Soft Comput
DOI 10.1007/s00500-015-1723-4

METHODOLOGIES AND APPLICATION

Distilling middle-age cement hydration kinetics from observed data using phased hybrid evolution

Lin Wang[1] · **Bo Yang**[1,2] · **Ajith Abraham**[3]

Abstract The hydration of cement is of great importance to the formation of the microstructure and development of strength. However, the complex nature of cement hydration results in that the existing manually derived model from some assumptions that are made to simplify the problem and make it mathematically and computationally tractable is not satisfactory in comparison with experimental results. In this paper, the middle-age hydration kinetics is distilled from observed data reversely using phased hybrid evolution method. The task that distils the hydration kinetics is divided into two phases and combines different algorithms. Furthermore, some strategies are also adopted for enhancement. To solve the problem of high time complexity, the searching process is accelerated by graphics processing units in parallel. The middle-age cement hydration kinetics model is distilled successfully from observed data. Studies show that the simulation result is close to the same with experimental results according to the distilled kinetics.

Keywords Evolutionary computation · Cement hydration kinetics · Reverse modeling

Communicated by V. Loia.

✉ Bo Yang
wangplanet@gmail.com; yangbo@ujn.edu.cn

[1] Shandong Provincial Key Laboratory of Network based Intelligent Computing, University of Jinan, Jinan 250022, China

[2] School of Informatics, Linyi University, Linyi 276000, China

[3] Machine Intelligence Research Labs (MIR Labs), Scientific Network for Innovation and Research Excellence, Auburn, USA

1 Introduction

Modeling and simulation enable the scientists to compare observations with theory, extract physical parameters from experimental data, and predict the behaviors of system (Thomas et al. 2011). The study of cement hydration modeling and simulation is of profound theoretical and practical significance and range from simple single particle models to complex hydration simulation models, from macro-scale to micro-scale, from short-term simulation to long-term prediction. The simple models gain insight about the major controlling processes at different periods of hydration on the lower stage. Complex models can also be developed for simulation by incorporating as much of the known physics and chemistry as possible to predict complicated phenomena on the advanced stage (Thomas et al. 2011).

The hydration of cement in middle-age is closely related to the formation of microstructure, development of strength, and the final performance of hardened cement. Although the cement hydration is a long process, the middle-age hydration kinetics covers all five periods, including dissolution, induction, acceleration, deceleration, and termination. Therefore, it is of practical significance to research the middle-age hydration kinetics.

Currently, the cement hydration models are built based on first principle method and with some assumptions that are made to simplify the problem and make it mathematically and computationally tractable. However, due to the inhomogeneity and heterogeneity of cement, complex interrelated chemical reactions and physical changes take place during the process of hydration. The manually derived models from assumptions are not satisfactory in comparison with the observed experimental results. The manually derived models cannot represent the cement hydration process as a system completely and accurately.

Published online: 23 May 2015

With the popularization and improvement of experimental devices and approaches, more data can be acquired conveniently. At the same time, with the advancement in computer technology, many high-performance computers have emerged. The computers' peak performances increase rapidly. Furthermore, recent studies have shown that evolutionary computation (Koza 1992; Holland 1975) has the ability to discover natural law and formula from observed experimental data. However, the cement hydration modeling benefits few from all of these advances. This paper explores distilling the middle-age hydration kinetics unified form, extracting relationships between kinetic coefficients and cement parameters from observed experimental data using a phased hybrid evolution method.

This paper makes the following major contributions:

- First time that the middle-age cement hydration kinetics is distilled reversely from observed data using evolutionary computation.
- Also the first time that the relationships between kinetics coefficients and cement hydration parameters are extracted reversely and analyzed.
- Propose a phased hybrid evolution method with some strategies to increase the diversity of searching for kinetics evolution and accelerate it using graphics processing units (GPUs).

2 Related works

2.1 Cement hydration kinetics

Modeling and simulation of cement hydration are effective tools to reveal the principle of cement hydration, to predict the hydration process, to study the relationship between microstructure and properties, and to improve the design of high-performance cement material. The researchers build models and simulate hydration for cement using many kinds of methods. One of the most commonly used is studying hydration kinetics equation (Thomas et al. 2011; Tomosawa 1997; Krstulovic and Dabic 2000; Dabic et al. 2000; Lin and Meyer 2009). The degree of hydration α refers to the weight fraction of cement reacted in a specified time (Fig. 1). "Mechanical and transport properties of a hardened cement paste are significantly influenced by its degree of hydration, so that computer models having an objective of predicting performance properties must provide an accurate portrayal of hydration kinetics, whether from first principle calculations, empirical relationships, or calibration versus experimental data" (Bentz 2011). The hydration kinetics equation reveals the evolution of the degree of hydration α with cement hydration over time t and is often represented by differential equation. We can also build models of microstructure (Pignat et al. 2005) and performance indicators (Park et al. 2005) based on hydration kinetics.

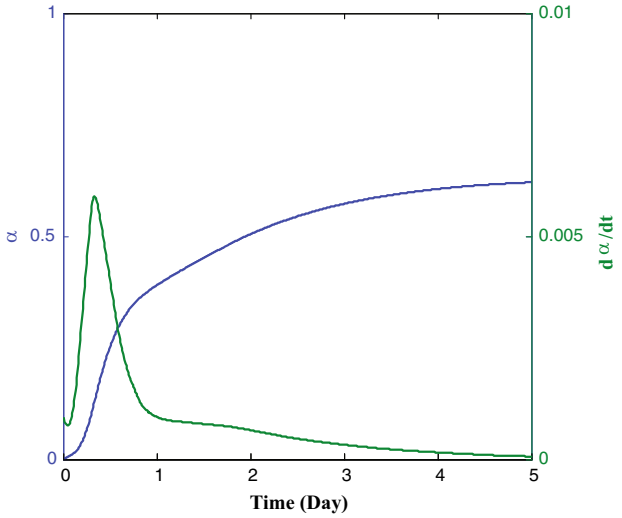

Fig. 1 The time series of degree of hydration and hydration rate

2.2 Reverse engineering using evolutionary computation

In recent years, it has been found that evolutionary computation enables us to discover nature law and formula from observed experimental data. Schmidt and Lipson (2009) searched motion-tracking data captured from various physical systems. They confirm that genetic programming is fully capable of discovering Hamiltonians, Lagrangians, and other laws of geometric and momentum conservation without the help from any prior knowledge about physics, kinematics, or geometry. Fan et al. (2004) proposed a new systematic method that can automatically generate ranking strategies for different contexts based on Genetic Programming. Floares (2007) proposed a reverse engineering algorithm for (drug) gene regulating networks, based on genetic programming-RODES, which automatically discovers the structure, estimates the parameters, and identifies the molecular mechanisms involved. Sanz et al. (2007) applied two evolutionary programming algorithms to design prototype lowpass Finite Impulse Response filters for use in a modulated filterbank. Huang et al. (2012) presented an ant colony optimization-based algorithm to support end users in making sensible resource allocations and made it possible to search an optimal task operation path on the generated task operation model. Qian et al. (2008) inferred noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering. Yang et al. (2012) propose an evolutionary approach to parameter identification for thermal models that are formulated as an optimization task. Krogmann et al. (2010) presents a novel comprehensive approach for reverse engineering and performance prediction

of components by utilizing genetic programming for reconstructing a behavior model from monitoring data, runtime byte code counts, and static bytecode analysis.

In previous research, the authors also provided preliminary attempts (Wang et al. 2010, 2012a, b) to extract the early-age hydration kinetics from the early 24 h data. However, the cement hydration is a long process and still continues after 24 h. Furthermore, the hydration process after 24 h also plays an important role on the growth of strength. The early-age hydration kinetics is incomplete and insufficient. In addition, previous studies also did not get the functional relationship between kinetics coefficients and cement hydration parameters, e.g., the chemical composition, physical parameters, and curing conditions. Therefore, it can help to neither understand the hydration process, nor forecast hydration directly.

2.3 GPU computing

The time consumption of evolutionary computation in solving complex problems is very high. The development of GPU brings dawn and hope to solve this problem using a low-cost platform. The GPU computing is quickly becoming as common as CPU computing with the help of Compute Unified Device Architecture (CUDA) (CUDA 2009), which is a development platform and considers the GPU as a parallel data computing device. CUDA enables the allocation and management of computing tasks. Even an ordinary GPU in a system can speedup programs with the help of CUDA.

The high computational complexity faced by the evolutionary computation and the progress achieved by the GPU computing motivate many researchers to explore speeding up evolutionary computation algorithms using CUDA. Luo et al. (2014) extended the Bees Algorithm to be run on the CUDA. Zhang and He (2009) presented a hierarchical parallel genetic algorithm, implemented by CUDA. Mixed with master–slave parallelization method and multiple-demes parallelization method, this method contributes to better utilization of threads and highspeed shared memory in CUDA. Zhou and Tan (2009) presented a parallel approach to run standard particle swarm optimization on GPU. Robilliard et al. (2009) proposed a method to accelerate genetic programming on GPU. Their work focuses on the possibilities offered by Nvidia G80 GPUs when programmed in the CUDA language.

2.4 Challenge

Due to the inhomogeneity and heterogeneity, there are complex interrelated chemical reactions and physical changes in cement hydration. "A critical component of any modeling or simulation effort is the choice of assumptions used to simplify the problem and make it mathematically and computationally tractable. These assumptions endow each particular model with its own set of strengths and limitations that should be acknowledged and considered when applying the model to a particular case" (Thomas et al. 2011).

The manually derived equation from some assumptions is not satisfactory in comparison with observed experimental results; for example, the model developed by Kondo and Kodama (1967) that characterized the hydration kinetics of alite, which is the chief component of Portland cement, using an assumption of concentric-layered growth of hydrates of uniform thickness on a single reacting spherical cement particle (Thomas et al. 2011). Another example is the JMAK model (Avrami 1939; Johnson and Mehl 1939), which has been used for many years. "The serious mismatch between its assumption of random volume nucleation and the observation that C–S–H nucleates on mineral surfaces means that the parameters from such fits will have little or no physical meaning" (Thomas et al. 2011).

Furthermore, the definite functions between model coefficients and cement hydration parameters are still unknown in manually derived kinetics. The coefficients in these models were determined from fits to experimental data on the rate of heat evolution obtained using microcalorimetry. This disadvantage resulted in these models cannot be used directly in forecasting of hydration or in the formulation of different systems.

Therefore, with above-mentioned limitations, the manually derived models cannot fully represent the cement hydration process as a system that is complete and accurate. The difficulties faced by manual derivation of kinetic equation and the progress achieved by evolutionary computation in natural law discovery lead the current research to explore inference of cement hydration kinetics using evolutionary computation methods from observed data. "We can analyze the data without hypotheses about what it might show. We can throw the numbers into the biggest computing clusters the world has ever seen and let statistical algorithms find patterns where science cannot." stated by Anderson (2008).

3 Methodology

The unified form of kinetic equation is required by systems of cement hydration. Moreover, for different systems of cement hydration, the difference of them is reflected in the difference of value of coefficients. The cement hydration kinetics is expressed through a first-order differential equation as follows

$$\frac{\mathrm{d}\alpha}{\mathrm{d}t} = f_{C_0, C_1, \ldots, C_{n-1}}(t, \alpha) \tag{1}$$

where f represents the unified form of kinetic equation, and $C_0, C_1, \ldots, C_{n-1}$ represent coefficients of the kinetic

equation. To adapt to different systems, the value of coefficients should be changed correspondingly when the curing condition or the type of cement is changed. That is to say, the coefficients are endowed with ability to adjust kinetics for various systems of cement hydration. The coefficients in different models represent the different physical meanings. For example, in the Tomasawa's model (Tomosawa 1997), the coefficients consist of a reaction rate per unit area of reaction surface, radius of unhydrated cement particle, etc. Conversely, in the Krstulovic–Dabic's model (Krstulovic and Dabic 2000), the coefficients reflect the rate of different subprocesses in hydration. The cement hydration parameters determine the value of the coefficients as well as the behaviors the system exhibits. Therefore, the mathematical relationships between coefficients and hydration parameters have to be determined as follows to forecast hydration process and to explain the cement hydration.

$$\begin{cases} C_0 = g_0\,(F_0, F_1, \ldots, F_{m-1}) \\ C_1 = g_1\,(F_0, F_1, \ldots, F_{m-1}) \\ \cdots\cdots \\ C_{n-1} = g_{n-1}\,(F_0, F_1, \ldots, F_{m-1}) \end{cases} \tag{2}$$

where $F_0, F_1, \ldots, F_{m-1}$ represent parameters of the cement hydration system, e.g., water-to-cement ratio (w/c), temperature, weight percentage content of tricalcium silicate, etc. $g_0, g_1, \ldots, g_{n-1}$ represent the functional relationship from parameters space to each coefficient, i.e., the coefficient function.

In designing the method for evolving hydration kinetics, some points should be taken into consideration first. There are $n+1$ functions that should be optimized in this task. The unified form of kinetic equation f and the functional relationship $g_0, g_1, \ldots, g_{n-1}$ of its coefficients cannot evolve simultaneously on account of unified form, complexity, and data. First, most of the researches of cement hydration modeling focus on deriving unified form of kinetics f that is satisfactory in comparison with the experimental results of hydration. However, such unified form that fits the curve of hydration satisfactorily has not been discovered up to now. Second, for evolving the expression of unified form of kinetics f and expressions of coefficients $g_0, g_1, \ldots, g_{n-1}$ simultaneously, the searching space will extend, which leads to the increase of complexity of evolution. Finally, owing to the preparation of samples, the slowness of cement hydration, the guarantee of quality of data, and the restriction of available machine time, we were unable to collect a large amount of data. Therefore, it is hard to find an accurate model mixed with hydration parameters based on small data. However, the collected data are enough for getting a unified form of kinetics that fits observed results satisfactorily by adjusting coefficients. This phenomenon can be observed from the experimental section.

The task that distils the hydration kinetics reversely from observed data is divided into two phases and combines different evolution algorithms. Furthermore, the hybrid evolution with some enhancement strategies are also adopted. At first, the focus is on distilling the unified form of kinetic equation using Gene Expression Programming (GEP) and Particle Swarm Optimization (PSO) to infer the best form f. This hybrid method further combines with some enhancement strategies and is accelerated by GPUs in parallel. At the output of the first phase, the best expression only represents the unified form of equation, where the unified form of kinetics and the position of coefficients are fixed, regardless of the value of coefficients or the type of cement. Second, investigate the mathematical relationship $g_0, g_1, \ldots, g_{n-1}$ between coefficients and hydration parameters using GEP algorithm. The objective is to infer the function for every coefficient in the evolved unified equation form.

Figure 2 shows the main flowchart of the proposed approach. The thermal signal of heat releases in the hydration process over time is determined by chemical experiment using microcalorimetry and is further transformed to the corresponding time series of degree of hydration. The hybrid evolutionary computation, involving GEP, PSO, and some reinforcement strategies evolves the unified form of equation of kinetics by taking collected time series as training data in the environment of the GPU-based high-performance computer in the first phase. The next phase is to analyze the relationship between kinetics coefficients and hydration parameters. The PSO and GEP are used to calculate the accurate value of coefficients and to extract the functional relationship between coefficients and parameters, i.e., the coefficient function, respectively. After finishing both of the phases, the cement hydration kinetics will be distilled.

3.1 Collect data and estimate degree of hydration

Both of direct determination and indirect determination can be adopted to estimate degree of hydration. The direct determination, which comprises petrographic analysis, X-ray analysis, etc., spends too much time to determine the data in real time. Therefore, the indirect determination is adopted in this research. It comprises the microcalorimetry, quantitative analysis of calcium hydroxide, and bound water determination, etc. Considering the damaging effect made by the latter two approaches disables the continuous determination for the same sample, the microcalorimetry (Wang et al. 2005) is selected in this research to estimate the degree of hydration (Wang et al. 2010).

A pack of cement powders will turn into paste after being mixed with water. Then, it will set in a few hours. During this process, the hydration will take place and release heat. The release of heat is closely bound up with the development of hydration. Therefore, this phenomenon enables us

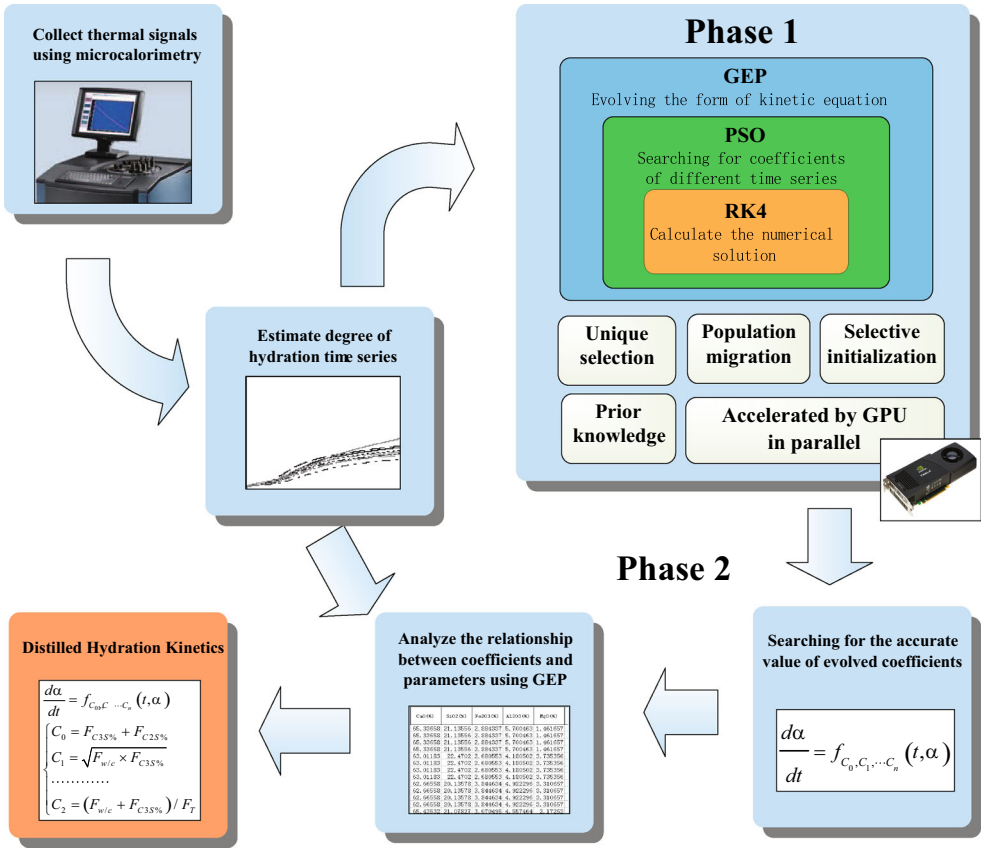

Fig. 2 Main flowchart of mining cement hydration kinetics

to determine the degree of hydration at a certain time point t as follows

$$\alpha(t) = \frac{Q(t)}{Q_{\max}} \tag{3}$$

where $\alpha(t)$ represents the degree of hydration at time t. $Q(t)$ represents heat released in hydration at the end of t. $Q_{\max}$ represents the total heat that can be released by a cement sample at the end of hydration. $Q(t)$ is determined directly using microcalorimeter. $Q_{\max}$ (J/g) is estimated according to the chemical composition of samples (Schindler and Folliard 2005). According to this approach, time series with the development of hydration can be collected (Wang et al. 2010).

3.2 Phase 1: inference of unified form of kinetic equation

The first problem in the first phase is how to find the best unified form of hydration kinetics. As a variant of genetic programming, GEP is adopted to focus on searching for optimal form of the equation in this phase. It is proposed by Ferreira (2001) for optimizing the computer program or function automatically. It combines the advantages of linear encoding from genetic algorithm and expression evolution from genetic programming via evolving the complex formula or program by operating a simple linear string which is coded by K-expression Ferreira (2001). Furthermore, compared with Multi Expression Programming (MEP) Oltean and Dumitrescu (2002), which is one of the most powerful tools in finding functions and programs, for this research, GEP is more suitable on account of complexity. The objective of this phase is to find a satisfied unified form, where the position of coefficients is fixed. Considering the characteristic of MEP chromosome, it can be decoded to several different expressions but only the best one is valid. Each of the expression represents a possible unified form of model. A coefficient in a gene can be decoded into different unified forms of model and will be optimized for different expressions, respectively, to find the best expression that can represent this chromosome. In this case, the time consumption of searching of MEP will increase sharply. In contrast, a chromosome of GEP can be decoded to a unique unified form of kinetics. Figure 3 shows an example of a GEP chromosome during hydration kinetics mining.

Another problem is how to evaluate the form of the equation. The value of coefficients are very different in different systems. Coefficients of all of the equations for every time series should be determined to evaluate forms in each generation of the GEP algorithm. Iba (2008) presented a method to deal with kinetics which contains transcendental functions

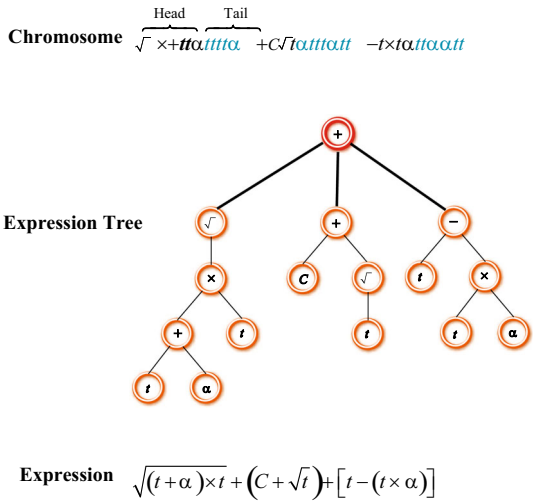

Fig. 3 GEP chromosome, its corresponding expression, and expression tree. There are three genes in this chromosome. Its link function is set to +. The head length of each gene is set to five while the maximum number of arguments is two. The tail is calculated as being 6 Ferreira (2001). The total length of the gene is the sum of the head length and tail length, i.e., 11. It is sufficient to decode an expression tree even using part of symbols (*black*) in front of the gene. The *blue* part in the gene is called the non-coding region which doesn't appear in the decoded expression tree (color figure online)

by searching for coefficients using traditional Levenberg–Marquardt method. However, this traditional method is easy to fall into the local extremum. Kennedy and Eberhart (1995) proposed PSO for global numerical optimization via emulating the swarm behavior. The individual in PSO, named particle, represents a point or a solution inside the searching space. There are two best positions recorded with the evolution of PSO: the best position found by the particle itself and the best position found by the whole population. The new positions of particles are updated by tracking both of them. Compared with genetic algorithm, PSO is easy to implement and tends to converge quickly, which benefits reducing the complexity of evaluating form of equation. Furthermore, some studies (Bae et al. 2010; Yan and Zeng 2006) have shown that PSO which performs a global search for finding optimum has the ability to aid genetic programming. Therefore, PSO is chosen to search for the best value of coefficients for a given form of kinetics.

In this phase, GEP evolves the unified equation form of kinetics iteratively by inputting the degree of hydration time series. The coefficients for each generated equation are optimized by PSO at each step of the GEP generation. It determines whether a potential unified form can fit the real-time series by adjusting its coefficients. However, to reduce the time complexity at this phase, PSO only runs with small population and generations to get a loose fit. When the unified form of kinetics and the coefficient for a certain series are fixed, the hybrid evolution calculates the numerical solution using the fourth-order Runge–Kutta method (RK4). This process is repeated until the termination conditions are satisfied, such as the maximum generation or the minimum error. To find the better form of kinetics efficiently and to avoid premature convergence to local extremum, the following strategies are used to aid the hybrid evolution to improve the performance and speedup the searching. This method is described in Algorithm 1.

Algorithm 1: Discovery of Unified Form of Kinetic Equation

```
Initialize chromosomes for all of GEP subpopulation at random
using selective initialization strategy;
while the maximum GEP generation has not been reached do
    while there exist unevaluated chromosomes do
        while there exist uncalculated time series do
            while the maximum number of trials has not been
            reached do
                Initialize particles (the combination of coefficients
                for current chromosome) at random;
                while the maximum PSO generation has not been
                reached do
                    while there exist unevaluated particles do
                        Calculate numerical solution using RK4
                        method for kinetic equation whose
                        equation form is current chromosome and
                        coefficients are the position of current
                        particle;
                        Calculate the relative error between
                        numerical solution and current time series
                        as the goodness of current particle;
                    end
                    Update pbest and gbest for particles;
                    Update position and velocity of particles;
                end
            end
            The best result is preserved for current time series;
        end
        Calculate the cost function for current chromosome;
    end
    if the current epoch has ended then
        Perform population migration strategy;
    end
    for every subpopulation do
        Tournament selection operator using unique selection
        strategy;
        Uniform mutation operator;
        One-point and two-point recombination operator;
        Insertion sequence and root insertion sequence operator;
        Gene recombination and transposition operator;
    end
end
```

3.2.1 Unique selection

The expressions in GEP are easily the same as each other as a result of the existence of non-coding region in the GEP chromosomes. If the length of the coding region is much smaller than the total chromosome length, there will be a lot of repeated chromosomes in the population. The repeated

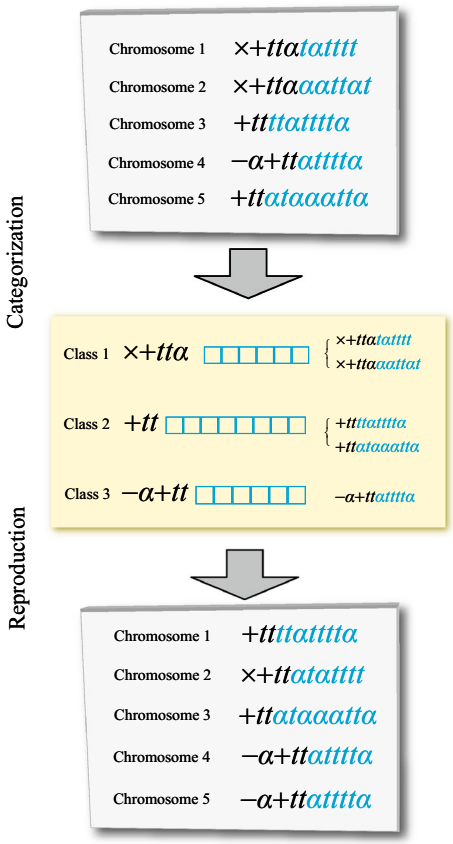

Fig. 4 The strategy of unique selection. The chromosomes are categorized into several classes. The selection operator is performed on different classes instead of different chromosomes. The *blue* parts represent non-coding regions (color figure online)

chromosomes are different in the whole linear sequence, but the same in the coding regions. Therefore, they will be decoded to the same expression trees.

When the coding region of a chromosome is shorter in length, there will be more chromosomes with the same effective coding region in the population. If they are better in cost, the corresponding chromosomes will be copied to the next generation with high probability, which often leads to premature convergence into local minima.

To solve this problem, a unique selection strategy (US) is proposed to avoid the repetition of expressions. At first, the chromosomes in the population are categorized into several classes. The chromosomes which have the same effective coding region are classified into the same class. Then, perform the selection operator on different classes instead of different chromosomes. If the selected class contains more than one chromosome, one of the chromosomes are selected randomly to the next generation. This strategy is shown in Fig. 4.

3.2.2 Population migration

In the theory of punctuated equilibrium, the species in a stable environment will not change if they bring about equilibrium. However, they will evolve rapidly when separated from the common ancestors. Applied to evolutionary computation, it is better to put chromosomes into several competing subpopulations rather than put all the chromosomes together in a single big-population when the total number of chromosomes is fixed Chen (2005). This model is named island model.

The island model has been used to improve the diversity of population for evolutionary computation in many works. As far as genetic algorithm is concerned, Zheng et al. (2014) found that the asynchronous island scheme, island/ master–slave hierarchy parallel genetic algorithm (PGA) and island/ cellular hierarchy PGA are the best for multi-core, multi-socket multi-core and many-core architectures, respectively. Guan and Szeto (2013) discussed the topological features of the communication network between computing nodes in PGA under the framework of the island model. As far as genetic programming is concerned, Vega et al. (2004) presented a new proposal for reducing bloat in genetic programming and concluded that island model helps to prevent the bloat phenomenon. For GEP, Du (2008) proposed a new asynchronous distributed parallel gene expression programming based on Estimation of Distribution Algorithm. The improved GEP is implemented by an asynchronous distributed parallel method based on the island model on a message passing interface environment.

A population migration (PM) strategy, which is a kind of island model, is adopted in the first phase of this research. The population of GEP is divided into several subpopulations which evolve the form of kinetics independently. The best chromosome in the whole population is selected after an epoch (a certain number of iterations). Then, the worst chromosome in all of subpopulations is replaced by the selected best chromosome according to a certain probability. Figure 5 illustrates the population migration strategy.

3.2.3 Selective initialization

Numerical solution of differential equations is very sensitive to small changes in equation form. This effect is more apparent when there exists transcendental functions. Therefore, the cost of initialized chromosomes is often beyond the maximum or minimum representable floating-point number. To fully develop the performance of the GPU platform, the single-precision floating point is adopted in experiment as the speed of single precision is higher than double precision. Its maximum representable number is 3.4e+38 while the minimum representable number is −3.4e+38. If the initialization process is not improved, only a small number of chromosomes in the initial population will achieve superior cost and the remaining ones will be beyond the representable number. As a result, these “not very bad” chromosomes will

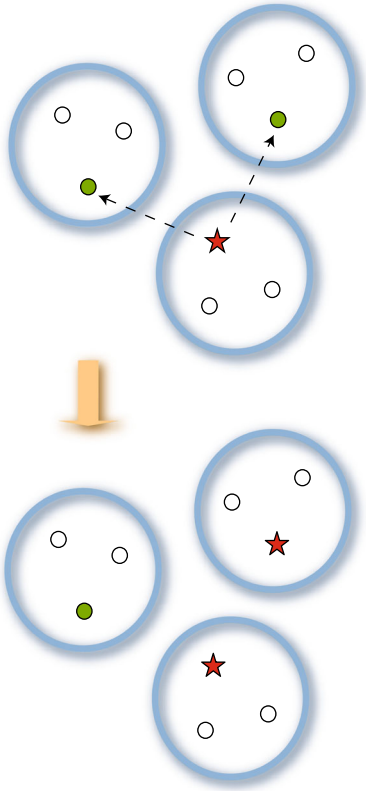

Fig. 5 Population migration. The *green circle* represents the worst chromosome in the subpopulation. The *star* represents the best chromosome in the whole population (color figure online)

soon control the whole population and result in premature convergence.

Therefore, to avoid the "not very bad" chromosomes dominating the whole population rapidly and to strengthen the competition between chromosomes in the early-age of evolution, some chromosomes are produced randomly in the initialization of the population. The chromosomes whose cost is smaller than a predefined value are picked out; add them into the initial population and go back to the first step of random production. It is repeated until the number of chromosomes in the initial population meets the predefined size. This strategy is named selective initialization (SI). According to this way, the competitiveness of initial chromosomes can be improved.

3.2.4 Prior knowledge

Prior knowledge is explained as a combination of preexisting experiences and knowledge. In the field of cement hydration modeling, scientists have already developed some models based on first principle method. The knowledge from these models can be borrowed to narrow down the searching scope of GEP to find better equations. Therefore, the Krstulovic–Dabic's model (Krstulovic and Dabic 2000) and Tomasawa's model (Tomosawa 1997) are decomposed into some independent blocks and added to the function set and terminal set, respectively. The function set and terminal set are shown as follows

$$\text{Function Set} = \{+,\ -,\ \times,\ \div,\ -1\times(\ ),\ \frac{1}{(\ )},\ \sqrt{},\ \sqrt[3]{},\ (\ )^2,\ (\ )^3,\ (\ )^{(\ )},\ e^{(\ )},\ \ln(\ )\}$$

$$\text{Terminal Set} = \{C,\ \alpha,\ t,\ 1-\alpha,\ \sqrt[3]{1-\alpha},\ \sqrt[3]{(1-\alpha)^2},\ \ln(1-\alpha),\ 1-\sqrt[3]{1-\alpha},\ \alpha^3\}$$

where C represents the constant term, i.e., coefficient term. After adding the coefficient term into the terminal set, the number and positions of coefficients can be determined by the GEP itself. The term C at a different position implies a different coefficient. The number of C in each GEP chromosome is counted before running the PSO algorithm to fit the form to different series. Another essential task is to choose the link function in the GEP chromosome to connect the genes. In the theory of cement hydration, the hydration rate approaches to zero gradually with the development of hydration.

$$\lim_{t\to\infty}\frac{\mathrm{d}\alpha}{\mathrm{d}t}=0 \tag{4}$$

This research's preliminary study of different link functions (inverse proportional function, exponential function, etc.) which can implement this characteristic suggests that the following function is the best link function between genes (results not shown here)

$$\frac{1}{(\ )(\ )\cdots(\ )} \tag{5}$$

3.2.5 GPU acceleration

The first phase is very time-consuming because of the introduction of hybrid evolution. To accelerate the evolution of form of hydration kinetic equation, the study adopted a master/slave architecture and divided it into two parts: the main serial part performed on the CPU and the parallel evaluation part performed on the GPU. The fourth-order Runge–Kutta method, which is used to calculate the numerical solution of α to find out relative error of these time series on the condition that equation form and coefficients combination are given, is accelerated using GPU threads in parallel. The CPU requests the GPU to carry out the evaluation for all equations, after the CPU finished the update of them in each iteration. At the same time, the CPU will not resume to the active state until the GPU returns all of requested tasks. The evaluation of kinetics consists of solving differential equation and calculating cost function. Each of them occupies one thread and runs in parallel on the GPU independently. Figure 6 shows the sequential relationship.

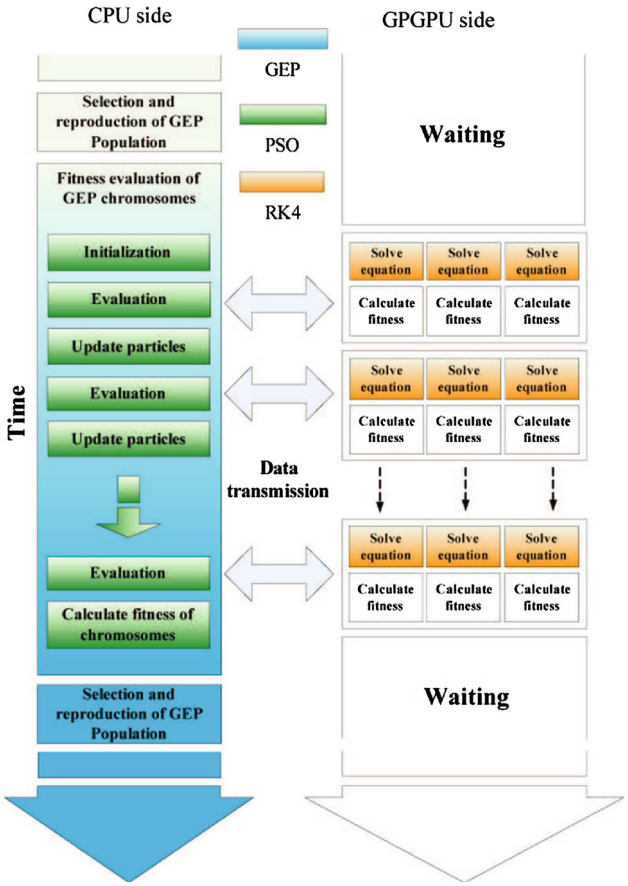

Fig. 6 Sequential relationship of GPU acceleration for phase 1

In the aspect of computing, to further improve the performance of execution on the GPU, the number of memory operations inquired by threads should be reduced because the memory operation is much slower than the register operation (CUDA 2009). Therefore, the expression tree is consolidated on the CPU first and then recoded into a new K-expression, which will be calculated on the GPU. The process of consolidating can be described as follows. First, it needs to know whether there is a terminal node in the child branch. If not, go to the next function node. If there are several terminal nodes in the branch, the function node and terminal nodes are combined into a new single terminal node. This new node is used to replace the original function node. The operation of the combination is limited only for one upper layer. There is no recursive operation in consolidating. Figure 7 shows the method of consolidating an expression tree. On the assumption that a function node is a binary operator, the thread will access memory for three times if tree branches were not consolidated. On the contrary, the thread just needs to read the new node from memory only one time if the branch of the expression tree was consolidated. Therefore, the output value of the original function can be calculated more quickly.

In the aspect of storage, each K-expression is divided into two parts: one part is expression string and another part is corresponding value of coefficients. When tree consolidating operation is completed by the CPU, the recoded K-expression and its corresponding coefficients will be transferred from main memory (in the host) to global memory (in the GPU module) once and for all. A block of shared memory in multiprocessor is allocated to thread during initialization. Then, this thread copies the string of expression tree from global memory to its own shared memory and copies the coefficients to its corresponding local memory. The K-expression string is stored in shared memory because shared memory space in multi-processor is too few to store all information of K-expression. Moreover, as far as calculation speed is concerned, the frequency of function node access is much higher than coefficients access in solving differential equation.

3.3 Phase 2: inference of relationship between coefficients and parameters

The relationship between hydration parameters (such as chemical composition, particle size distribution and curing conditions) and their influence on different period of cement hydration are also needed to be analyzed and investigated. They are coefficient functions consisting of hydration parameters. Different from distilling the unified form of equation in the pervious phase, it is necessary to extract the defi-

Fig. 7 Consolidating of expression tree

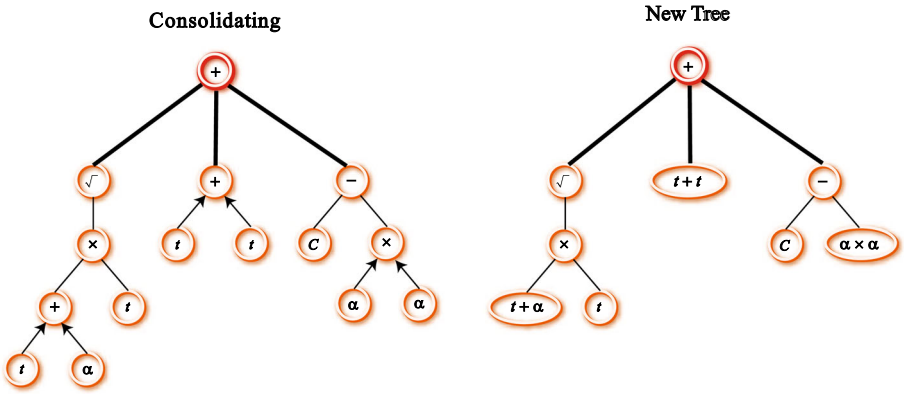

Table 1 Cement hydration parameters

F_0	CaO (%)
F_1	SiO_2 (%)
F_2	Fe_2O_3 (%)
F_3	Al_2O_3 (%)
F_4	MgO (%)
F_5	SO_3 (%)
F_6	K_2O (%)
F_7	Na_2O (%)
F_8	fCaO (%)
F_9	Specific surface (m^2/kg)
F_{10}	Sieve residue on 74 μ (%)
F_{11}	Water-to-cement ratio (w/c)
F_{12}	Reaction temperature (°C)

nite functional relation from cement hydration parameters to coefficients at this phase. Therefore, this phase continues to use the GEP algorithm to find the function from parameters to coefficients and focus on the searching of definite functional relation but not the form.

At first, based on the distilled unified form of kinetics, for different samples (systems), the PSO algorithm with bigger population and more generations are still adopted to get a more accurate value of coefficients. Cost function and other parameters remain unchanged. Then, for each coefficient, we have a set of parameters from different samples and their corresponding value of coefficient. Taking the hydration parameters of every sample as inputs, and their corresponding coefficients as outputs, a data set can be obtained. The functional relationship from parameters to coefficients can be found by approximating this data set using the GEP algorithm. The hydration parameters used in this phase include thirteen items, $F_0, F_1, \ldots, F_{12}$. The meanings of them are shown in Table 1.

Since the definite functions between coefficients and cement hydration parameters are still unknown in current manually derived models, there is no prior knowledge which can be used to determine function set and terminal set. Therefore, the terminal set is made up of thirteen hydration parameters and ephemeral random constants. For function set, various functions are tested to find the best combinations. Then, function set and terminal set which are shown as follows are adopted at this phase

$$\begin{aligned}\text{Function Set} = \{&+,\ -,\ \times,\ \div,\ \sqrt{\ },\ \sqrt[3]{\ },\ (\ )^2,\ (\ )^3,\\ &e^{(\ )},\ \ln(\ )\}\\ \text{Terminal Set} = \{&?, F_0, F_1, F_2, F_3, F_4, F_5, F_6, F_7, F_8,\\ &F_9, F_{10}, F_{11}, F_{12}\}\end{aligned}$$

The term "?" represents the ephemeral random constants. For each chromosome, the constants are randomly generated at the beginning of a run, but remain unchanged during evolution. Their circulation is guaranteed by the genetic operators.

Table 2 Chemical compositions and physical properties of the cements prepared

	F_{01}	F_{31}	F_{38}	F_{51}
CaO (%)	65.34	63.01	62.67	65.44
SiO_2 (%)	21.14	22.47	20.14	21.08
Fe_2O_3 (%)	2.88	2.68	3.84	3.07
Al_2O_3 (%)	5.70	4.18	4.92	4.56
MgO (%)	1.46	3.74	3.31	2.17
SO_3 (%)	2.01	2.47	2.49	2.45
K_2O (%)	0.45	0.69	0.35	0.62
Na_2O (%)	0.14	0.13	0.16	0.13
fCaO (%)	0.99	1.35	1.44	1.43
C3S (%)	53.08	41.24	50.50	58.31
C2S (%)	20.63	33.39	19.72	16.53
C3A (%)	10.23	6.55	6.55	6.89
C4AF (%)	8.77	8.15	11.69	9.33
Specific surface (m^2/kg)	360	344	340	346
Sieve residue on 74 μ(%)	3.20	3.20	3.80	2.80
Q_{max} (J/g)	481.26	447.09	470.24	484.09

4 Experiments

4.1 Data collection

The Portland cement specimen F_{01}, F_{31}, F_{38} and F_{51}[1] were used in the chemical experiment. The details of chemical composition and physical properties of these cements are listed in Table 2. The weight ratios of tricalcium silicate (C3S), dicalcium silicate (C2S), tricalcium aluminate (C3A), and tetracalcium aluminoferrite (C4AF) are estimated according to Bogue formula (Bogue 1955).

Two grams of samples of Portland cement and water were added into the container of microcalorimetry (TAM Air 8-channel calorimeter) using various combination of experimental parameters. For different time series, the combination of water-to-cement ratio, curing conditions, and cement type are shown in Table 3. To get the data of middle-age hydration, the duration of hydration was fixed to 5 days and the sampling interval was set to 30 min (totally 240 time points). Sixteen groups of time series of degree of hydration were collected by feeding obtained heat flow information from microcalorimetry and the total heat can be released by the current sample at the end of hydration to Eq. (3). Figure 8 shows the collected time series.

[1] These specimens were provided by The Cement Quality Supervision and Inspection Station of Shandong Province, China.

Table 3 The cement type, water-to-cement ratio, and curing conditions of each time series

Times series	Specimen type	Water-to-cement ratio (w/c)	Reaction temperature (°C)
0	F_{01}	0.35	22
1	F_{01}	0.35	35
2	F_{01}	0.55	22
3	F_{01}	0.55	35
4	F_{31}	0.35	22
5	F_{31}	0.35	35
6	F_{31}	0.55	22
7	F_{31}	0.55	35
8	F_{38}	0.35	22
9	F_{38}	0.35	35
10	F_{38}	0.55	22
11	F_{38}	0.55	35
12	F_{51}	0.35	22
13	F_{51}	0.35	35
14	F_{51}	0.55	22
15	F_{51}	0.55	35

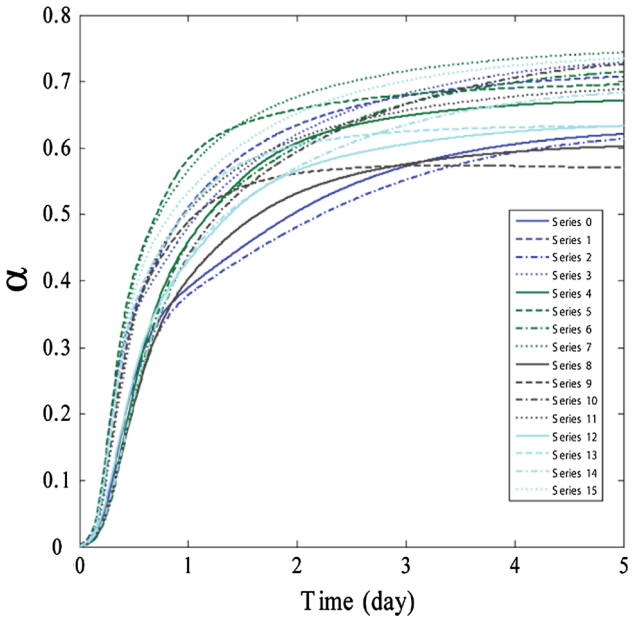

Fig. 8 Collected time series of degree of hydration within 5 days

4.2 Experiment settings

The goodness of distilled kinetics in the first phase was measured by cost function. It determines wether the kinetics performs well for hydration simulation. The cost function for a single time series is defined as follows

$$\text{Cost}' = \frac{\sum_{j=0}^{n-1} \left| \beta \frac{\alpha_{ij} - \widehat{\alpha_{ij}}}{\alpha_{ij}} \right| + \sum_{j=n}^{m-1} \left| \frac{\alpha_{ij} - \widehat{\alpha_{ij}}}{\alpha_{ij}} \right|}{m} \tag{6}$$

Table 4 Experiment settings in the first phase

	Parameters	Settings
GEP	Total population size	960
	Number of Subpopulation	4
	Number of generations	1000
	Number of generations in epoch	5
	Migration probability	0.1
	Number of genes	3
	One-point recombination probability	0.4
	Two-point recombination probability	0.4
	Gene transposition probability	0.1
	Gene recombination probability	0.1
	Mutation probability	0.003
	Insertion sequence probability	0.1
	Root insertion sequence probability	0.1
	Head length	30
	Initialization maximum cost	1
	Selection operator	Tournament with size 2
PSO	Population size	16
	Number of generations	200
	$\varphi 0$	1
	$\varphi 1$	1.8
	$\varphi 2$	1.8
	Number of trails	4
GPU	Number of threads in each block	512
	Number of blocks in each device	960
	Number of devices	2
Cost	β	10

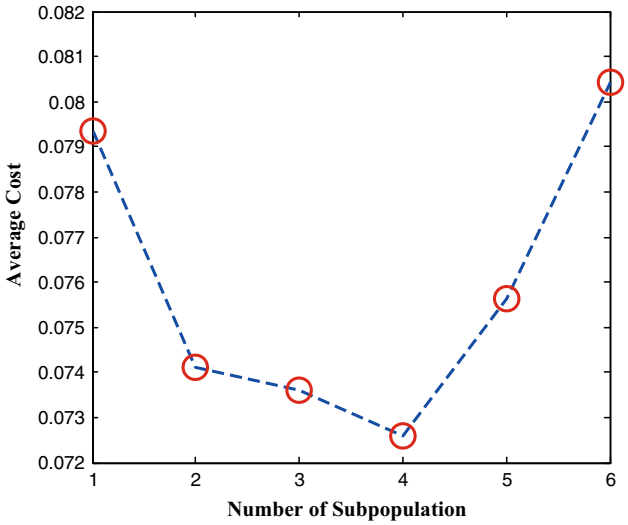

Fig. 9 The variation of number of subpopulation on the cost value

where α_{ij} represents the real value of α at time j in time series i. $\widehat{\alpha}_{ij}$ represents the numerical solution of kinetics at the same time point. m represents the number of time points in each time series. Each of series consists of 240 time points, which means $m = 240$. n represents the number of points in the induction period. β is a weight constant. The induction period plays very important role in the cement hydration. However, the number of time points in the induction period is less than 3 % among all of time points in a time series. It leads to the fact that the evolved curve often mismatches this period. Therefore, n and β are introduced to solve this problem by increasing the weight of this period. Since the unhydrated state of cement remains unchanged at the moment of mixing with water, the degree of hydration at $t = 0$ is set to zero. Since the induction period usually runs for two hours, n is set to 5. Since the early-age hydration of cement is very dramatic and the performance of set cement is closely related to early-age hydration, this equation is measured by percent errors to emphasize early-age hydration whose α is small. According to Eq. (6), the cost function for the whole data is defined as follows

$$\text{Cost} = \frac{\sum_{i=0}^{N-1} \text{Cost}'}{N} = \frac{\sum_{i=0}^{N-1} \left(\sum_{j=0}^{n-1} \left| \beta \frac{\alpha_{ij} - \widehat{\alpha_{ij}}}{\alpha_{ij}} \right| + \sum_{j=n}^{m-1} \left| \frac{\alpha_{ij} - \widehat{\alpha_{ij}}}{\alpha_{ij}} \right| \right)}{mN} \tag{7}$$

where N represents the number of time series. The number of group of time series is sixteen, which means $N = 16$. The smaller the value of cost is, the better the distilled kinetics is.

The settings of experimental parameters affect the evolution process and sensitivity of model greatly. Considering the high time complexity of proposed method, and the probability to find optimal solution increases gradually with the increase of size of population and maximum number of generations, the settings are tuned by running preliminary experiment under small population size and few generations. Then, the population size and number of generations are increased in the formal experiment. After trial and error, the settings shown in Table 4 were adopted in the first phase.

Particularly, as far as the adopted strategies are concerned, Fig. 9 and Table 5 shows the influence of number of subpopulation and different strategies, respectively. About the experimental settings of Fig. 9 and Table 5, for GEP, the total population size is decreased to 60 while the number of generations is decreased to 100. For PSO, the number of trails is set to 2 while the number of generations is set to 10. The size of tournament is set to 4 to accelerate convergence under few generations. The other parameters remain unchanged. In Fig. 9, it can be observed that the best cost is achieved when subpopulation size is set to 4. Moreover, the results shown in Table 5 further confirms the effectiveness of adopted strategies in comparison with the traditional method. The combined strategy PM+US+SI exhibited the lowest average cost.

Table 5 The influence of population migration (PM), unique selection (US), and selective initialization (SI) on results of cost [Eq. (7)]

Trail	Traditional	PM	US	SI	PM+US+SI
1	8.18E−02	7.74E−02	7.14E−02	6.53E−02	5.90E−02
2	5.26E−02	9.07E−02	5.79E−02	6.68E−02	6.53E−02
3	9.76E−02	6.44E−02	5.59E−02	1.04E−01	5.90E−02
4	5.19E−02	7.77E−02	7.56E−02	6.09E−02	4.47E−02
5	1.04E−01	3.55E−02	6.00E−02	5.01E−02	6.25E−02
6	5.92E−02	9.75E−02	8.40E−02	7.10E−02	5.95E−02
7	9.20E−02	5.94E−02	5.90E−02	6.43E−02	5.81E−02
8	7.47E−02	6.38E−02	5.46E−02	7.98E−02	6.35E−02
9	8.00E−02	9.02E−02	7.00E−02	1.01E−01	5.14E−02
10	9.53E−02	6.93E−02	6.57E−02	6.15E−02	7.16E−02
Mean	7.89E−02	7.26E−02	6.54E−02	7.24E−02	5.95E−02

Traditional represents the method which closes population migration and selective initialization, and adopts ordinary tournament selection. PM+US+SI represents the method which adopts all three strategies

Table 6 GEP settings in the second phase

Parameters	Settings
Total population size	200
Number of generations	50,000
Number of genes	3
One-point recombination probability	0.4
Two-point recombination probability	0.4
Gene transposition probability	0.1
Gene recombination probability	0.1
Mutation probability	0.003
Insertion sequence probability	0.1
Root insertion sequence probability	0.1
Head length	20
Selection operator	Tournament with size 4
Cost function	Root mean square error
Link function	+

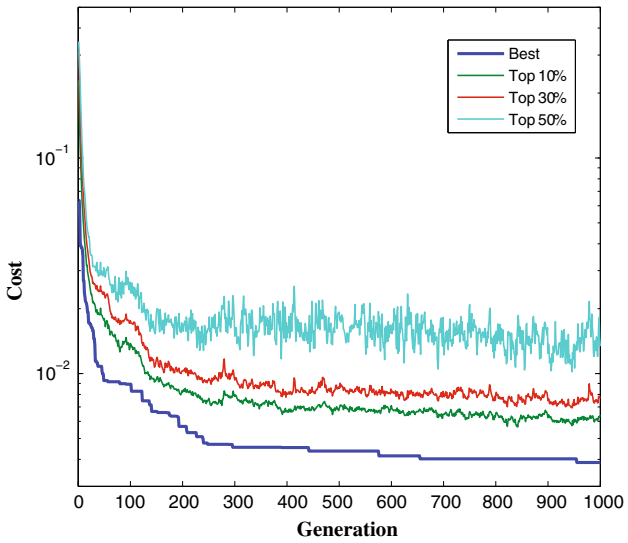

Fig. 10 Evolution process of unified form of kinetic equation

In the second phase, to search for the best value of coefficients, there are 7680 particles are used simultaneously. The maximum generation of PSO is increased to 1000. After finding the value of coefficients, the settings of GEP in this phase are also tuned by preliminary experiment. Table 6 shows the formal settings used by GEP to extract the relation between hydration parameters and kinetics coefficients. Because the objective at this phase is coefficient, but not the degree of hydration, the root mean square error (RMSE) is adopted as cost function

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=0}^{N-1} \left(C - \widehat{C}\right)^2}{N}} \tag{8}$$

where C is the value of coefficient, $\widehat{C}$ is the predicted value of coefficient, and N is the number of time series.

A GPU-based desktop powerful computer with Linux operation system and C programming environment was selected as our experimental platform. This powerful com-

Table 7 Comparison of execution times and speedup ratios between GPU and CPU

Generation	Execution time on CPU (s)	Execution time on GPU (s)	Speedup ratio
100	12,046	1079	11.15
200	10,003	1097	9.11
300	8878	1136	7.81
400	9020	1038	8.69
500	11,538	1048	11.00
600	10,776	1023	10.53
700	10,360	1084	9.56
800	10,617	1092	9.72
900	10,708	1129	9.48
1000	9566	1106	8.65
Mean	10,352	1083	9.57

puter contains two high-performance Tesla C1060 GPUs produced by NVIDIA. This GPU contains 30 stream multiprocessors (SMs) with eight stream processors (SPs) on it, totally 240 SPs. The clock frequency of SP is 1.3 GHz and the size of device memory is 4GB. Each multi-processor contains 16KB of shared memory and 16,384 registers. Therefore, this powerful computer reaches the peak speed at 2Tflops.

4.3 Results and discussion

It took 1,071,446 s (297.5 h) to perform the first phase to distil the unified form of middle-age hydration kinetics for Portland cement. On the one hand, the initialization process spent 1751 s (half an hour). On the other hand, the equation evolution process spent 1,069,695 s (297 h). The evolution process of middle-age kinetics is shown in Fig. 10, which illustrates the variation of cost of the historically best chromosome. Moreover, to illustrate the information from the whole population, the variations of average cost of top 10 %, top 30 % and top 50 % chromosomes in the population are also plotted in the same figure. This figure shows that the evolution process converges gradually after 250 generations. Furthermore, the fluctuation range of top 50 % is larger than it of the others, which reflects the nonuniformity of the population. Table 7 illustrates the comparison of execution times and speedup ratios between GPU and CPU. It is obtained by performing the GEP population in 100th, 200th, 300th, 400th, 500th, 600th, 700th, 800th, 900th, 1000th generation on CPU (Intel Core i7 920) again. The average speedup ratio is 9.57. Although it took a long time for distilling the result, considering the frequency of execution of reverse extraction, the time consumption is acceptable because the objective of this research is hydration kinetics itself but not the system of reverse extraction.

After evolution of first phase, the optimal equation with the best cost 0.00387 is obtained and shown in Eq. (9) by decoding the optimal chromosome to expression and simplifying it. It is the middle-age hydration kinetics of Portland cement mined from observed data. Its unit of time is half an hour.

$$
\begin{aligned}
\frac{\mathrm{d}\alpha}{\mathrm{d}t} &= C_5\left(\frac{C_7}{\mathrm{e}^{e^{3t}}} - C_6 - \alpha\right)\left[\left(\mathrm{e}^{\left[(1-\alpha)^{C_2\log^2(1-\alpha)}-C_1\right]\alpha^{\frac{\mathrm{e}^{C_3+1-\sqrt[3]{1-\alpha}}}{t}}} - C_0\right)\left(e^{(\sqrt{1-\alpha}-C_4)\alpha^{\frac{1-\alpha}{\alpha}}} - \log(1-\alpha)\right)\right]^{1-\alpha} \\
&\left\{
\begin{array}{l}
C_0 = F_6 - \dfrac{0.232744}{\frac{F_{12}F_6^2}{F_{11}} - \frac{F_5}{F_6} - \frac{\sqrt[4]{F_6}}{\ln^3 F_4} - F_7 + F_8} + 0.141344\dfrac{\left(\frac{F_{10}}{F_{12}-\frac{F_3}{F_7}} + F_3\right)^6}{F_1(F_7F_{12})^6} + \dfrac{F_7F_8}{\sqrt{\frac{\sqrt{F_{11}}}{\ln F_2} - F_8}{\sqrt{0.08536F_7+F_{11}}} + \frac{F_0}{F_{12}}}} \\
C_1 = \dfrac{F_2F_8}{F_1F_9\left(\frac{F_5^2}{F_1^2} - \frac{F_{11}}{F_{12}}\right)} - \dfrac{F_6F_7F_1^2}{F_{12}(F_4F_7-F_{11})(F_7-F_0-F_1+F_{11}F_1^2)} + \dfrac{0.130672F_5(F_2-F_0)(F_0+F_{11})}{F_9F_4^2(F_1-F_{12})} + \ln\dfrac{F_4^2}{F_3F_{10}} + \dfrac{F_5}{F_{12}} \\
C_2 = 0.000222F_8\left(\dfrac{F_{12}}{F_2^3F_4F_{11}}\right)^{12} + \left(0.858915F_2^2F_{11}^2F_6^4F_8^4\right)^3 + \dfrac{F_1^2F_3^2}{F_4^2F_5^2F_6^2\left(\left[\frac{0.372273+(F_2-F_{11})^3}{F_1}\right]^3 - F_{12}\right)^2} \\
C_3 = \dfrac{F_3}{F_{12}-F_2+\frac{F_{11}}{F_{11}-\frac{F_{12}-F_4^3F_6^3F_{11}^3}{F_0}}} + \dfrac{F_4F_6F_8^2F_{11}^3\ln(\ln(F_{10}))}{F_7F_3} + \dfrac{F_3}{F_{12}}\sqrt[3]{F_0 + \dfrac{e^{F_6}F_5}{\frac{F_0}{F_{12}} - F_4^3F_6^3 + F_4F_7}}} \\
C_4 = \dfrac{F_2F_5F_7F_{11}F_{12}}{F_3F_{11}(F_3-F_{11})-F_4F_5F_7F_{12}} + \dfrac{F_8F_{11}}{\frac{(F_3-F_{10})F_{12}}{F_0} - \frac{F_{10}}{F_8} - F_3F_{11} + F_2} + \dfrac{F_7}{\frac{F_0}{F_3(F_4+F_{11})} - 2F_{10}} + 2\dfrac{F_3}{F_4} + F_3 - F_2 \\
C_5 = \dfrac{1}{e^{F_4}\sqrt{F_1}\left(F_8+F_{11}+F_{12}-\frac{F_2F_4}{F_{11}}\right)} + \dfrac{0.124426}{F_{11}\left(F_0 - \frac{2F_6F_{12}\sqrt[3]{F_4-F_{11}+0.075591}}{\sqrt[9]{\frac{F_3}{F_{11}}}}\right)} + \dfrac{\sqrt{F_1}}{e^{F_4}F_{12}\left(\frac{F_2F_7}{F_2(F_6-F_{11})} + F_{11} + F_{12} - e^{F_2}\right)} \\
C_6 = \dfrac{F_3F_5F_7F_6^3F_{11}^3}{F_4} + \dfrac{2.19931F_6F_{11}}{F_5\left(F_{12} - \frac{0.177593}{F_6^2F_7^2\sqrt{F_3}}\right)} + \dfrac{F_7^{\frac{4}{3}}}{F_{11}}\sqrt[3]{\dfrac{F_7^2}{\ln\left(\frac{F_3F_8}{F_1} + \sqrt[3]{F_{11}}\right)}} + 2F_8 - 0.944175 \\
C_7 = \dfrac{F_6F_{12}}{F_2-F_6^2+F_1F_2^2F_4^2F_6^3F_{11}^8} + \left(\dfrac{F_{12}}{F_1 - \frac{F_{11}F_{12}}{F_{12}-F_1+F_{12}(F_6-F_{11})+\frac{F_{11}-F_8}{F_3F_6}}}\right)^3 + (F_{11} - 0.542355) \\
\left[F_7(F_2+F_7+F_{10}) + F_8 + F_{10} - 0.542355\right]^2
\end{array}
\right. \qquad (9)
\end{aligned}
$$

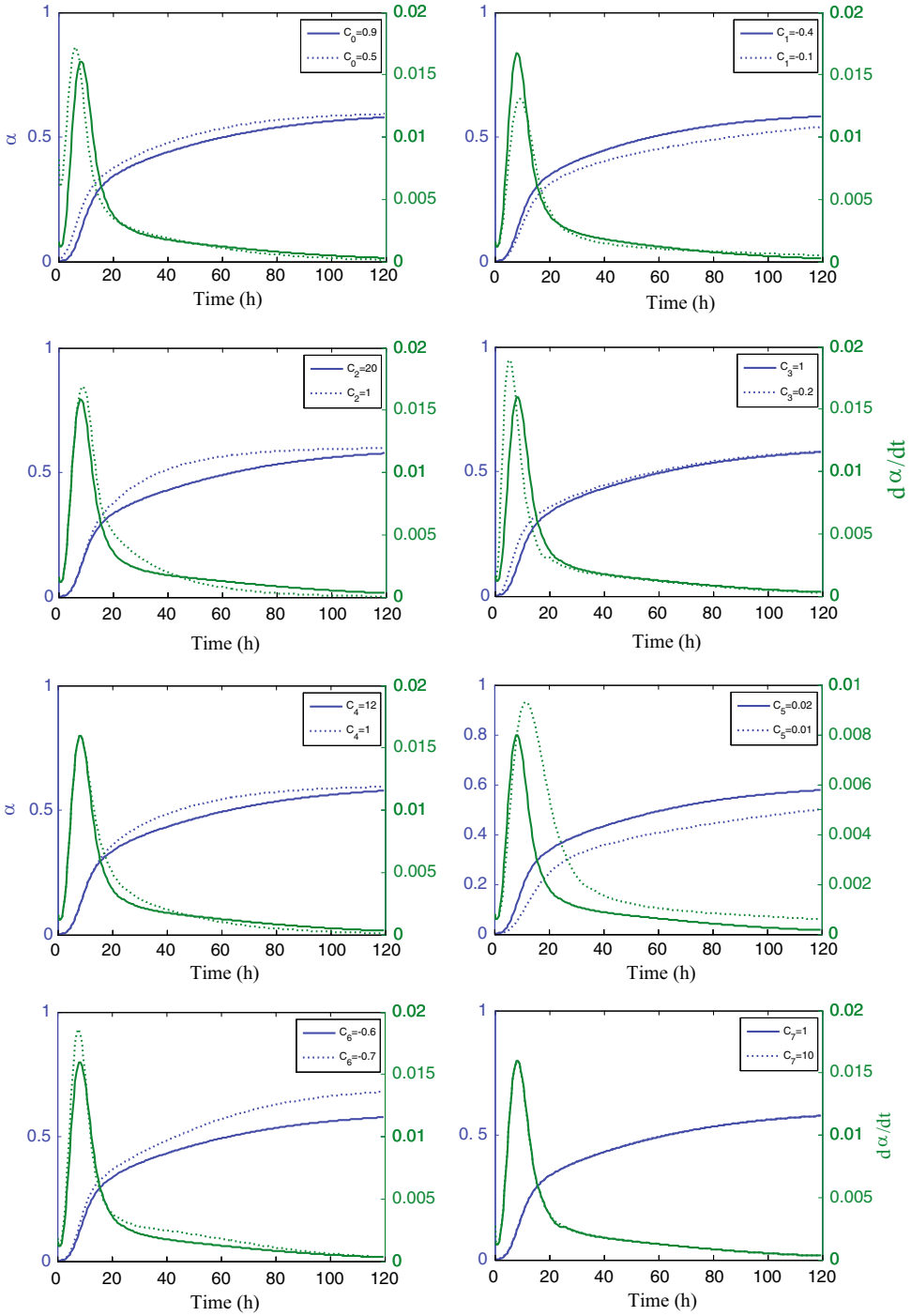

Fig. 11 The impact of coefficients on the hydration process

There are eight coefficient terms in the obtained GEP chromosome. As mentioned above in Sect. 3, they are different coefficients. To distinguish them, these coefficients are identified by C with a subscript according to the order in chromosome, named $C_0, C_1, \ldots, C_7$. Despite the difficulties of understanding the physical meaning of these coefficients in the distilled kinetics, their affection can be evaluated on different periods of cement hydration. Figure 11 shows the impacts of change of value of coefficients on hydration rate and degree of hydration simulated by distilled kinetics. It can be observed that the main effect of C_0 on simulation is to adjust the proportional relationship between induction period and the following periods. The decrease of C_0 reduces the durations of induction period and accelerating period.

Table 8 The comparison of cost between Krstulovic–Dabic's model, Tomosawa's model, early-age model, and distilled model [Eq. (9)]

	Krstulovic–Dabic's model	Tomosawa's model	Early-age model	Distilled model
Series 0	0.41167	0.04453	0.05759	0.00272
Series 1	0.38190	0.04382	0.11764	0.00289
Series 2	0.47326	0.05296	0.05358	0.00437
Series 3	0.44928	0.05379	0.14164	0.00452
Series 4	0.89866	0.14654	0.07507	0.00614
Series 5	0.64103	0.07739	0.14628	0.00370
Series 6	0.62233	0.07616	0.42886	0.00115
Series 7	0.62741	0.08867	0.14454	0.00258
Series 8	0.53913	0.05966	0.08930	0.00116
Series 9	0.69562	0.14207	0.39357	0.00100
Series 10	0.35713	0.02686	0.16312	0.00163
Series 11	0.55596	0.07959	0.14220	0.00197
Series 12	0.43481	0.04500	0.14308	0.00425
Series 13	0.82162	0.14865	0.42303	0.00311
Series 14	0.47353	0.05720	0.03610	0.00182
Series 15	0.59853	0.08457	0.13255	0.00130
Mean	0.56137	0.07672	0.16801	0.00277
SD	0.15411	0.03822	0.12866	0.00150

Furthermore, the hydration rate in terminating period are also reduced with the decrease of this coefficient. The effect of C_1 is to adjust hydration rate of accelerating period and decelerating period. After the induction period, the larger the C_1, the higher the peak of hydration rate. C_6 has the same effect as C_1. Nevertheless, the larger the C_6, the lower the peak of hydration rate. C_2 and C_4 are used to adjust decelerating period and terminating period. Larger value of C_2 and C_4 increases the hydration rate in decelerating period and decreases it in the terminating period. The main difference between both of them is C_2 affects the peak of hydration rate. C_3 adjusts the duration of induction period, accelerating period, and decelerating period. The duration of induction period and accelerating period decreases with the decrease of C_3. However, the duration of decelerating period increases. There is an integral impact on every period of simulation curve using C_5. For C_7, the effect on hydration rate in induction period can be also observed. However, different from C_1, it has little effect on the other periods.

Table 8 compares the distilled middle-age hydration kinetics with three other kinetics models, including the Krstulovic–Dabic's model (Krstulovic and Dabic 2000), Tomosawa's model (Tomosawa 1997), and early-age hydration model (Wang et al. 2010). For a fair comparison, the coefficients in these models were determined from fits to experimental data to get the best matching. The PSO algorithm whose population size is 7680 and maximum generation is 1000 is adopted to find the best coefficients for all of the compared models. This table depicts the results for comparison between Eq. (9) and the other models. The statistical analysis t test (Box 2005) was also performed for a thorough comparison of the distilled middle-age hydration kinetics and the other algorithms. The t test results show that the distilled middle-age model performs significantly better than the compared algorithms, with a significance level of 0.05. Moreover, the smallest standard deviation (SD) is also produced by the Eq. (9) which means it is more stable. Although the manually derived models simulate all life of cement hydration, the nature of reverse distilling method makes the discovered model especially accurate in middle-age hydration. Furthermore, the results of early-age model show that it is not effective in simulating middle-age hydration though it can simulate all life of cement hydration besides the early-age hydration. The results indicate that the distilled unified form of kinetic equation is fully capable of modeling middle-age cement hydration kinetics.

In the second phase, the relationship between the eight coefficients and cement hydration parameters $F_0, F_1, \ldots, F_{12}$ are extracted and listed in Eq. (9). It took 1831 s (half an hour) to perform the second phase. The evolution processes for all of coefficients are shown in Fig. 12, which illustrates the variation of RMSE of the historically best chromosome. This figure shows that the evolution process converges gradually after 20,000 generations. Figure 13 illustrates the accuracy of coefficients, which is calculated using extracted functional relationship, to the target value which is optimized

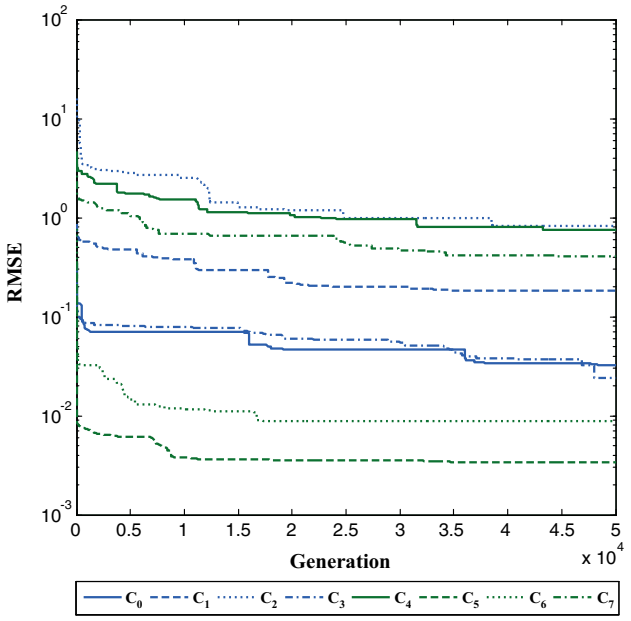

Fig. 12 Evolution processes for all of coefficients in the second phase

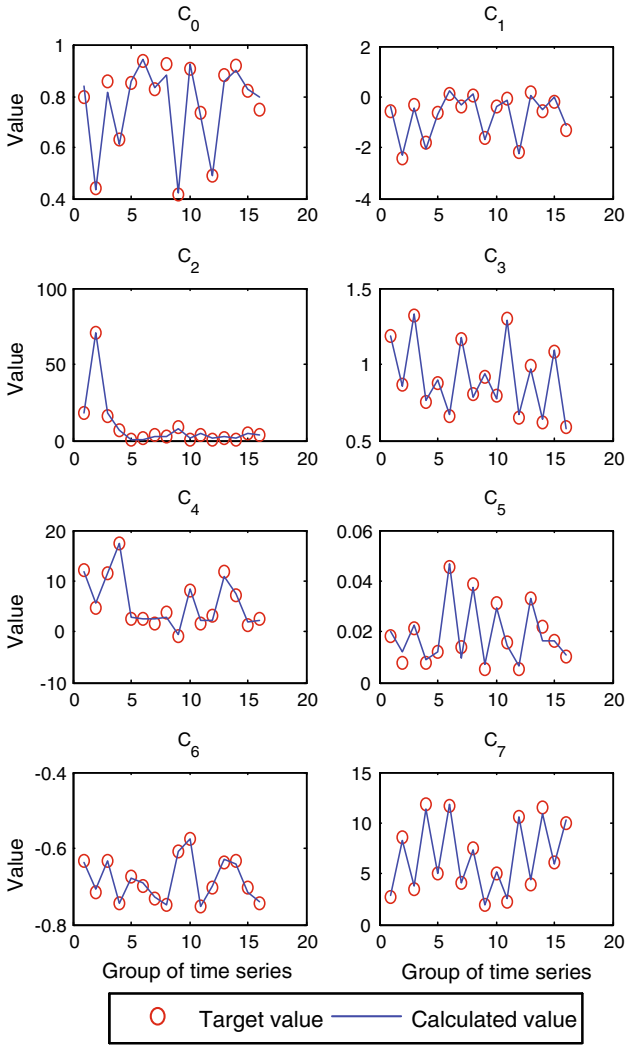

Fig. 13 Accuracy of calculated coefficients

using particle swarm optimization. It can be observed that there is high similarity between calculated coefficients and the target.

Based on the distilled middle-age hydration kinetics, the curve of degree of hydration and hydration rate can be simulated. Figure 14 illustrates the comparison between simulation curves and experimental curves. It can be observed that the simulation curves of hydration are in agreement with experimental data on series 0, series 4, series 5, series 7, series 9, series 10, series 12, series 14, series 15 compared with the others. The remaining ones are generally in accordance with experimental data, except time series 6. In addition, the distilled kinetics not only simulate periods after the induction period but also simulate the induction period of Portland cement successfully, which should be attributed to the introduction of n and β. Considering the coefficients in the other models are determined from fits to experimental data but not calculated directly from hydration parameters, it is unable to compare them with distilled model in this figure. However, the comparison shown in Table 8 confirms the unified form of distilled equation is better than the other models.

Figure 15 shows the generalization ability on test data (another type of cement differs from F_{01}, F_{31}, F_{38}, F_{51}. It reacts at 30 °C. The water-to-cement ratio is 0.55). For comparison, the coefficients of distilled unified form of kinetics were also determined from fits to experimental data to get the best matching. The PSO algorithm is used as optimization method with population size 1,22,880 and maximum generation 1000. The result using the best coefficients illustrates that the distilled unified form of kinetics performs well in fitting observed results even if sample's chemical composition, particle size and curing conditions are changed. However, the accuracy of kinetics using coefficients estimated by coefficient functions on test data is not as high as the results using the best coefficients, which reflects that although the unified form of kinetics is good in fitting observed results, this is ascribed to the complexity of problem and inadequacy of data, which is costly to collect. Therefore, the generalization ability of coefficient functions which are obtained in the second phase should be improved further.

5 Conclusions

In this paper, the middle-age hydration kinetic equation of Portland cement is distilled reversely from observed data. A phased hybrid evolution method with some reinforcement

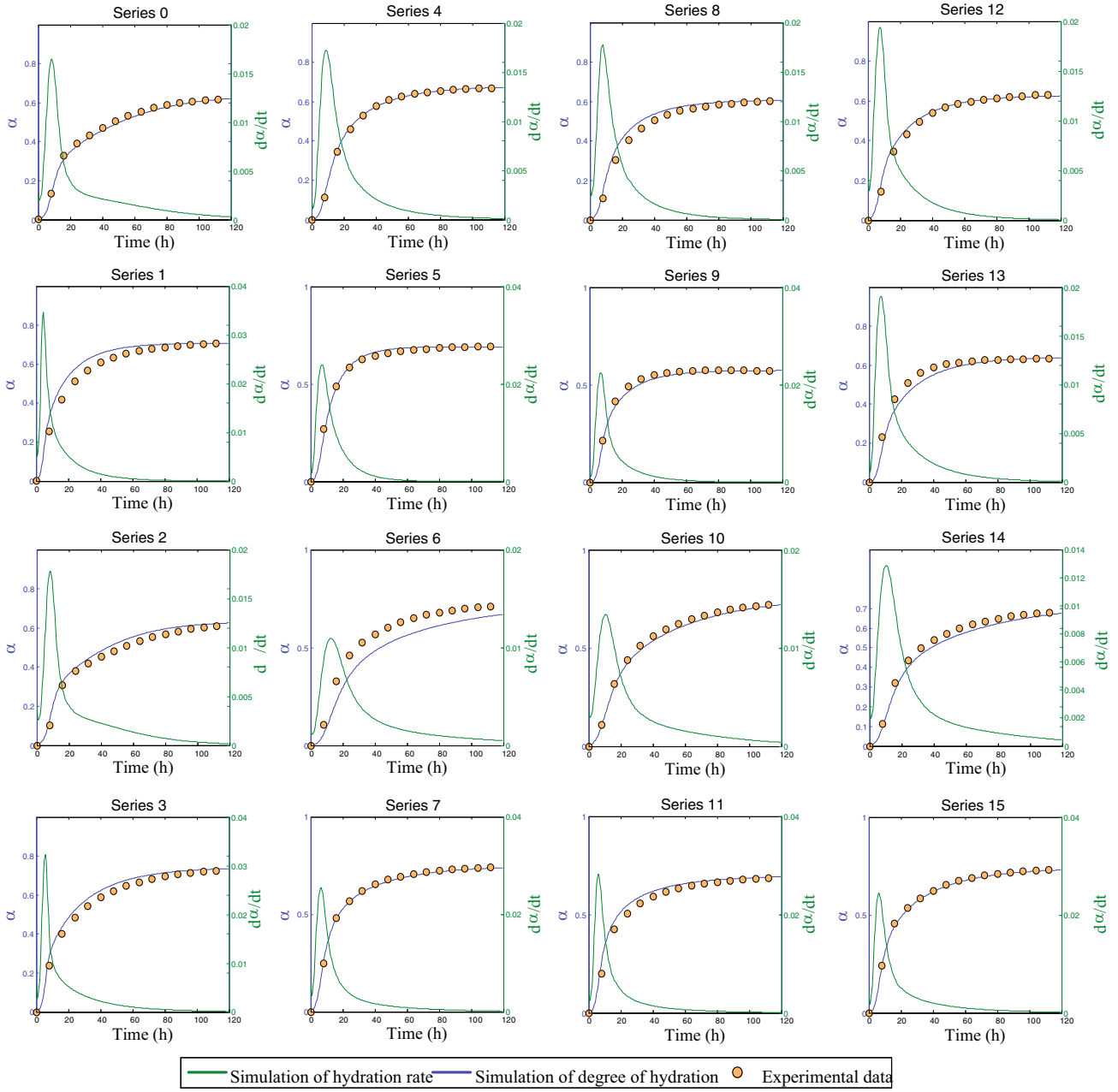

Fig. 14 Comparison of experimental and simulation curves

strategies and GPU acceleration is proposed to increase the diversity of searching and to speedup the evolution of kinetics. The middle-age hydration kinetics and its corresponding coefficient functions are distilled successfully. This is the first time that the middle-age hydration kinetics of Portland cement is distilled reversely from observed data. This is also the first time that the relationships between kinetics coefficients and cement hydration parameters are extracted and analyzed.

It takes a long time to distil the middle-age hydration kinetics. However, considering the frequency of execution of reverse extraction, the time consumption is acceptable because the objective of this research is hydration kinetics itself but not the system of reverse extraction. The impacts of change of value of coefficients on hydration rate and degree of hydration simulated by distilled kinetics are also evaluated. Compared with the manually derived models and early-age model, the nature of reverse distilling method makes the discovered model especially accurate in middle-age hydration. The generalization ability on test data proves that the distilled unified form of kinetics performs well in fitting observed results even if sample's chemical composition, particle size

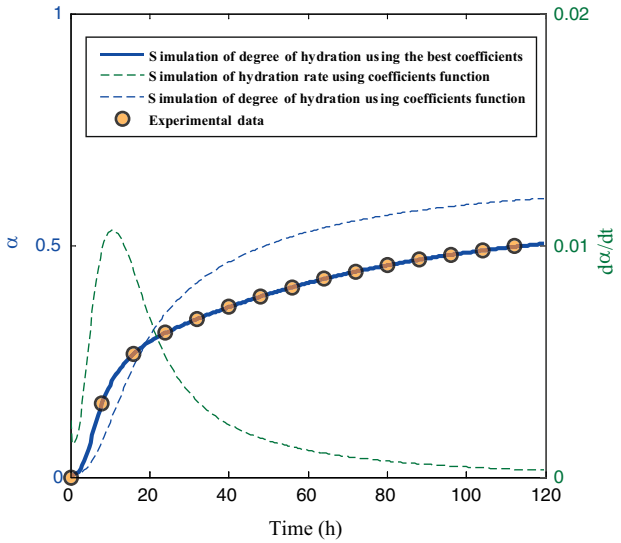

Fig. 15 Comparison of experimental and simulation curves on test data

and curing conditions are changed. However, the accuracy of kinetics using coefficients estimated by coefficient functions on test data is not as high as the results using the best coefficients, which reflects the high complexity of problem and inadequacy of data, which is costly to collect. The generalization ability of coefficient functions which are obtained in the second phase should be improved further.

In future works, considering the unsatisfactory generalization ability of obtained coefficient functions, additional data using more types of cement specimens and curing conditions are also required to improve the coefficient functions. Furthermore, the reverse extraction method needs to be improved to reduce its time complexity so that the framework of this method can be widely used in different systems. Its parameters' impact on different systems should also be studied.

Acknowledgments This work was supported by National Natural Science Foundation of China under Grant No. 61203105, No. 61173078, No. 61373054, No. 61173079, No. 81301298, No.61302128. Shandong Provincial Natural Science Foundation, China, under Grant No. ZR2012FQ016, No. ZR2012FM010. National Key Technology Research and Development Program of the Ministry of Science and Technology under Grant 2012BAF12B07-3.

References

Anderson C (2008) The End of Theory: The Data Deluge Makes the Scientific Method Obsolete. http://www.wired.com/science/discoveries/magazine/16-07/pb_theory/

Avrami M (1939) Kinetics of phase change. J Chem Phys 7:1103–1112

Bae H, Jeon TR, Kim S, Han SS (2010) Modified genetic programming combining with particle swarm optimization and performance criterion in solar cell fabrication. Int J Control Autom Syst 8:841–849

Bentz DP (2011) Critical observations for the evaluation of cement hydration models. Int J Adv Eng Sci Appl Math 2:75–82

Bogue RH (1955) The chemistry of Portland cement. Reinhold Publishing Corporation, pp 245–268

Box GEP, Hunter JS, Hunter WG (2005) Statistics for experiments: design, innovation, and discovery, 2nd edn. Wiley, New York

Chen L (2005) Optimal design for machinery: genetic algorithm. Machinery Industry Press (in Chinese), Beijing

CUDA (2009) CUDA programming guide version 2.3.1, NVIDIA

Du X, Ding L, Jia L (2008) Asynchronous distributed parallel gene expression programming based on estimation of distribution algorithm. In: Proceedings 4th international conference national computing, pp 433–437

Dabic P, Krstulovic R, Rusic D (2000) A new approach in mathematical modelling of cement hydration development. Cem Concr Res 30:1017–1021

Fan WG, Gordon MD, Pathak P (2004) Discovery of context-specific ranking functions for effective information retrieval using genetic programming. IEEE Trans Knowl Data Eng 16:523–527

Ferreira C (2001) Gene expression programming: a new adaptive algorithm for solving problems. Comput Syst 13:87–129

Floares AG (2007) Automatic reverse engineering algorithm for drug gene regulating networks. In: Proceedings of 11th Iasted International Conference Artificial Intelligence Soft Computing, pp 238–243

Guan W, Szeto KY (2013) Topological effects on the performance of island model of parallel genetic algorithm. In: Proceedings international work conference artificial neural network, pp 11–19

Holland JH (1975) Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence, second edn. MIT Press, University of Michigan Press, p 1992

Huang Z, Lu X, Duan H (2012) A task operation model for resource allocation optimization in business process management. IEEE Trans Syst Man Cybern Part A Syst Hum 42:1256–1270

Iba H (2008) Inference of differential equation models by genetic programming. Inf Sci 178:4453–4468

Johnson WA, Mehl RF (1939) Reaction kinetics in processes of nucleation and growth. Trans Am Inst Min Metall Eng 135:416

Kennedy J, Eberhart RC (1995) A new optimizer using particle swarm theory. In: Proceedings of 6th international symposium micro machine human science, pp 39–43

Kondo R, Kodama M (1967) On the hydration kinetics of cement. Semento Gijutsu Nenpo 21:77–82 (in Japanese)

Koza JR (1992) Genetic programming: on the programming of computers by means of natural selection. MIT Press, Cambridge

Krogmann K, Kuperberg M, Reussner R (2010) Using genetic search for reverse engineering of parametric behavior models for performance prediction. IEEE Trans Softw Eng 36:865–877

Krstulovic R, Dabic P (2000) A conceptual model of the cement hydration process. Cem Concr Res 30:693–698

Lin F, Meyer C (2009) Hydration kinetics modeling of Portland cement considering the effects of curing temperature and applied pressure. Cem Concr Res 39:255–265

Luo GH, Huang SK, Chang YS, Yuan SM (2014) A parallel Bees Algorithm implementation on GPU. J Syst Archit 60:271–279

Oltean M, Dumitrescu D (2002) Multi expression programming. Technical report, UBB-01-2002, Babes-Bolyai University

Park KB, Noguchib T, Plawsky J (2005) Modeling of hydration reactions using neural networks to predict the average properties of cement paste. Cem Concr Res 35:1676–1684

Pignat C, Scrivener P, Navi K (2005) Simulation of cement paste microstructure hydration, pore space characterization and permeability determination. Mater Struct 38:459–466

Qian L, Wang H, Dougherty ER (2008) Inference of noisy nonlinear differential equation models for gene regulatory networks using genetic programming and Kalman filtering. IEEE Trans Signal Process 56:3327–3339

Robilliard D, Poty VM, Fonlupt C (2009) Genetic programming on graphics processing units. Genet Program Evol Mach 10:447–471

Sanz SS, Roldan FC, Heneghan C, Yao X (2007) Evolutionary design of digital filters with application to subband coding and data transmission. IEEE Trans Signal Process 55:1193–1203

Schindler AK, Folliard KJ (2005) Heat of hydration models for cementitious materials. ACI Mater J 102:24–33

Schmidt M, Lipson H (2009) Distilling free-form natural laws from experimental data. Science 324:81–85

Thomas JJ, Biernacki JJ, Bullard JW, Bishnoi S, Dolado JS, Scherer GW, Luttge A (2011) Modeling and simulation of cement hydration kinetics and microstructure development. Cem Concr Res 41:1257–1278

Tomosawa F (1997) Development of a kinetic model for hydration of cement. In: Proceedings of tenth international congress chemistry of cement, pp 2ii051

Vega FF, Gil GG, Pulido JAG, Guisado JL (2004) Control of bloat in genetic programming by means of the island model. In: Proceedings of international conference parallel prob solv nat, pp 263–271

Wang L, Yang B, Zhao XY, Chen YH, Chang J (2010) Reverse extraction of early-age hydration kinetic equation from observed data of Portland cement. Sci China Technol Sci 53:1540–1553

Wang L, Yang B, Chen YH, Zhao XY, Chang J, Wang HY (2012) Modeling early-age hydration kinetics of Portland cement using flexible neural tree. Neural Comput Appl 21:877–889

Wang L, Yang B, Chen YH, Zhao XY (2012) Predict the hydration of Portland cement using differential evolution. In: Proceedings of IEEE congress evolution computer, pp 3388–3392

Wang PM, Feng SX, Liu XP (2005) Research approaches of cement hydration degree and their development. J Build Mater 8:646–652 (in Chinese)

Yan LP, Zeng JC (2006) Using particle swarm optimization and genetic programming to evolve classification rules. In: Proceedings of 6th world congress intelligence control automation, pp 3415–3419

Yang ZY, Li XL, Bowers CP, Schnier T, Tang K, Yao X (2012) An efficient evolutionary approach to parameter identification in a building thermal model. IEEE Trans Syst Man Cybern Part C Appl Rev 42:957–969

Zhang S, He Z (2009) Implementation of parallel genetic algorithm based on CUDA. In: Proceedings of international symposium intelligence computer application, pp 24–30

Zheng L, Lu Y, Guo M, Guo S, Xu CZ (2014) Architecture-based design and optimization of genetic algorithms on multi- and many-core systems. Future Gener Comput Syst 38:75–91

Zhou Y, Tan Y (2009) GPU-based Parallel Particle Swarm Optimization. In: Proceedings of IEEE congress evolution computer, pp 1493–1500