equation 4.4 can transform to matrix equation $B^T(Y - BP) = 0$;

- The solution of matrix equation above is $P = (B^T B)^{-1} B^T Y$;

After caculated above, it can get $P = \begin{pmatrix} -0.2033 \\ 3091.5752 \end{pmatrix}$; where $a = -0.2033$ and 3091.5752 .

Step 5: Subtitute a and b into the solution of equation 2.3 to get fitting 1-AGO $\hat{X}^{(1)}(t)$, where $x^{(1)}(t) = (x^{(0)}(1) - b/a)e^{-at} + b/a$ when $t \leq n$, it called GM(1,1) fitting model, when $t > n$, it called GM(1,1) predictting model. The results as follow in Table 4.4. It can see the result is very worse than the polynomial funtion fitting. Moreover, the results in original paper as follow in Figure 4.10.

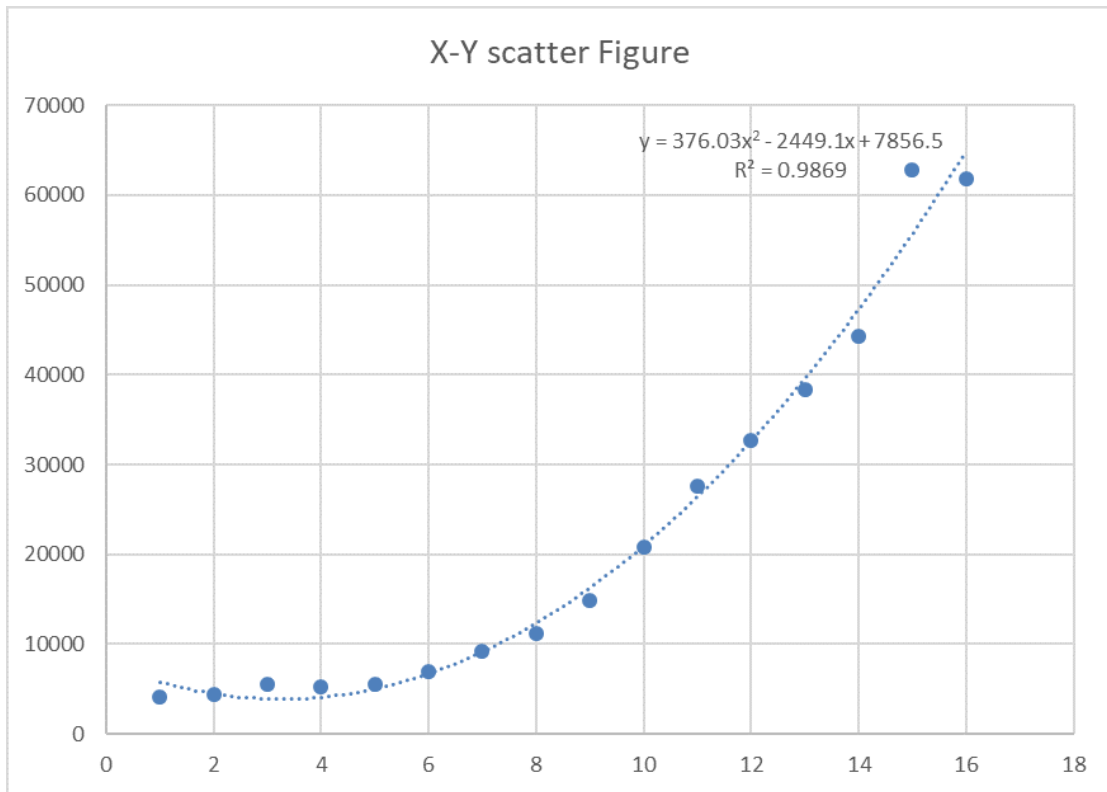


FIGURE 4.9: X-Y scatter chart of original data with polynomial function

GM(1,1)				
Actual	Forecast	abs(Forecast–Actual)/Actual(%)		MAPE(%)
4115	4115	0		16.7569
4387	4355.6928	3.1317		
5510	5337.4437	26.3372		
5177	6540.4763	45.0093		
5527	8014.6662	40.3620		
6997	9821.1311	29.1145		
9231	12034.7640	32.2750		
11149	14747.3383	22.0045		
14812	18071.3132	6.4230		
20808	22144.44949	1.4070		
27523	27135.75108	1.9094		
32629	33252.0109	6.3636		
38309	40746.8445	12.5687		
44356	49930.9753	2.5373		
62778	61185.1624	21.1968		
61863	74975.9857	4.9651		
68608	91875.1903	33.9132		
74360	112583.3894	51.4032		
81910	137959.1109	68.4277		
93234.18	169054.3906	81.3223		
95272.29	207158.3875	117.4382		70.5009

TABLE 4.4: China's iron ore import with classical GM(1,1)

Table 4
Forecasting results of import volume for four models.

Year	Actual import volume (10,000 t)	GM(1,1)		PSO-GM(1,1)		Rolling GM(1,1)		PSO-rolling GM(1,1)	
		Predicted value	PE(%)	Predicted value	PE(%)	Predicted value	PE(%)	Predicted value	PE (%)
1995	4,115	4,115	0	4115	0	4115	0	4,115	0
1996	4,387	4,351.77	0.8031	4,178.89	4.7438	4,651.85	-6.0406	4,453.30	-1.1511
1997	5,510	5,333.15	3.2096	5,088.04	7.6581	5,013.71	9.0018	4,801.64	12.8559
1998	5,177	6,535.86	-26.2480	6,194.99	-19.6637	5,403.72	-4.3848	5,177.23	-0.0044
1999	5,527	8,009.78	-44.9209	7,542.75	-36.4710	5,824.08	-5.3736	5,582.19	-0.9986
2000	6,997	9,816.10	-40.2901	9,183.74	-31.2525	5,422.05	22.5096	5,423.24	22.4919
2001	9,231	12,029.78	-30.3194	11,181.73	-21.1324	8,013.82	13.1838	9,099.19	1.4279
2002	11,149	14,742.66	-32.2330	13,614.40	-22.1132	11,793.81	-5.7853	11,148.96	3.5978e-004
2003	14,812	18,067.34	-21.9777	16,576.31	-11.9114	14,006.66	5.4348	14,811.93	4.7259e-004
2004	20,808	22,141.78	-6.4099	20,182.62	3.0055	18,546.13	10.8708	20,808.000014	-6.5720e-008
2005	27,523	27,135.06	1.4095	24,573.50	10.7165	27,939.29	-1.5115	27,523.000432	-1.5690e-006
2006	32,629	33,254.40	-1.9167	29,919.66	8.3035	36,922.00	-13.157	32,629.03	-9.1942e-005
2007	38,309	40,753.74	-6.3816	36,428.91	4.9077	40,744.60	-6.3588	38,308.93	1.8272e-004
2008	44,356	49,944.29	-12.5987	44,354.30	0.0038	45,064.09	-1.5962	44,356.001	-2.2545e-006
2009	62,778	6,1207.43	2.5018	54,003.91	13.9764	51,613.96	-17.7833	58,345.56	7.0605
2010	61,863	75,010.58	-21.2527	65,752.88	-6.2879	79,193.60	-28.0151	62,950.24	-1.7575
MAPE(%)		16.8315		13.4769		10.0671		3.2072	

FIGURE 4.10: The fitting results in original papre
[14]

Coding description

In order to use conveniently in next section, here, it compiled 1-AGO coefficient matrix A as r-AGO matrix named "mm.m". When it to use, it only need enter the order of AGO and the number of data records. Moreover, the order can be the fractional number which defined in Chapter 2. Therefore, it compiled the fractional factorial named "fractionalfactorial.m" to achieve that. In addition, considering the next model GM(1,1, λ), it compiled the $z^{(1)}(k) = \lambda x^{(1)}(k) + (1 - \lambda)x^{(1)}(k - 1)$ named "Mean.m" where $k = 2, 3, \dots, n$. The last coding is the main coding to fit and forecast with MAPE named "GM11.m". The coding listed in the Appendix B. The main principle to code in MATLAB is transformed the mathematical formula to a form with matrix. In addition, a great deal of conditional and looping statements is necessary to achieve the matrix compiling.

It can see the GM(1,1) results which fitting MAPE equals 16.7569% in this paper is a little better than the results in the Figure 4.10 in the original paper which fitting MAPE equals 16.8315%. There are also some reasons such as the number of decimal retained in the caculation, the version of the MATLAB, the operation system of running computer, etc. Obviously, the results including fitting and forecasting are worse than the results with classical regression methods. Moreover, the methods with EXCEL tools is much easier than the GM(1,1). Therefore, scholars proposed the a great deal of GM(1,1) variants to improve the performance of GM(1,1). Next, it will give GSA-GM(1,1, λ) to analyze it.

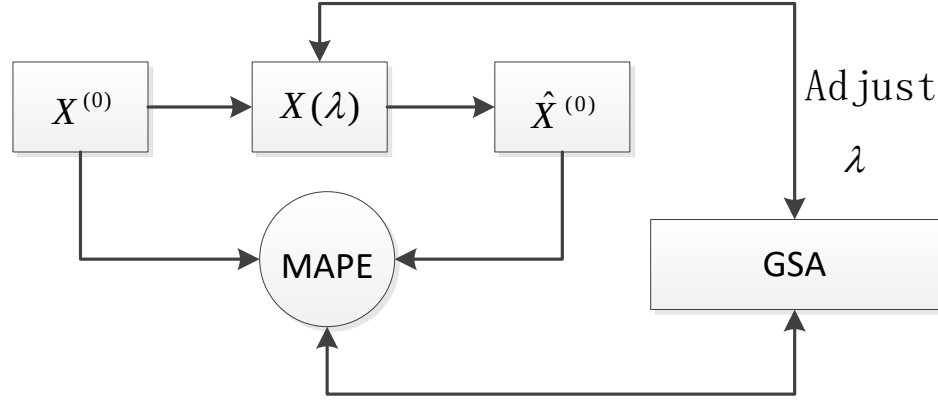


FIGURE 4.11: The schematic diagram of GSA-GM(1,1)

4.2.1.4 Fitting and forecasting used GSA-GM(1,1, λ)

The main steps are similar with PSO-GM(1,1, λ) in the section 4.2.1. It only use GSA to replace the PSO. Here, it gives the GSA-GM(1,1, λ) schematic diagram as follow in Figure 4.11: The main steps in MATLAB software:

Step 1: Modify the "GM11.m" to the "MAPE.m" by adding funtion name "function=MAPE(lambda)" to prepare in the main program can be called as Appendix B.

Step 2: Update "fit=sum(L.^2)" to "fit=MAPE(L)" in the "test_functions.m" and its range from 0 to 1 in the "test_functions.range.m"

Obviously, MAPE function is the 1-dimension function with respect to the "lambda". Considering the objective function can be only one, here, it used fitting MAPE function as objective function at first. Then it used forecasting MAPE function as the objective function. The result as follow in Table 4.5:

Compared to the Figure 4.10, it can see that the fitting MAPE of the GSA-GM(1,1, λ) which equals 13.4116% is a little smaller than the 13.4769% in the Figure 4.10. Obviously, the forecasting MAPE which equals 70.5009% is very bad. Therefore, next, we set the forecasting MAPE function as the objective function to search the optimum.

The result with forecasting MAPE function as the objective function as follow in Table 4.6: It can see that the forecasting MAPE with GSA-GM(1,1, λ) within the forecasting MAPE function as the objective function is much smaller than forecasting MAPE with GM(1,1) which one is 14.3354% and another is 70.5009%. Moreover, the forecasting MAPE of GSA-GM(1,1, λ) is also better than the result of polynomial function which one is 14.3354% and another is 17.6669%.

Coding discription

It only need transform the MAPE function in the "MAPE.m". In the "MAPE.m", it

GSA-GM(1,1)			
Actual	Forecast	abs(Forecast–Actual)/Actual(%)	MAPE(%)
4115	4115	0	13.4116
4387	4183.0948	4.6479	
5510	5092.7489	7.5726	
5177	6200.2160	19.7647	
5527	7548.5124	36.5752	
6997	9190.0089	31.3421	
9231	11188.4644	20.0350	
11149	13621.5033	22.1769	
14812	16583.6299	11.9608	
20808	20189.8993	2.9705	
27523	24580.3865	10.6915	
32629	29925.6274	8.2852	
38309	36433.2421	4.8964	
44356	44356.0000	5.8659	
62778	54001.6376	13.9800	
61863	65744.8117	6.2749	
68608	80041.6516	16.6652	
74360	97447.4765	31.0482	
81910	118638.3649	44.8399	
93234.18	144437.4153	54.9190	
95272.29	175846.7167	84.5728	70.5009

TABLE 4.5: China's iron ore import with GSA-GM(1,1) with fitting MAPE function as the objective function

used the comment symbols which does not participate in the compiled process to change the objective function as in AppendixB.

4.2.1.5 Fitting and forecasting used rolling GM(1,1)

Rolling GM(1,1)

Here, use this data as an example to explain.

Step1 : Select the first four records $X^{(0)}_1 = (4115, 4387, 5510, 5177)$ as the testing data, use GSA to predict the fifth record $x^{(0)}(5) = 5824.0751$. Predicting methods is similar with subsection 4.2.1.3. Here, it will not repeated again.

Step2: Add the fifth record into the $X^{(0)}_1$, and remove the first record 4115. Then it get a new sequence testing sequence named $X^{(0)}_2 = (4387, 5510, 5177, 5824.5527)$. After that, use the same step in the above to get sixth record. Then repeat that until to predict all data. The result as follow in Table

Until now, it can see the best results with fitting MAPE or forecasting MAPE are the results used Rolling GM(1,1). The concept of this model is when a dataset have different

GSA-GM(1,1, λ)			
Actual	Forecast	abs(Forecast – Actual)/Actual(%)	MAPE(%)
4115	4115	0	16.5324
4387	3873.5510	11.7039	
5510	4660.1437	15.4239	
5177	5606.4679	8.2957	
5527	6744.9598	22.0365	
6997	8114.6426	15.9732	
9231	9762.4635	4.7362	
11149	11744.9035	5.7362	
14812	14129.9129	4.6050	
20808	16999.2403	18.3043	
27523	20451.2351	25.6940	
32629	24604.2183	24.5940	
38309	29600.5377	22.7321	
44356	35611.4477	19.7145	
62778	42842.9788	31.7548	
61863	51542.9994	16.6820	
68608	62009.7124	9.6174	
74360	74601.8756	0.3253	
81910	89751.0990	9.5728	
93234.18	107976.6389	15.8123	
95272.29	129903.1953	36.3494	14.3354

TABLE 4.6: China's iron ore import with GSA-GM(1,1, λ) with fitting MAPE function as the objective function

Year	Actual	forecasting of PSO-rolling GM (1,1)	abs(Forecast-Actual)/Actual	MAPE
2011	68608	71443.7	0.04133	
2012	74360	76534.7	0.02925	
2013	81910	84819.41	0.03553	
2014	93234.18	92540.76	0.0075	
2015	95272.29	101691.14	0.0674	3.62%

FIGURE 4.12: The results in original paper

trends in different intervals, it can divide it some subdataset to fitting and forecasting. In this model, the $\lambda = 0.5$ Next, it will try the GSA-rolling GM(1,1, λ) with this data.

4.2.1.6 Fitting and forecasting used GSA-rolling GM(1,1, λ)

The methods in this subsubsection is combined GSA-GM(1,1, λ) and rolling GM(1,1). Here, it will not introduce in detail. The result as follow in Table 4.8:

In order to compare the forecasting MAPE of PSO-rolling GM(1,1) in Figure 4.12 and GSA-rolling GM(1,1), it computed the forecasting MAPE of PSO-rolling GM(1,1) used actual value in the Figure 4.12. The result is 3.62%.

Rolling-GM(1,1)			
Actual	Forecast	abs(Forecast–Actual)/Actual(%)	MAPE(%)
4115	4115	0	10.2458
4387	4651.8459	6.0371	
5510	5013.7099	9.0071	
5177	5403.7231	4.3794	
5527	5824.0751	5.3750	
6997	5422.0523	22.5089	
9231	8013.8238	14.0240	
11149	11959.0600	7.2657	
14812	14018.0819	5.3600	
20808	18454.8827	11.3087	
27523	27939.2882	1.5125	
32629	36921.9992	13.1570	
38309	40744.5962	6.3578	
44356	45064.0871	1.5964	
62778	51613.9604	17.7834	
61863	79193.6017	28.0145	
68608	74902.8590	9.1751	
74360	70580.0318	5.0833	
81910	81654.3548	0.3121	
93234.18	89262.9642	4.2594	
95272.29	103917.5636	9.0743	5.581

TABLE 4.7: China's iron ore import with rolling-GM(1,1)

GSA-Rolling GM(1,1, λ)			
Actual	Forecast	abs(Forecast–Actual)/Actual(%)	MAPE(%)
4115	4115	0	7.8362
4387	4453.2489	1.5101	
5510	4801.5855	12.8569	
5177	5177.1693	0.0033	
5527	5582.1316	0.9975	
6997	5421.1139	22.5223	
9231	8221.9082	11.7916	
11149	11833.3696	6.1384	
14812	14773.6912	0.2586	
20808	18853.7498	9.3918	
27523	27763.2177	0.8728	
32629	36322.3219	11.3191	
38309	39245.7107	2.445	
44356	45078.5037	1.6289	
62778	52062.2378	17.0693	
61863	73454.4968	18.7374	
68608	68608.0000	0	
74360	71872.8854	3.3447	
81910	81909.9999	4.7186e-11	
93234.18	89556.4064	3.9447	
95272.29	102739.6065	7.8379	3.0255

TABLE 4.8: China's iron ore import with GSA-rolling GM(1,1, λ)

Algorithm	MAPE of GM(1,1, λ)(fitting)	MAPE of rolling GM(1,1, λ)(fitting)	MAPE of rolling GM(1,1, λ)(forecasting)
PSO	13.4769%	3.2072%	3.62%
GSA	13.4116%	7.8362%	3.0255%

TABLE 4.9: Compare the result of GSA and PSO

Compare the result of GSA and PSO

The result as follow in Table 4.9

It can see that the fitting result of PSO-rolling GM(1,1, λ) is better than the fitting result of GSA-rolling GM(1,1, λ). However, the forecasting result of GSA-rolling GM(1,1, λ) is better than the PSO-rolling forecasting result.

In conclusion, in this subsection, it chose four models and two classical regression methods to fit and forecast. Firstly, all results in the experiment are better than the existing result in the [14] except the fitting MAPE of PSO-rolling GM(1,1). Secondly, GM(1,1) result is the worst. The result of classical regression methods is similar of GM(1,1) with parameters λ . The rolling GM(1,1) can get high accuracy of fitting and forecasting. In addition, when it use PSO and GSA to find the optimum in grey model, it is hard to say which one is best due the diversity of grey models.

4.2.2 The second grey model is chosen in this paper

In this reference paper [3], it only used the fractional grey model to fit instead of forecasting. In order to compare PSO and GSA, it requires that the other conditions must be the same. Therefore, in this subsection, it only fit the data.

Before this model, it add a previously mentioned concept.

Definition 4.2. As mentioned of definition 2.2 in the Chapter 2, the fitting and forecasting of $X^{(0)}$ is $\hat{x}^{(0)}(k) = \hat{x}^{(1)}(k) - \hat{x}^{(1)}(k-1)$, $k = 2, 3, \dots, n$, $\hat{x}^{(0)}(1) = \hat{x}^{(1)}(1)$. It can be

$$\text{transformed as follow: } (\hat{x}^{(0)}(1), \hat{x}^{(0)}(2), \dots, \hat{x}^{(0)}(n)) = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -1 & 1 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & 0 \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & -1 & 1 & 0 \\ 0 & 0 & \dots & 0 & -1 & 1 \end{pmatrix}_{n \times n} \cdot (\hat{x}^{(1)}(1), \hat{x}^{(1)}(2), \dots, \hat{x}^{(1)}(n))^{\top}$$

It can see this coefficient matrix is the inverse of coefficient matrix A in the definition 2.2. It called 1 order inverse accumulated generating operator (1-IAGO).

Number	Actual1	Actual2	Actual3
1	0.26	34.15	357.4
2	0.73	12.74	164.6
3	2.06	5.77	85.1
4	7.08	2.76	59.4
5	21.22	1.75	40.3
6	63.98	1.23	31.0
7	202.51	0.90	
8		0.74	

TABLE 4.10: The three data from reference

[3]

Definition 4.3. [3] Set $z^{(r)}(k) = 0.5[x^{(r)}(k) + x^{(r)}(k-1)]$ is the mean generating sequence of r-AGO sequence, $k = 2, 3, \dots, n$, a is the development coefficient of the model, b is the grey input of the model

$$x^{(r-q)}(k) + az^{(r)}(k) = b \quad (4.5)$$

And the whitenization formula of the FGM(q,1)model is

$$\frac{d^q x^{(r)}}{dt^q} + ax^{(r)} = b \quad (4.6)$$

Where $X^{(r-q)}$ means that using r-AGO to handle the original sequence $X^{(0)}$ at first to get $X^{(r)}$, then used the q-IAGO to handle the $X^{(r)}$ to get $X^{(r-q)}$. In [3], it gived the proof in detail between equation 4.5 and between 4.6.

4.2.2.1 Testing data

It can see the testing data in this model as follow in Table 4.10. There are three datasets in total which the first one has 7 records, the second one has 8 records and the last one has 6 records.

4.2.2.2 Fitting with EXCEL

The result of Actual1 data as follow in Table 4.11 and Figure 4.13 The fitting curve of Actual1 dataas follow:

The result of Actual 2 as follow in Table 4.12 and Figure 4.14 The fitting curve of Actual2 dataas follow:

The result of Actual 3 as follow in Table 4.13 and Figure 4.15 The fitting curve of

Actual1	Fit	abs(Fit–Actual1)/Actual	MAPE(%)
0.26	0.24334242	0.064067617	
0.73	0.742980341	0.017781289	
2.06	2.268489762	0.101208622	
7.08	6.926220672	0.021720244	
21.22	21.14734375	0.003423951	
63.98	64.56770136	0.009185704	
202.51	197.1400336	0.026517043	3.4843496

TABLE 4.11: The fitting result of exponential function

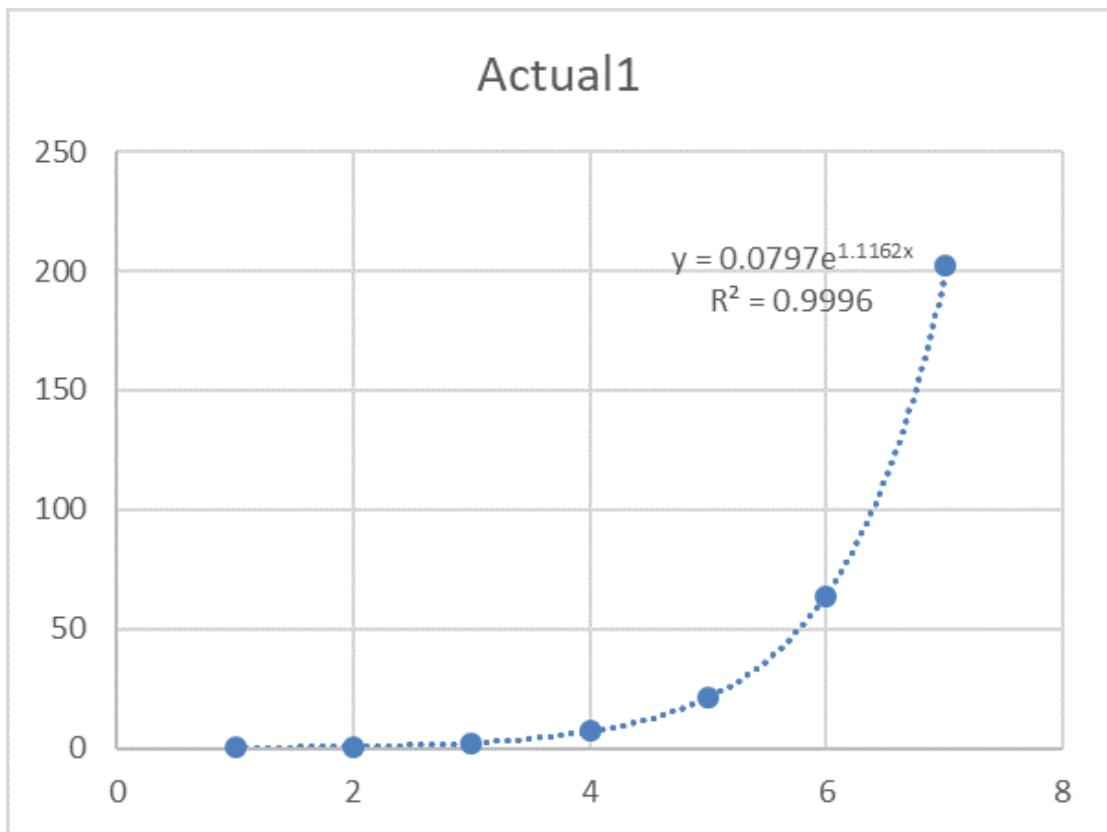


FIGURE 4.13: The fitting result of Actual1 data

Actual2	Fit	abs(Fit–Actual1)/Actual	MAPE(%)
34.15	40.471	0.1851	
12.74	10.6650	0.1629	
5.77	4.8884	0.1528	
2.76	2.8105	0.01829	
1.75	1.8295	0.0454	
1.23	1.2882	0.0473	
0.9	0.9576	0.0640	
0.74	0.7406	0.0008	8.458

TABLE 4.12: The fitting result of power function

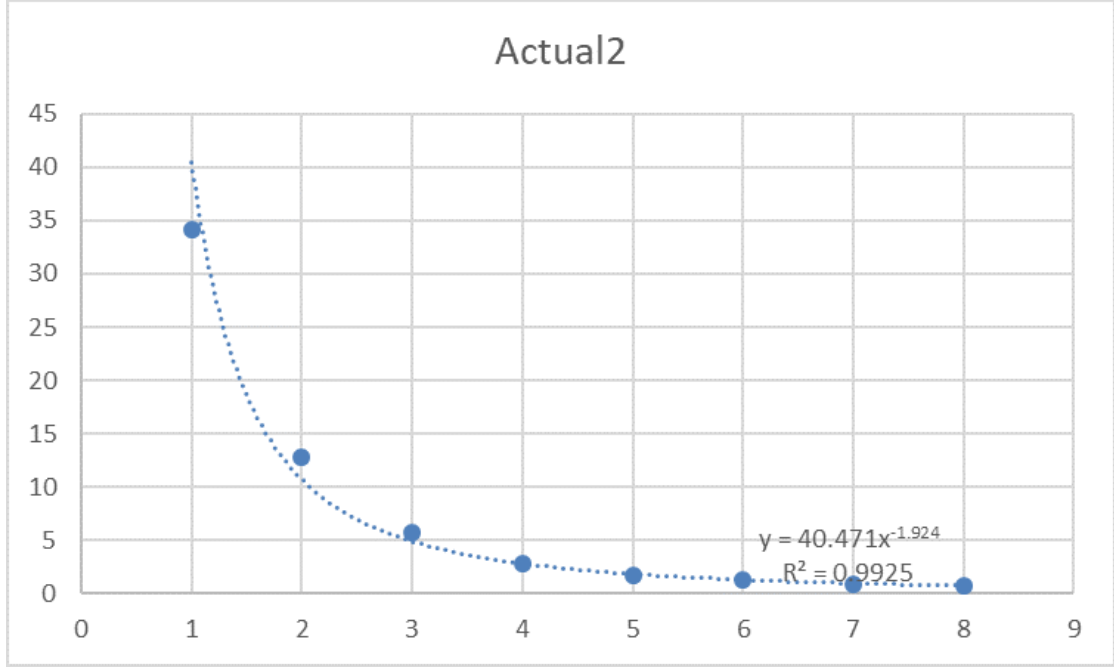


FIGURE 4.14: The fitting result of Actual2 data

Actual3	Fit	abs(Fit-Actual1)/Actual	MAPE(%)
357.4	384.96	0.0771	
164.6	147.9089	0.1014	
85.1	84.5256	0.00675	
59.4	56.8294	0.04328	
40.3	41.7674	0.03641	
31	32.4763	0.0476	5.21

TABLE 4.13: The fitting result of power function

Actual3 dataas follow:

The result of reference paper [3] as follow from Figure 4.16 to 4.18:

It can see that the results of EXCEL have already high accuracy. However, it can see in Figure 4.16, Figure 4.17 and Figure 4.18, Mao et al used the r and q as the parameters to search the minimum of MAPE. All results is much better than classical regression methods of EXCEL. Next, it will use GSA to replace PSO based on the same other conditions.

The schematic diagram is similar with Figure 4.11 except used $X(r, q)$ to replace $X(\lambda)$. Here, it have to be aware of that is the compiling of coefficient matrix $A(r, q)$ which expanded fractional order AGO coefficient matrix. It is a little complicated as showed in the Chapter 2. The main steps of compiling as follow:

- It need to compile the position element of the matrix at first where $j \leq i$, $a_{ij}^r = \frac{(i-j+r-1)!}{(r-1)!(i-j)!}$, otherwise it equals 0 of $A^r = (a_{ij}^r)_{n \times n}$. It can find the elements of

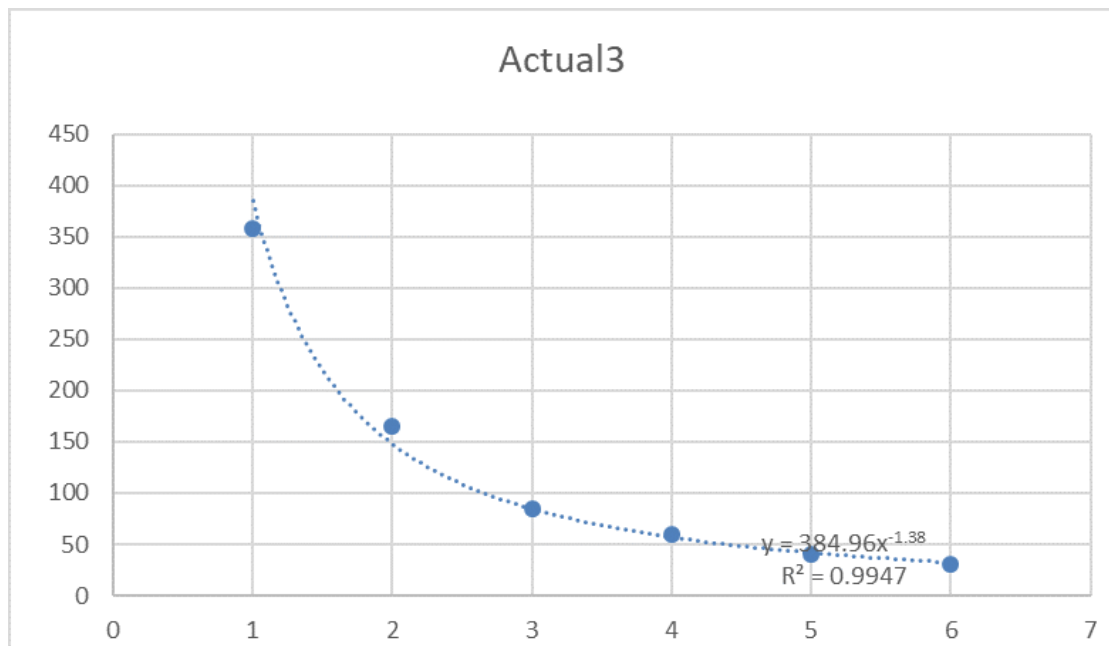


FIGURE 4.15: The fitting result of Actual3 data

The results in original paper			MAPE
Actual 1	Fit	abs(Fit-Actual)/Actual(%)	
0.26	0.26	0	
0.73	0.76	3.78	
2.07	2.14	3.23	
7.08	6.64	6.23	
21.22	20.61	2.88	
63.98	63.98	0	
202.51	198.61	1.92	3.01%

The results in original paper			MAPE
Actual 2	Fit	abs(Fit-Actual)/Actual(%)	
34.15	34.15	0	
12.74	12.79	0.36	
5.77	5.6	2.99	
2.76	2.89	4.7	
1.75	1.75	0	
1.23	1.21	0.76	
0.9	0.92	1.85	
0.74	0.74	0	1.66%

FIGURE 4.16: The original result in paper [3] of Actual1 data

FIGURE 4.17: The original result in paper [3] of Actual2 data

The results in original paper			MAPE
Actual 3	Fit	abs(Fit-Actual)/Actual(%)	
357.4	357.4	0	
164.6	165	0.24	
85.1	85.1	0	
59.4	57.7	2.86	
40.3	41.8	3.73	
31	31	0	1.37%

FIGURE 4.18: The original result in paper[3] of Actual3 data

Actual1	Fit	abs(Fit–Actual1)/Actual	MAPE
0.26	0.26	0	
0.73	0.73	0	
2.06	2.0874	0.0084	
7.08	6.4270	0.0922	
21.22	21.8903	0.0316	
63.98	65.9204	0.0303	
202.51	198.8696	0.0180	3.01%

TABLE 4.14: The Actual1 data used GSA-FGM_{q,1}

the diagonal and the slash below the diagonal are the same. Therefore, it only need to compile the the first column. Moreover, the elements of the first column is the factorial process with respect to row number. At last, it used two condition statements and one loop statement to compile it as a preparation file with the AGO order r and the numbers of records n which named *fractionalfactorial.m*. In classical GM(1,1), it only need to enter $r = 1$. The coding can be seen in the AppendixA.

- After that, it need to compile all elements of matrix $A(r, n)$, where n is the number of records. The compiling priciple is that the elements of the diagonal and the slash below the diagonal are the same. It also need to combine the condition statement and loop statement which named *mm.m*. The coding can be seen in the AppendixA.

The other difference is the compiling of $X(r, q)$. As mentioned of the difinition of the IAGO, the coefficient matrix of IAGO is the inverse of the AGO. Therefore, their power matrixs are also inverse of each other. Moreover, in [3], it had improved the inverse matrix $A^{r(-1)}$ of A^r is use $-r$ to replace r . That means $A^{r(-1)}$ in MATLAB is $A(-r, n)$.

In addition, the objective funtion is a two-dimension function with respect to r and q . It also need to change the *test_functions.m* and *test_functions_range.m*. Here, it used 0 to 3 as the range of parameteres. Both of them and new MAPE file can bee seen in the AppendixC.

The running result of FGM($q, 1$) with GSA as follow:

Actual1 data

The fitting result as follow in Table 4.14

Actual2 data

The fitting result as follow in Table 4.15

Actual3 data

The fitting result as follow in Table 4.16

Actual2	Fit	abs(Fit–Actual1)/Actual	MAPE(%)
34.15	34.15	0	
12.74	12.8134	0.0058	
5.77	5.5013	0.0466	
2.76	2.9674	0.0752	
1.75	1.7494	0.0003	
1.23	1.2265	0.0028	
0.9	0.9324	0.0359	
0.74	0.74	0	2.38

TABLE 4.15: The Actual2 data used GSA-FGM_{q,1}

Actual3	Fit	abs(Fit–Actual1)/Actual(%)	MAPE(%)
357.4	357.4	0	
164.6	164.285	0.0019	
85.1	86.6513	0.0182	
59.4	56.8322	0.0432	
40.3	41.8392	0.0382	
31	31.0010	0.0000	2.03%

TABLE 4.16: The Actual3 data used GSA-FGM(_{q,1})

Algorithm	MAPE of Actual1	MAPE of Actual2	MAPE of Actual3
PSO	3.01%	1.66%	1.37%
GSA	3.01%	2.38%	2.03%

TABLE 4.17: Compare the result of GSA and PSO

Compare the result of GSA and PSO

The result as follow in Table 4.17

It can see the result in Table 4.17. In the fitting MAPE of Actual1 data, both of them get the same MAPE. However, in the fitting MAPE of Actual2 and Actual3, the result of PSO is better than the result of GSA.

4.3 Data Evaluation

Table 4.9 and Table 4.17 showed the result of GSA and PSO. It can see some is better than PSO and some is not better. In [1], "In order to solve the high dimensions problems,it proposed GSA". Therefore, it combined this characteristic and grey model itself to expand the dimension to try to improve the accuracy when GSA applied in grey model. In addition, the grey model in the [3] can only fit the data. Therefore, this new model can only fit the data as well.

It combined the GM(1,1, λ) and FGM($q,1$). Now, the objective function is a 3-dimensional function with respect to the λ, r, q . In the MATLAB coding, it need to use the "Mean.m"

Actual3	Fit	abs(Fit–Actual1)/Actual	MAPE(%)
357.4	357.4	0	
164.6	164.60	0	
85.1	86.5403	0.0169	
59.4	56.7608	0.0443	
40.3	41.7637	0.0362	
31	31.0000	0	1.95%

TABLE 4.18: The Actual3 data used GSA-FGM($q,1,\lambda$)

Actual2	Fit	abs(Fit–Actual1)/Actual	MAPE(%)
34.15	34.15	0	
12.74	12.7399	7.8e-06	
5.77	5.5542	0.0374	
2.76	2.9807	0.077996	
1.75	1.7500	0.000	
1.23	1.2264	0.0029	
0.9	0.9324	1.3169e-06	
0.74	0.74	0	2.23

TABLE 4.19: The Actual3 data used GSA-FGM($q,1,\lambda$)

Actual1	Fit	abs(Fit–Actual1)/Actual	MAPE(%)
0.26	0.26	0	
0.73	0.73	0	
2.06	2.0699	4.336e-14	
7.08	6.3967	0.0965	
21.22	21.8245	0.0285	
63.98	65.5807	0.0250	
202.51	197.7734	0.0234	2.89%

TABLE 4.20: The Actual3 data used GSA-FGM($q,1,\lambda$)

in GM($1,1,\lambda$) coding to replace the "Mean.m" in FGM($q,1$) coding. Moreover, it need modify the *test_functions.m* and "test_functions.range.m" as well. It can see the coding in the AppendixD.

The running results as follow in Table 4.18 to Table 4.21:

The fitting results of GM($1,1,\lambda$), FGM($q,1$), FGM($q,1,\lambda$) based on PSO and GSA respectively as follow in Table 4.22 and 4.23:

Compare the result of GSA and PSO

The result as follow in Table 4.22 and Table 4.23

It can see that the fitting results in Table 4.22 and Table 4.23, The new grey model GSA-FGM($q,1,\lambda$) results is better than the results of GSA-FGM($1,1,\lambda$) and PSO-FGM($1,1,\lambda$).

GSA- GM(q,1, λ)			
Actual	Forecast	abs(Forecast – Actual)/Actual	MAPE(%)
4115	4115	0	9.18
4387	4380.5405	0.0015	
5510	5537.4237	0.0050	
5177	6922.1782	0.3371	
5527	6635.8549	0.2006	
6997	7038.9921	0.006	
9231	8711.8265	0.0654	
11149	11405.2545	0.02298	
14812	13608.8275	0.0812	
20808	17877.6774	0.1408	
27523	24871.1924	0.0963	
32629	32835.6181	0.0063	
38309	39097.5498	0.0206	
44356	45973.5163	0.0365	
62778	53293.9935	0.1511	
61863	74558.0855	0.2052	

TABLE 4.21: China's iron ore import with GSA-GM(q,1, λ)

Algorithm	MAPE of Actual1	MAPE of Actual2	MAPE of Actual3
PSO-FGM(q,1)	3.01%	1.66%	1.37%
GSA-FGM(q,1)	3.01%	2.38%	2.03%
GSA-FGM(q,1, λ)	2.89%	2.23%	1.95%

TABLE 4.22: Compare the result of GSA and PSO

Algorithm	MAPE of China's iron import
PSO-FGM(1,1, λ)	13.4769%
GSA-FGM(1,1, λ)	13.4116%
GSA-FGM(q,1, λ)	9.18%

TABLE 4.23: Compare the result of GSA and PSO

However, the GSA-FGM(q,1, λ) results is better than the results of the GSA-FGM(q,1) while is also worse than the results of PSO-FGM(q,1).

Chapter 5

Summary

In conclusion, this paper compared four methods which is EXCEL regression tools, Grey model, Grey model with PSO and Grey model with GSA to fit and forecast four datasets. Moreover, it is the first time to use the GSA to handle the parameters problems of Grey models.

At first, in the results of China's iron import dataset, the GM(1,1)'s is worse than the EXCEL regression tools' while EXCEL regression tools' is similar with the GSA-GM(1,1, λ) and PSO-GM(1,1, λ) and rolling GM(1,1). However, the fitting results of the PSO-rolling GM(1,1, λ) are the best results. The forecasting results of GSA-rolling GM(1,1, λ) are the better results than PSO-rolling GM(1,1, λ)'s.

Secondly, in the two-thirds results of the datasets in the [3], GSA results are worse than the results of PSO while all results of GSA or PSO are better than the regressions tools'. Considering the some results of GSA is not better than the PSO's, in this paper, we proposed a new grey model GSA-FGM(q,1, λ) which combined the GSA characteristic which main solved the high dimensional problems. However, the results are also not better than the PSO while it is better than the GSA-FGM(q,1)'s.

At last, due to the objectives of this paper, we compared the PSO and GSA when it applied in the same grey model although that grey model may not be the grey model with higher accuracy. In the first dataset, the GSA-GM(1,1, λ)'s results are better than the PSO-GM(1,1, λ). The rolling PSO-GM(1,1, λ) fitting results is better than the rolling GSA-GM(1,1, λ) while in this rolling GM(1,1, λ), the GSA's forecasting is better than the PSO's. In the second datasets[3], the one-third datasets showed that the GSA-FGM(q,1)'s fitting results equals the PSO-FGM(q,1). Therefore, it is hard to judge the GSA and PSO which one is better when they applied in the grey models. Noticed that both of the China's iron dataset and the first dataset in the [3] are monotonically increasing. Most of GSA's results are better than PSO's. While the other two datasets are monotonically decreasing. PSO's fitting results are better than GSA's based on

these two datasets. Maybe when the datasets with increasing trends, it can choose the GSA in the grey model. Otherwise, it can choose PSO.

The problem in this paper is still worthy to investigate. In addition, this paper provided a guide when scholars use GSA applied in the grey model.

Chapter 6

Appendix A

[1] **massCalculation.m** % GSA code v1.1.

% Generated by Esmat Rashedi, 2010.

% " E. Rashedi, H. Nezamabadi-pour and S. Saryazdi,

%GSA: A Gravitational Search Algorithm Information sciences, vol. 179,

%no. 13, pp. 2232-2248, 2009." The coding as follow in Figure 5.1.

Gconstant.m

% This function calculates Gravitational constant. eq.13.

The coding as follow in Figure 5.2.

test_functions.m

% This function calculates the value of the objective function.

The coding as follow in Figure 5.3. Here, it only lists the first eight test functions as an example.

test_functions_range.m

% This function gives boundaries and dimension of search space for test functions. The coding as follow in Figure 5.4. Here, it only lists the first eight test functions as an example as well.

main.m

% Main function for using GSA algorithm. The coding as follow in Figure 5.5.

```

function [M]=massCalculation(fit,min_flag);
    %%%here, make your own function of 'mass calculation'

    Fmax=max(fit); Fmin=min(fit); Fmean=mean(fit);
    [i N]=size(fit);

    if Fmax==Fmin
        M=ones(N,1);
    else

        if min_flag==1 %for minimization
            best=Fmin;worst=Fmax; %eq.17-18.
        else %for maximization
            best=Fmax;worst=Fmin; %eq.19-20.
        end

        M=(fit-worst)./(best-worst); %eq.15,

    end

M=M./sum(M); %eq. 16.

```

FIGURE 6.1: massCalculation.m

```

function G=Gconstant(iteration,max_it)
    %%%here, make your own function of 'G'

    alfa=20;G0=100;

    G=G0*exp(-alfa*iteration/max_it); %eq. 28.

```

FIGURE 6.2: Gconstant.m

```

function fit=test_functions(L,F_index,dim)
%Insert your own objective function with a new F_index.
if F_index==1
    fit=sum(L.^2);
end

if F_index==2
    fit=sum(abs(L))+prod(abs(L));
end

if F_index==3
    fit=0;
    for i=1:dim
        fit=fit+sum(L(1:i))^2;
    end
end

if F_index==4
    fit=max(abs(L));
end

if F_index==5
    fit=sum(100*(L(2:dim)-(L(1:dim-1).^2)).^2+(L(1:dim-1)-1).^2);
end

if F_index==6
    fit=sum(floor((L+.5)).^2);
end

if F_index==7
    fit=sum([1:dim].*(L.^4))+rand;
end

if F_index==8
    fit=sum(-L.*sin(sqrt(abs(L))));
end

```

FIGURE 6.3: test_functions.m

```

function [down,up,dim]=test_functions_range(F_index)
%If lower bounds of dimensions are the same, then 'down' is a value.
%Otherwise, 'down' is a vector that shows the lower bound of each dimension.
%This is also true for upper bounds of dimensions.

%Insert your own boundaries with a new F_index.

dim=30;
if F_index==1
    down=-100;up=100;
end

if F_index==2
    down=-10;up=10;
end

if F_index==3
    down=-100;up=100;
end

if F_index==4
    down=-100;up=100;
end

if F_index==5
    down=-30;up=30;
end

if F_index==6
    down=-100;up=100;
end

if F_index==7
    down=-1.28;up=1.28;
end

if F_index==8
    down=-500;up=500;
end

```

FIGURE 6.4: test_functions_range.m

```

clear all;clc
% inputs:
% N: Number of agents.
% max_it: Maximum number of iterations (T).
% ElitistCheck: If ElitistCheck=1, algorithm runs with eq.21 and if =0, runs with eq.9.
% Rpower: power of 'R' in eq.7.
% F_index: The index of the test function. See tables 1,2,3 of the mentioned article.
%           Insert your own objective function with a new F_index in 'test_functions.m'
%           and 'test_functions_range.m'.
% outputs:
% Fbest: Best result.
% Lbest: Best solution. The location of Fbest in search space.
% BestChart: The best so far Chart over iterations.
% MeanChart: The average fitnesses Chart over iterations.
N=50;
max_it=1000;
ElitistCheck=1; Rpower=1;
min_flag=1; % 1: minimization, 0: maximization
F_index=1
[Fbest,Lbest,BestChart,MeanChart]=GSA(F_index,N,max_it,ElitistCheck,min_flag,Rpower);Fbest,
semilogy(BestChart,'--k');
title(['\fontsize{12}\bf F',num2str(F_index)]);
xlabel('\fontsize{12}\bf Iteration');ylabel('\fontsize{12}\bf Best-so-far');| legend('\fontsize{10}\bf GSA',1)

```

FIGURE 6.5: main.m

Chapter 7

AppendixB

mm.m, fractionalfactorial.m and Mean.m

The coding as follow in Figure 6.1.

GM11.m

The coding as follow in Figure 6.2.

MAPE.m

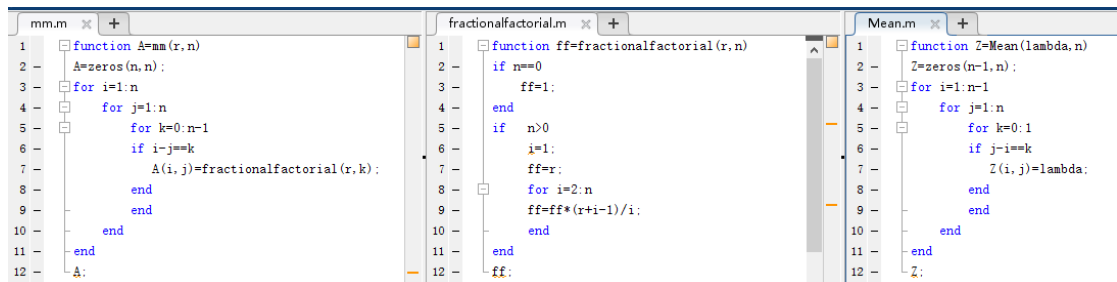


FIGURE 7.1: The main coding of GM(1,1)

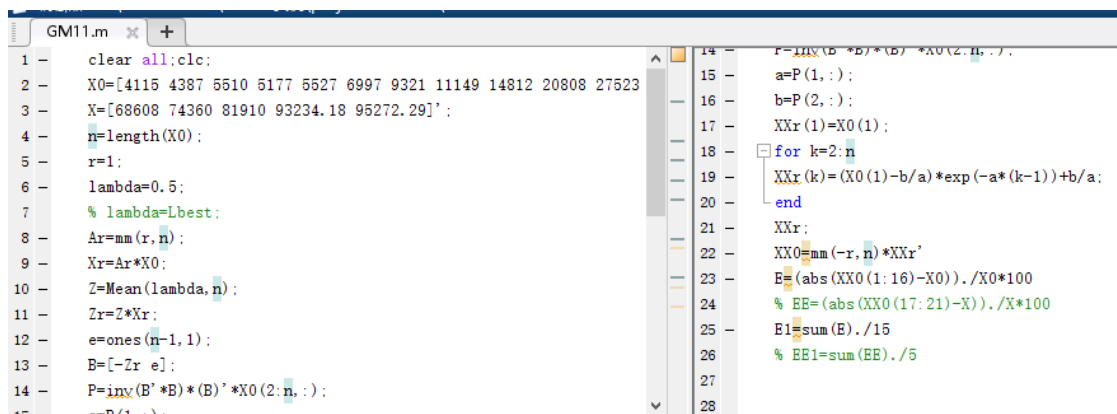


FIGURE 7.2: The main running coding of GM(1,1)

```

1 function MAPE=MAPE(lambda)
2 X0=[4115 4387 5510 5177 5527 6997 9321 11149 14812 20808 27523
3 X=[68608 74360 81910 93234.18 95272.29]';
4 n=length(X0);
5 r=1;
6 %lambda=0.5;
7 Ar=mm(r,n);
8 Xr=Ar*X0;
9 Z=Mean(lambda,n);
10 Zr=Z*Xr;
11 e=ones(n-1,1);
12 B=[-Zr e];
13 P=inv(B'*B)*(B')*X0(2:n,:);
14 a=P(1,:);
15 b=P(2,:);
16 XXr(1)=X0(1);
17 for k=2:n
18 XXr(k)=(X0(1)-b/a)*exp(-a*(k-1))+b/a;
19 end
20 XXr;
21 %XX0=mm(-r,n)*XXr';
22 XX0=mm(-r,n)*XXr';
23 %E=(abs(XX0-X0))./X0*100
24 E=(abs(XX0(1:16)-X0))./X0*100
25 % EE=(abs(XX0(17:21)-X))./X*100
26 E1=sum(E)./15
27 % EE1=sum(EE)./5
28 MAPE=E1
29

```

FIGURE 7.3: MAPE function coding of GM(1,1)

The coding as follow in Figure 6.3.

Chapter 8

Appendix C

MAPE.m

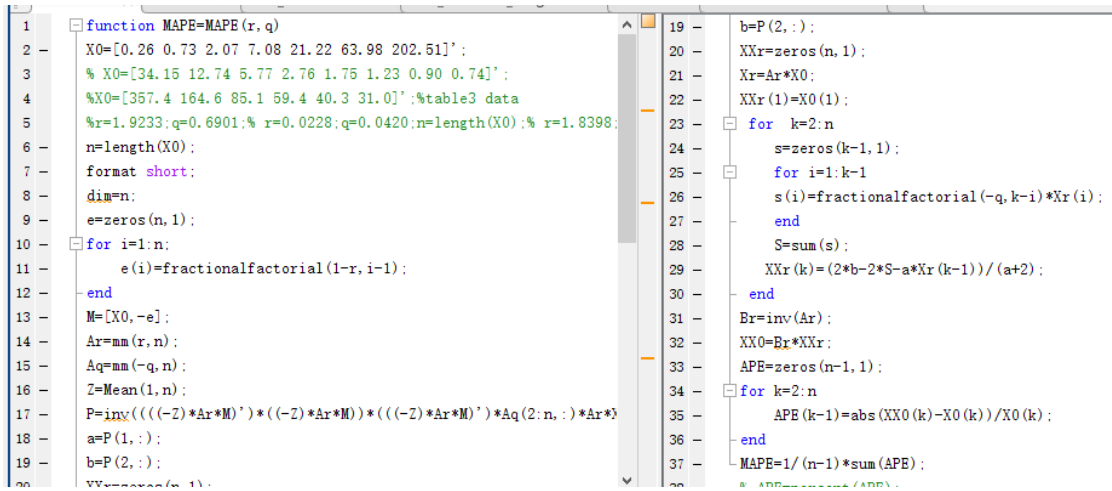
The coding as follow in Figure 8.1.

test_functions.m

% This function calculates the value of the objective function. The coding as follow in Figure 7.2.

test_functions_range.m

% This function gives boundaries and dimension of search space for test functions. The coding as follow in Figure 7.3.



```
1 function MAPE=MAPE(r,q)
2 X0=[0.26 0.73 2.07 7.08 21.22 63.98 202.51]';
3 % X0=[34.15 12.74 5.77 2.76 1.75 1.23 0.90 0.74]';
4 %X0=[357.4 164.6 85.1 59.4 40.3 31.0]';%table3 data
5 %r=1.9233;q=0.6901;% r=0.0228;q=0.0420;n=length(X0);% r=1.8398;
6 n=length(X0);
7 format short;
8 dim=n;
9 e=zeros(n,1);
10 for i=1:n;
11 e(i)=fractionalfactorial(1-r,i-1);
12 end
13 M=[X0,-e];
14 Ar=mm(r,n);
15 Aq=mm(-q,n);
16 Z=Mean(1,n);
17 P=inv((((-Z)*Ar*M) *((-Z)*Ar*M) *(((-Z)*Ar*M) *Aq(2:n,:) *Ar*
18 a=P(1,:);
19 b=P(2,:);
20 b=P(2,:);
21 XXr=zeros(n,1);
22 Xr=Ar*X0;
23 XXr(1)=X0(1);
24 for k=2:n
25 s=zeros(k-1,1);
26 for i=1:k-1
27 s(i)=fractionalfactorial(-q,k-i)*Xr(i);
28 end
29 S=sum(s);
30 XXr(k)=(2*b-2*S-a*Xr(k-1))/(a+2);
31 end
32 Br=inv(Ar);
33 XX0=Br*XXr;
34 APE=zeros(n-1,1);
35 for k=2:n
36 APE(k-1)=abs(XX0(k)-X0(k))/X0(k);
37 end
38 MAPE=1/(n-1)*sum(APE);
39 % APE=mean(APE);
```

FIGURE 8.1: MAPE function of FGM(q,1).m

```

% GSA code v1.1.
% Generated by Esmat Rashedi, 2010.
% " E. Rashedi, H. Nezamabadi-pour and S. Saryazdi,
%GSA: A Gravitational Search Algorithm? Information sciences, vol. 179,
%no. 13, pp. 2232-2248, 2009."
%
% This function calculates the value of the objective function.
function fit=test_functions(L,F_index,dim)

%Insert your own objective function with a new F_index.

if F_index==1
fit=MAPE(L(1),L(2));
end

```

FIGURE 8.2: test_functions.m in $FGM(q,1)$

```

% GSA code v1.1.
% Generated by Esmat Rashedi, 2010.
% " E. Rashedi, H. Nezamabadi-pour and S. Saryazdi,
%GSA: A Gravitational Search Algorithm? Information sciences, vol. 179,
%no. 13, pp. 2232-2248, 2009."
%
% This function gives boundaries and dimension of search space for test functions.
function [down,up,dim]=test_functions_range(F_index)

%If lower bounds of dimensions are the same, then 'down' is a value.
%Otherwise, 'down' is a vector that shows the lower bound of each dimension.
%This is also true for upper bounds of dimensions.

%Insert your own boundaries with a new F_index.

dim=2;
if F_index==1
    down=0;up=3;
end

```

FIGURE 8.3: test_functions_range.m in $FGM(q,1)$

Chapter 9

AppendixD

[1] **test_functions.m** % GSA code v1.1.

% Generated by Esmat Rashedi, 2010.

% " E. Rashedi, H. Nezamabadi-pour and S. Saryazdi,

%GSA: A Gravitational Search Algorithm Information sciences, vol. 179,

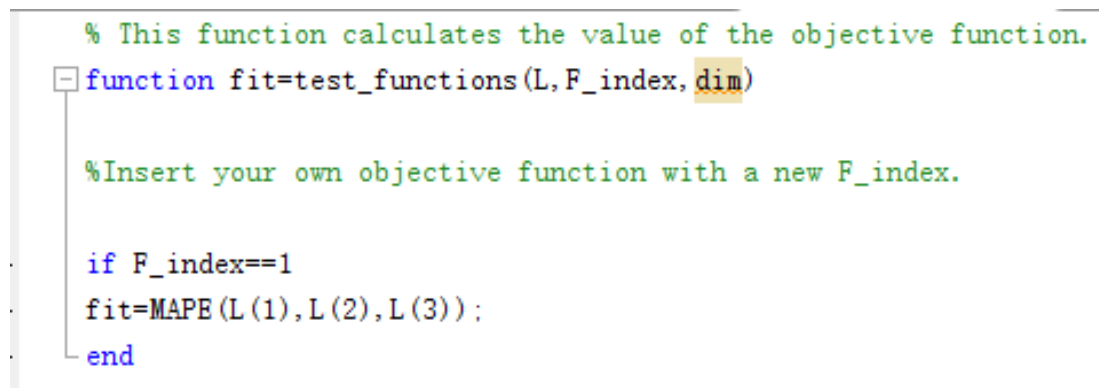
%no. 13, pp. 2232-2248, 2009." The coding as follow in Figure 8.1. % This function calculates the value of the objective function.

test_functions_range.m

% This function calculates objective function range

The coding as follow in Figure 8.2.

MAPE.m

A screenshot of a MATLAB code editor showing the beginning of a function named 'test_functions'. The code is as follows:

```
% This function calculates the value of the objective function.  
function fit=test_functions(L,F_index,dim)  
  
%Insert your own objective function with a new F_index.  
  
if F_index==1  
fit=MAPE(L(1),L(2),L(3));  
end
```

The code is color-coded: comments are green, function names and keywords are blue, and variable names are black. The 'dim' parameter in the function signature is highlighted with a yellow background.

FIGURE 9.1: test_function with 3-dimensional.m

```

%Insert your own boundaries with a new F_index.

dim=3;
if F_index==1
    down=[0 0 0];up=[3 3 1];
    %down=[0,0];up=[2,1];
end

```

FIGURE 9.2: test_functions_range with 3-dimensional.m

```

function MAPE=MAPE(r,q,alpha)
% X0=[0.26 0.73 2.07 7.08 21.22 63.98 202.51]';
% X0=[4115 4387 5510 5177 5527 6997 9321 11149 14812 20808 27115];
X0=[4115 4387 5510 5177]';
% 68608 74360 81910 93234.18 95272.29
% X0=[34.15 12.74 5.77 2.76 1.75 1.23 0.90 0.74]';
% X0=[357.4 164.6 85.1 59.4 40.3 31.0]';%table3 data
%r=1.9233;q=0.6901;% r=0.0228;q=0.0420;n=length(X0);% r=1.8398;
n=length(X0);
e=zeros(n,1);
for i=1:n;
    e(i)=fractionalfactorial(1-r,i-1);
end
M=[X0,-e];
Ar=mm(r,n);
Aq=mm(-q,n);
Z=Mean(alpha,n);
P=inv((-Z*Ar*M)'*((-Z)*Ar*M))*((-Z)*Ar*M)'*Aq(2:n,:)*Ar*X0;
a=P(1,:);
b=P(2,:);
XXr=zeros(n,1);
Xr=Ar*X0;
XXr=zeros(n,1);
Xr=Ar*X0;
XXr(1)=X0(1);
for k=2:n
    s=zeros(k-1,1);
    for i=1:k-1
        s(i)=fractionalfactorial(-q,k-i)*Xr(i);
    end
    S=sum(s);
    XXr(k)=(2*b-2*S-a*Xr(k-1))/(a+2);
end
Br=inv(Ar);
XX0=Br*XXr;
APE=zeros(n-1,1);
for k=2:n
    APE(k-1)=abs(XX0(k)-X0(k))/X0(k);
end
MAPE=1/(n-1)*sum(APE);
% APE=percent(APE);
% MAPE=percent(MAPE);

```

FIGURE 9.3: MAPE in the new grey model.m

Bibliography

- [1] Esmat Rashedi, Hossein Nezamabadi-pour, and Saeid Saryazdi. Gsa: A gravitational search algorithm. *Information Sciences*, 179(13):2232 – 2248, 2009. ISSN 0020-0255. doi: <http://dx.doi.org/10.1016/j.ins.2009.03.004>. URL <http://www.sciencedirect.com/science/article/pii/S0020025509001200>. Special Section on High Order Fuzzy Sets.
- [2] Nazmul Siddique and Hojjat Adeli. Gravitational search algorithm and its variants. *International Journal of Pattern Recognition and Artificial Intelligence*, 30(08):1639001, 2016. doi: 10.1142/S0218001416390018. URL <http://www.worldscientific.com/doi/abs/10.1142/S0218001416390018>.
- [3] Shuhua Mao, Mingyun Gao, Xinpeng Xiao, and Min Zhu. A novel fractional grey system model and its application. *Applied Mathematical Modelling*, 40(7):5063–5076, 2016. doi: <https://doi.org/10.1016/j.apm.2015.12.014>.
- [4] Deng Ju-Long. Control problems of grey systems. *Systems & Control Letters*, 1(5):288 – 294, 1982. ISSN 0167-6911. doi: [http://dx.doi.org/10.1016/S0167-6911\(82\)80025-X](http://dx.doi.org/10.1016/S0167-6911(82)80025-X). URL <http://www.sciencedirect.com/science/article/pii/S016769118280025X>.
- [5] Bao Hua Yang, Jin Shuai Zhao, School Of Business, Jiangsu Normal University, and Jiangsu Normal University. Fractional order discrete grey gm(1, 1) power model and its application. *Control & Decision*, 30:1264 – 1268, 2015. doi: 10.13195/j.kzyjc.2014.0578.
- [6] Ze Zhao, Jianzhou Wang, Jing Zhao, and Zhongyue Su. Using a grey model optimized by differential evolution algorithm to forecast the per capita annual net income of rural households in china. *Omega*, 40(5):525 – 532, 2012. ISSN 0305-0483. doi: <https://doi.org/10.1016/j.omega.2011.10.003>. URL <http://www.sciencedirect.com/science/article/pii/S0305048311001538>.
- [7] Peng-Wen Chen, Wei-Yuan Lin, Tsui-Hua Huang, and Wen-Tsao Pan. Using fruit fly optimization algorithm optimized grey model neural network to perform

- satisfaction analysis for e-business service. *Appl Math Inf Sci*, 7(2):459–465, 2013. URL <http://www.naturalspublishing.com/Article.asp?ArtcID=3236>.
- [8] Li-Chang Hsu. Forecasting the output of integrated circuit industry using genetic algorithm based multivariable grey optimization models. *Expert Systems with Applications*, 36(4):7898 – 7903, 2009. ISSN 0957-4174. doi: <http://dx.doi.org/10.1016/j.eswa.2008.11.004>. URL <http://www.sciencedirect.com/science/article/pii/S0957417408008233>.
- [9] MAO Shuhua, GAO Mingyun, and XIAO Xinping. Fractional order accumulation time-lag gm(1,n, τ) model and its application. *Systems Engineering - Theory & Practice*, 35(2):430, 2015. doi: 10.12011/1000-6788(2015)2-430.
- [10] Xin ping Xiao, Huan Guo, and Shu hua Mao. The modeling mechanism, extension and optimization of grey gm (1, 1) model. *Applied Mathematical Modelling*, 38(5):1896 – 1910, 2014. ISSN 0307-904X. doi: <http://dx.doi.org/10.1016/j.apm.2013.10.004>. URL <http://www.sciencedirect.com/science/article/pii/S0307904X13006161>.
- [11] Saadat Bahrami, Rahmat-Allah Hooshmand, and Moein Parastegari. Short term electric load forecasting by wavelet transform and grey model improved by pso (particle swarm optimization) algorithm. *Energy*, 72:434 – 442, 2014. ISSN 0360-5442. doi: <http://dx.doi.org/10.1016/j.energy.2014.05.065>. URL <http://www.sciencedirect.com/science/article/pii/S036054421400632X>.
- [12] Jianzhou Wang, Suling Zhu, Weigang Zhao, and Wenjin Zhu. Optimal parameters estimation and input subset for grey model based on chaotic particle swarm optimization algorithm. *Expert Systems with Applications*, 38(7):8151 – 8158, 2011. ISSN 0957-4174. doi: <http://dx.doi.org/10.1016/j.eswa.2010.12.158>. URL <http://www.sciencedirect.com/science/article/pii/S0957417410015216>.
- [13] Guo-Dong Li, Shiro Masuda, Daisuke Yamaguchi, and Masatake Nagai. The optimal gnn-pid control system using particle swarm optimization algorithm. *International Journal of Innovative Computing, Information and Control*, 5(10): 3457–3469, 2009.
- [14] Weimin Ma, Xiaoxi Zhu, and Miaomiao Wang. Forecasting iron ore import and consumption of china using grey model optimized by particle swarm optimization algorithm. *Resources Policy*, 38(4):613 – 620, 2013. ISSN 0301-4207. doi: <http://dx.doi.org/10.1016/j.resourpol.2013.09.007>. URL <http://www.sciencedirect.com/science/article/pii/S0301420713000780>.

- [15] Wei Meng and Bo Zeng. Discrete grey model with inverse fractional operators and optimized order. *Control & Decision*, 31(10):1903–1907, 2016. ISSN 1001-0920.
- [16] Ze dong Xu, Fu-Xiang Liu, and Jun Liu. Optimum grey action quantity for gm1,1 model with fractional order accumulate and its research. *Mathematics in Praticce and Theory*, 46(19):235–242, 2016. ISSN 1000-0984.
- [17] Deng Julong. Introduction to grey system theory. *The Journal of grey system*, 1(1):1–24, 1989.
- [18] A Okubo, PK Maini, MH Williamson, and JD Murray. On the spatial spread of the grey squirrel in britain. *Proceedings of the Royal Society of London B: Biological Sciences*, 238(1291):113–125, 1989.
- [19] Xin ping Xiao, Huan Guo, and Shu hua Mao. The modeling mechanism, extension and optimization of grey GM (1, 1) model. *Applied Mathematical Modelling*, 38(56):1896 – 1910, 2014. ISSN 0307-904X. doi: <http://dx.doi.org/10.1016/j.apm.2013.10.004>. URL <http://www.sciencedirect.com/science/article/pii/S0307904X13006161>.
- [20] Chin-Tsai Lin and Shih-Yu Yang. Forecast of the output value of taiwan’s optoelectronics industry using the grey forecasting model. *Technological Forecasting and Social Change*, 70(2):177 – 186, 2003. ISSN 0040-1625. doi: [https://doi.org/10.1016/S0040-1625\(01\)00191-3](https://doi.org/10.1016/S0040-1625(01)00191-3).
- [21] Mingzhi Mao and EC Chirwa. Application of grey model gm (1, 1) to vehicle fatality risk estimation. *Technological Forecasting and Social Change*, 73(5):588–605, 2006.
- [22] Ujjwal Kumar and VK Jain. Time series models (grey-markov, grey model with rolling mechanism and singular spectrum analysis) to forecast energy consumption in india. *Energy*, 35(4):1709–1716, 2010.
- [23] Chin-Tsai Lin and Shih-Yu Yang. Forecast of the output value of taiwan’s optoelectronics industry using the grey forecasting model. *Technological Forecasting and Social Change*, 70(2):177–186, 2003.
- [24] Lifeng Wu, Sifeng Liu, Ligen Yao, Ruiting Xu, and Xunping Lei. Using fractional order accumulation to reduce errors from inverse accumulated generating operator of grey model. *Soft Computing*, 19(2):483–488, 2015. ISSN 1433-7479. doi: [10.1007/s00500-014-1268-y](https://doi.org/10.1007/s00500-014-1268-y). URL <http://dx.doi.org/10.1007/s00500-014-1268-y>.

- [25] Chao-Hung Wang. Predicting tourism demand using fuzzy time series and hybrid grey theory. *Tourism Management*, 25(3):367 – 374, 2004. ISSN 0261-5177. doi: [http://dx.doi.org/10.1016/S0261-5177\(03\)00132-8](http://dx.doi.org/10.1016/S0261-5177(03)00132-8). URL [//www.sciencedirect.com/science/article/pii/S0261517703001328](http://www.sciencedirect.com/science/article/pii/S0261517703001328).
- [26] Lifeng Wu, Sifeng Liu, Ligen Yao, Shuli Yan, and Dinglin Liu. Grey system model with the fractional order accumulation. *Communications in Nonlinear Science and Numerical Simulation*, 18(7):1775 – 1785, 2013. ISSN 1007-5704. doi: <http://dx.doi.org/10.1016/j.cnsns.2012.11.017>. URL [//www.sciencedirect.com/science/article/pii/S1007570412005400](http://www.sciencedirect.com/science/article/pii/S1007570412005400).
- [27] Ju Long Deng. A course on grey system theory. *Huazhong University of Science and Technology Press, Wuhan*, 1990.
- [28] Si-Feng Liu and Ju-long Deng. The range suitable for gm (1, 1)[j]. *Systems Engineering-Theory & Practice*, 5:022, 2000.
- [29] Nai-ming Xie and Si-feng Liu. Discrete gm (1, 1) and mechanism of grey forecasting model. *Systems Engineering-theory & Practice*, 1:93–99, 2005.
- [30] Tianxiang Yao, Jeffery Forrest, and Zaiwu Gong. Generalized discrete gm (1, 1) model. *Grey Systems: Theory and Application*, 2(1):4–12, 2012.
- [31] Nai-ming Xie and Si-Feng Liu. Research on discrete grey model and its predicative result. *XITONG GONGCHENG XUEBAO*, 21(5):520, 2006.
- [32] Nai-ming Xie and Si-feng Liu. Discrete grey forecasting model and its optimization. *Applied Mathematical Modelling*, 33(2):1173–1186, 2009.
- [33] Nai-ming Xie and Si-feng Liu. Research on extension of discrete grey model and its optimize formula. *Systems Engineering-Theory & Practice*, 6:109–112, 2006.
- [34] ZX Wang, YG Dang, and SF Liu. An optimal gm (1, 1) based on the discrete function with exponential law. *Systems Engineering-Theory & Practice*, 2:61–67, 2008.
- [35] Erdal Kayacan, Baris Ulutas, and Okyay Kaynak. Grey system theory-based models in time series prediction. *Expert systems with applications*, 37(2):1784–1789, 2010.
- [36] Fang-Mei Tseng, Hsiao-Cheng Yu, and Gwo-Hsiung Tzeng. Applied hybrid grey model to forecast seasonal time series. *Technological Forecasting and Social Change*, 67(2):291–302, 2001.

- [37] Li-Chang Hsu and Chao-Hung Wang. Forecasting integrated circuit output using multivariate grey model and grey relational analysis. *Expert Systems with Applications*, 36(2, Part 1):1403 – 1409, 2009. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2007.11.015>.
- [38] Wann-Yih Wu and Shuo-Pei Chen. A prediction method using the grey model gmc(1,n) combined with the grey relational analysis: a case study on internet access population forecast. *Applied Mathematics and Computation*, 169(1):198 – 217, 2005. ISSN 0096-3003. doi: <http://dx.doi.org/10.1016/j.amc.2004.10.087>. URL <http://www.sciencedirect.com/science/article/pii/S0096300304007702>.
- [39] Wei-Zhong Lin, Jian-An Fang, Xuan Xiao, and Kuo-Chen Chou. idna-prot: Identification of dna binding proteins using random forest with grey model. *PLOS ONE*, 6(9):1–7, 09 2011. doi: 10.1371/journal.pone.0024756. URL <https://doi.org/10.1371/journal.pone.0024756>.
- [40] M.H. Wang and C.P. Hung. Novel grey model for the prediction of trend of dissolved gases in oil-filled power apparatus. *Electric Power Systems Research*, 67(1):53 – 58, 2003. ISSN 0378-7796. doi: [http://dx.doi.org/10.1016/S0378-7796\(03\)00047-6](http://dx.doi.org/10.1016/S0378-7796(03)00047-6). URL <http://www.sciencedirect.com/science/article/pii/S0378779603000476>.
- [41] Xiao Yi Che, You Xin Luo, and Zhe Ming He. Grey new information gom(1,1) model based on opposite-direction accumulated generating and its application. In *Mechanical Automation and Materials Engineering*, volume 364 of *Applied Mechanics and Materials*, pages 207–210. Trans Tech Publications, 10 2013. doi: 10.4028/www.scientific.net/AMM.364.207.
- [42] Y. Wang, Y. Dang, and X. Pu. Improved unequal interval grey model and its applications. *Journal of Systems Engineering and Electronics*, 22(3):445–451, June 2011. doi: 10.3969/j.issn.1004-4132.2011.03.012.
- [43] Z.-C. Lin and W.-S. Lin. The application of grey theory to the prediction of measurement points for circularity geometric tolerance. *The International Journal of Advanced Manufacturing Technology*, 17(5):348–360, Feb 2001. ISSN 1433-3015. doi: 10.1007/s001700170170. URL <https://doi.org/10.1007/s001700170170>.
- [44] Chun-I Chen. Application of the novel nonlinear grey bernoulli model for forecasting unemployment rate. *Chaos, Solitons & Fractals*, 37(1):278 – 287, 2008. ISSN 0960-0779. doi: <http://dx.doi.org/10.1016/j.chaos.2006.08.024>. URL <http://www.sciencedirect.com/science/article/pii/S0960077906008654>.

- [45] Xiaojun Guo, Sifeng Liu, Lifeng Wu, Yanbo Gao, and Yingjie Yang. A multi-variable grey model with a self-memory component and its application on engineering prediction. *Engineering Applications of Artificial Intelligence*, 42:82 – 93, 2015. ISSN 0952-1976. doi: <http://dx.doi.org/10.1016/j.engappai.2015.03.014>. URL <http://www.sciencedirect.com/science/article/pii/S0952197615000792>.
- [46] Fang-Mei Tseng, Hsiao-Cheng Yu, and Gwo-Hsiung Tzeng. Applied hybrid grey model to forecast seasonal time series. *Technological Forecasting and Social Change*, 67(2):291 – 302, 2001. ISSN 0040-1625. doi: [http://dx.doi.org/10.1016/S0040-1625\(99\)00098-0](http://dx.doi.org/10.1016/S0040-1625(99)00098-0). URL <http://www.sciencedirect.com/science/article/pii/S0040162599000980>. Technology Policy and Innovation in the Globalized Learning Society.
- [47] Lifeng Wu, Sifeng Liu, Ligen Yao, Shuli Yan, and Dinglin Liu. Grey system model with the fractional order accumulation. *Communications in Nonlinear Science and Numerical Simulation*, 18(7):1775 – 1785, 2013. doi: <http://dx.doi.org/10.1016/j.cnsns.2012.11.017>.
- [48] Li-Feng Wu, Si-Feng Liu, Wei Cui, Ding-Lin Liu, and Tian-Xiang Yao. Non-homogenous discrete grey model with fractional-order accumulation. *Neural Computing and Applications*, 25(5):1215–1221, 2014. ISSN 1433-3058. doi: 10.1007/s00521-014-1605-1.
- [49] Lifeng Wu. Using fractional gm (1, 1) model to predict the life of complex equipment. *Grey Systems: Theory and Application*, 6(1):32–40, 2016. doi: <http://dx.doi.org/10.1108/GS-07-2015-0034>.
- [50] Lifeng Wu, Sifeng Liu, Ligen Yao, Ruiting Xu, and Xunping Lei. Using fractional order accumulation to reduce errors from inverse accumulated generating operator of grey model. *Soft Computing*, 19(2):483–488, 2015. doi: 10.1007/s00500-014-1268-y.
- [51] Li-feng Wu, Si-feng Liu, and Ligen Yao. Grey model with caputo fractional order derivative. *Systems Engineering–Theory & Practice*, 35(5):1311–1316, 2015. doi: 10.12011/1000-6788(2015)5-1311.
- [52] F. Shili, W. Lifeng, Y. Liang, and F. Zhigeng. Using fractional gm(1,1) model to predict maintenance cost of weapon system. In *Proceedings of 2013 IEEE International Conference on Grey systems and Intelligent Services (GSIS)*, pages 177–181, Nov 2013. doi: 10.1109/GSIS.2013.6714760.

- [53] Igor Podlubny. *Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, volume 198. Academic press, 1998.
- [54] Kai Diethelm and Neville J Ford. Analysis of fractional differential equations. *Journal of Mathematical Analysis and Applications*, 265(2):229–248, 2002.
- [55] Jiefang Liu, Sifeng Liu, and Zhigeng Fang. Fractional-order reverse accumulation generation gm (1, 1) model and its applications. *The Journal of Grey System*, 27(4):52–63, 2015.
- [56] F. Shili, W. Lifeng, Y. Liang, and F. Zhigeng. Using fractional gm(1,1) model to predict maintenance cost of weapon system. In *Proceedings of 2013 IEEE International Conference on Grey systems and Intelligent Services (GSIS)*, pages 177–181, Nov 2013. doi: 10.1109/GSIS.2013.6714760.
- [57] Che-Chiang Hsu and Chia-Yon Chen. Applications of improved grey prediction model for power demand forecasting. *Energy Conversion and Management*, 44(14):2241 – 2249, 2003. ISSN 0196-8904. doi: [http://dx.doi.org/10.1016/S0196-8904\(02\)00248-0](http://dx.doi.org/10.1016/S0196-8904(02)00248-0). URL <http://www.sciencedirect.com/science/article/pii/S0196890402002480>.
- [58] Ujjwal Kumar and V.K. Jain. Time series models (grey-markov, grey model with rolling mechanism and singular spectrum analysis) to forecast energy consumption in india. *Energy*, 35(4):1709 – 1716, 2010. ISSN 0360-5442. doi: <http://dx.doi.org/10.1016/j.energy.2009.12.021>. URL <http://www.sciencedirect.com/science/article/pii/S0360544209005416>. Demand Response Resources: the US and International Experience.
- [59] Li-Chang Hsu and Chao-Hung Wang. Forecasting the output of integrated circuit industry using a grey model improved by the bayesian analysis. *Technological Forecasting and Social Change*, 74(6):843 – 853, 2007. ISSN 0040-1625. doi: <http://dx.doi.org/10.1016/j.techfore.2006.02.005>. URL <http://www.sciencedirect.com/science/article/pii/S0040162506000515>.
- [60] P. Zhou, B.W. Ang, and K.L. Poh. A trigonometric grey prediction approach to forecasting electricity demand. *Energy*, 31(14):2839 – 2847, 2006. ISSN 0360-5442. doi: <http://dx.doi.org/10.1016/j.energy.2005.12.002>. URL <http://www.sciencedirect.com/science/article/pii/S0360544205002793>.
- [61] Yi-Shian Lee and Lee-Ing Tong. Forecasting energy consumption using a grey model improved by incorporating genetic programming. *Energy Conversion and Management*, 52(1):147 – 152, 2011. ISSN 0196-8904. doi: <http://dx.doi.org/10.1016/j.energy.2010.12.002>.

- 1016/j.enconman.2010.06.053. URL <http://www.sciencedirect.com/science/article/pii/S0196890410002748>.
- [62] Chris Tofallis. A better measure of relative prediction accuracy for model selection and model estimation. *Journal of the Operational Research Society*, 66(8):1352–1362, Aug 2015. ISSN 1476-9360. doi: 10.1057/jors.2014.103. URL <https://doi.org/10.1057/jors.2014.103>.
- [63] R. Eberhart and J. Kennedy. A new optimizer using particle swarm theory. In *Micro Machine and Human Science, 1995. MHS '95., Proceedings of the Sixth International Symposium on*, pages 39–43, Oct 1995. doi: 10.1109/MHS.1995.494215.
- [64] Y. Shi and R. Eberhart. A modified particle swarm optimizer. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 69–73, May 1998. doi: 10.1109/ICEC.1998.699146.
- [65] R. Mendes, J. Kennedy, and J. Neves. The fully informed particle swarm: simpler, maybe better. *IEEE Transactions on Evolutionary Computation*, 8(3):204–210, June 2004. ISSN 1089-778X. doi: 10.1109/TEVC.2004.826074.
- [66] P. J. Angeline. Using selection to improve particle swarm optimization. In *1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence (Cat. No.98TH8360)*, pages 84–89, May 1998. doi: 10.1109/ICEC.1998.699327.
- [67] N. Higashi and H. Iba. Particle swarm optimization with gaussian mutation. In *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, pages 72–79, April 2003. doi: 10.1109/SIS.2003.1202250.
- [68] S. Baskar and P. N. Suganthan. A novel concurrent particle swarm optimization. In *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, volume 1, pages 792–796 Vol.1, June 2004. doi: 10.1109/CEC.2004.1330940.
- [69] B. Al-kazemi and C. K. Mohan. Multi-phase generalization of the particle swarm optimization algorithm. In *Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on*, volume 1, pages 489–494, May 2002. doi: 10.1109/CEC.2002.1006283.
- [70] N. El Hami and M. Bouchekourte. Optimising liquidity with modified particle swarm optimization application: Case of casablanca stock exchange. In *2016 4th*

- IEEE International Colloquium on Information Science and Technology (CiSt)*, pages 725–729, Oct 2016. doi: 10.1109/CIST.2016.7804981.
- [71] Weidong Liu, Zhaowen Luan, Yang Yang, Ruiyan Gan, and Hui Zhao. The application of the improved particle swarm optimization on dynamic economic dispatch of power system with wind farms. In *TENCON 2015 - 2015 IEEE Region 10 Conference*, pages 1–6, Nov 2015. doi: 10.1109/TENCON.2015.7372978.
- [72] K. Singh and G. Shankar. Pid parameters tuning using modified particle swarm optimization and its application in load frequency control. In *2016 IEEE 6th International Conference on Power Systems (ICPS)*, pages 1–6, March 2016. doi: 10.1109/ICPES.2016.7584160.
- [73] Zong Woo Geem, Joong Hoon Kim, and G.V. Loganathan. A new heuristic optimization algorithm: Harmony search. *SIMULATION*, 76(2):60–68, 2001. doi: 10.1177/003754970107600201. URL <http://dx.doi.org/10.1177/003754970107600201>.
- [74] Ioan Cristian Trelea. The particle swarm optimization algorithm: convergence analysis and parameter selection. *Information Processing Letters*, 85(6):317 – 325, 2003. ISSN 0020-0190. doi: [http://dx.doi.org/10.1016/S0020-0190\(02\)00447-7](http://dx.doi.org/10.1016/S0020-0190(02)00447-7). URL <http://www.sciencedirect.com/science/article/pii/S0020019002004477>.
- [75] Martin Pelikan, David E. Goldberg, and Erick Cantú-Paz. Boa: The bayesian optimization algorithm. In *Proceedings of the 1st Annual Conference on Genetic and Evolutionary Computation - Volume 1, GECCO'99*, pages 525–532, San Francisco, CA, USA, 1999. Morgan Kaufmann Publishers Inc. ISBN 1-55860-611-4. URL <http://dl.acm.org/citation.cfm?id=2933923.2933973>.
- [76] Dervis Karaboga and Bahriye Basturk. *Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems*, pages 789–798. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007. ISBN 978-3-540-72950-1. doi: 10.1007/978-3-540-72950-1_77. URL http://dx.doi.org/10.1007/978-3-540-72950-1_77.
- [77] R. S. Parpinelli, H. S. Lopes, and A. A. Freitas. Data mining with an ant colony optimization algorithm. *IEEE Transactions on Evolutionary Computation*, 6(4): 321–332, Aug 2002. ISSN 1089-778X. doi: 10.1109/TEVC.2002.802452.
- [78] J. J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):122–128, Jan 1986. ISSN 0018-9472. doi: 10.1109/TSMC.1986.289288.

- [79] Esmat Rashedi, Hossein Nezamabadi-pour, and Saeid Saryazdi. Bgsa: binary gravitational search algorithm. *Natural Computing*, 9(3):727–745, Sep 2010. ISSN 1572-9796. doi: 10.1007/s11047-009-9175-3. URL <https://doi.org/10.1007/s11047-009-9175-3>.
- [80] J. P. Papa, A. Pagnin, S. A. Schellini, A. Spadotto, R. C. Guido, M. Ponti, G. Chiacchia, and A. X. Falco. Feature selection through gravitational search algorithm. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2052–2055, May 2011. doi: 10.1109/ICASSP.2011.5946916.
- [81] R. E. Precup, R. C. David, E. M. Petriu, S. Preitl, and M. B. Radac. Novel adaptive gravitational search algorithm for fuzzy controlled servo systems. *IEEE Transactions on Industrial Informatics*, 8(4):791–800, Nov 2012. ISSN 1551-3203. doi: 10.1109/TII.2012.2205393.
- [82] R. E. Precup, R. C. David, E. M. Petriu, M. B. Radac, and S. Preitl. Adaptive gsa-based optimal tuning of pi controlled servo systems with reduced process parametric sensitivity, robust stability and controller robustness. *IEEE Transactions on Cybernetics*, 44(11):1997–2009, Nov 2014. ISSN 2168-2267. doi: 10.1109/TCYB.2014.2307257.
- [83] S. Mirjalili and S. Z. M. Hashim. A new hybrid psogsa algorithm for function optimization. In *2010 International Conference on Computer and Information Application*, pages 374–377, Dec 2010. doi: 10.1109/ICCIA.2010.6141614.
- [84] S. N. Shahab, A. R. Zainun, N. H. Noordin, I. I. Mohamed, and W. D. Abdullah. Null steering optimization based mvdr beamformer using hybrid psogsa approach for antenna array system. In *2016 IEEE Student Conference on Research and Development (SCOReD)*, pages 1–6, Dec 2016. doi: 10.1109/SCORED.2016.7810078.
- [85] S. Alkhalaf, O. Alfarraj, and A. M. Hemeida. Fuzzy-vq image compression based hybrid psogsa optimization algorithm. In *2015 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, pages 1–6, Aug 2015. doi: 10.1109/FUZZ-IEEE.2015.7337998.
- [86] E. Z. Zhang, Y. F. Wu, and Q. W. Chen. A hybrid multi-objective psogsa approach for environmental/economic dispatch. In *2015 IEEE 10th Conference on Industrial Electronics and Applications (ICIEA)*, pages 455–460, June 2015. doi: 10.1109/ICIEA.2015.7334156.

- [87] J. Radosavljevi, N. Arsi, and M. Jevti. Optimal power flow using hybrid psogsa algorithm. In *2014 55th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON)*, pages 136–140, Oct 2014. doi: 10.1109/RTUCON.2014.6998173.
- [88] P. Jiang, Z. Zeng, J. Chen, and H. Tang. A psogsa method to optimize the t-s fuzzy neural network for displacement prediction of landslide. In *2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1216–1221, Oct 2014. doi: 10.1109/SMC.2014.6974080.
- [89] Chaoshun Li, Jianzhong Zhou, Jian Xiao, and Han Xiao. Hydraulic turbine governing system identification using ts fuzzy model optimized by chaotic gravitational search algorithm. *Engineering Applications of Artificial Intelligence*, 26(9):2073 – 2082, 2013. ISSN 0952-1976. doi: <http://dx.doi.org/10.1016/j.engappai.2013.04.002>. URL <http://www.sciencedirect.com/science/article/pii/S0952197613000596>.
- [90] Mohammad Bagher Dowlatshahi, Hossein Nezamabadi-pour, and Mashaallah Mashinchi. A discrete gravitational search algorithm for solving combinatorial optimization problems. *Information Sciences*, 258:94 – 107, 2014. ISSN 0020-0255. doi: <http://dx.doi.org/10.1016/j.ins.2013.09.034>. URL <http://www.sciencedirect.com/science/article/pii/S0020025513006762>.
- [91] A. Chatterjee, S.P. Ghoshal, and V. Mukherjee. A maiden application of gravitational search algorithm with wavelet mutation for the solution of economic load dispatch problems. *International Journal of Bio-Inspired Computation*, 4(1):33–46, 2012. doi: 10.1504/IJBIC.2012.044934. URL <http://www.inderscienceonline.com/doi/abs/10.1504/IJBIC.2012.044934>. PMID: 44934.
- [92] Ali Ghasemi, Hossein Shayeghi, and Hasan Alkhatib. Robust design of multi-machine power system stabilizers using fuzzy gravitational search algorithm. *International Journal of Electrical Power & Energy Systems*, 51:190 – 200, 2013. ISSN 0142-0615. doi: <http://dx.doi.org/10.1016/j.ijepes.2013.02.022>. URL <http://www.sciencedirect.com/science/article/pii/S0142061513000823>.
- [93] L. Wei and B. Ma. Application of improved gravitational search algorithm in pid control for boiler drum water level. In *2017 29th Chinese Control And Decision Conference (CCDC)*, pages 1852–1857, May 2017. doi: 10.1109/CCDC.2017.7978818.
- [94] Jing-zhi LIU and Wei SUN. Zhang shao-juan school of information and electrical engineering, china university of mining and technology, xuzhou, jiangsu 221008,

- china; particle swarm optimization-based pid control of drum water level. *Computer Engineering and Applications*, 21, 2009.
- [95] H. R. Hassanzadeh and M. Rouhani. A multi-objective gravitational search algorithm. In *2010 2nd International Conference on Computational Intelligence, Communication Systems and Networks*, pages 7–12, July 2010. doi: 10.1109/CICSyN.2010.32.
- [96] S. Mostaghim and J. Teich. Strategies for finding good local guides in multi-objective particle swarm optimization (mopso). In *Swarm Intelligence Symposium, 2003. SIS '03. Proceedings of the 2003 IEEE*, pages 26–33, April 2003. doi: 10.1109/SIS.2003.1202243.
- [97] T. Wang, X. Wei, Q. Sun, and F. Hu. Gsa-based jammer localization in multi-hop wireless network. In *2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC)*, volume 1, pages 410–415, July 2017. doi: 10.1109/CSE-EUC.2017.80.
- [98] S. K. Mohapatra, B. S. Dash, and P. M. Dash. Application of gsa optimized pi controller parameters in automatic generation control for interconnected power system. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPEs)*, pages 1522–1527, Oct 2016. doi: 10.1109/SCOPEs.2016.7955694.
- [99] A. Sharma and S. Mathur. Deterministic maximum likelihood direction of arrival estimation using gsa. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 415–419, March 2016. doi: 10.1109/ICEEOT.2016.7755482.
- [100] M. S. Nobarian, R. M. Asl, M. Nemati, and F. Hashemzadeh. Optimal l1 control for linear time-delayed systems using gsa algorithm. In *2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA)*, pages 111–115, Jan 2016. doi: 10.1109/ICCIAutom.2016.7483145.
- [101] F. Saadat and M. Nasri. A gsa-based method in human identification using finger vein patterns. In *2016 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC)*, pages 102–106, March 2016. doi: 10.1109/CSIEC.2016.7482109.
- [102] Yun F. Li. The application of matlab software on the college network teaching platform of the mathematics experiment. *Applied Mechanics and Materials*, 55-57:1459, 05 2011. URL <https://search.proquest.com/docview/1443591048?>

- [accountid=130472](#). Copyright Trans Tech Publications Ltd. May 2011;- 2013-10-22.
- [103] M. Segou and N. Voulgaris. Proschema: A matlab application for processing strong motion records and estimating earthquake engineering parameters. *Computers & Geosciences*, 36(7):977 – 986, 2010. ISSN 0098-3004. doi: <http://dx.doi.org/10.1016/j.cageo.2009.09.020>. URL [//www.sciencedirect.com/science/article/pii/S0098300410000798](http://www.sciencedirect.com/science/article/pii/S0098300410000798).
- [104] Yang Yang and Dingy Xue. Continuous fractional-order grey model and electricity prediction research based on the observation error feedback. *Energy*, 115, Part 1:722 – 733, 2016. ISSN 0360-5442. doi: <http://dx.doi.org/10.1016/j.energy.2016.08.097>. URL [//www.sciencedirect.com/science/article/pii/S0360544216312105](http://www.sciencedirect.com/science/article/pii/S0360544216312105).
- [105] JetChau Wen, KuoHsun Huang, and KunLi Wen. The study of in gm(1,1) model. *Journal of the Chinese Institute of Engineers*, 23(5):583–589, 2000. doi: [10.1080/02533839.2000.9670579](http://dx.doi.org/10.1080/02533839.2000.9670579). URL <http://dx.doi.org/10.1080/02533839.2000.9670579>.