SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

November 2012 Vol. 55 No. 11: 2508–2520
doi: 10.1007/s11432-012-4646-z

Optimized statistical analysis of software trustworthiness attributes

ZHANG Xiao[1*], LI Wei[2], ZHENG ZhiMing[1,2*] & GUO BingHui[1]

[1]*Key Laboratory of Mathematics, Informatics and Behavioral Semantics, Ministry of Education, Beihang University, Beijing* 100191*, China;*
[2]*State Key Laboratory of Software Development Environment, Beihang University, Beijing* 100191*, China*

Received January 5, 2012; accepted April 25, 2012

Abstract Software trustworthiness has become one of the key restrictions for software service quality and the development of the software industry. However, trustworthiness attributes interlace structured and dynamical coupling relations, which causes great barriers for trustworthiness measurements of large-scale software. According to the dynamical evolutionary characteristics of software trustworthiness attributes, this paper proposes a new approach for optimizing the trustworthiness measurement in terms the kernel trustworthiness attributes, and improves a downsize-optimized statistical analysis method for software trustworthiness attributes based on their nonlinear relations. The improved method considerably simplifies the trustworthiness assessment of large-scale software. Using theoretical analysis and numerical simulations, the feasibility of this method is verified using two typical examples that illustrate the realization of the trustworthiness measurement.

Keywords software trustworthiness, kernel attributes, optimization, statistical analysis

Citation Zhang X, Li W, Zheng Z M, et al. Optimized statistical analysis of software trustworthiness attributes. Sci China Inf Sci, 2012, 55: 2508–2520, doi: 10.1007/s11432-012-4646-z

1 Introduction

As information transmission environments become more open and dynamic, research on the trustworthiness of large-scale software has become increasingly crucial for the development and application of modern software technology. Since the 1980s, IBM, Intel, Microsoft and other leading occidental enterprises and institutions have launched correlative research on trusted computing to establish trustworthiness criteria and theoretical platforms, such as TPM, palladium and open trusted computing [1–4]. Software trustworthiness is considered to be an integrated demonstration and assessment of various attributes that affect trusted software evolution, such as reliability, safety and security, correctness, timeliness, availability, and maintainability [5–7]. At present, enhanced functionality and comprehensive use of large-scale software have promoted trustworthiness attribute based research at the frontiers of information science and IT industry [8,9].

Trustworthiness attributes of large-scale software interlace structured and dynamical coupling relations. In particular, the nonlinear relations lead to extremely high trustworthiness complexity. Take aircraft

* Corresponding author (email: zhangxiao@ss.buaa.edu.cn, zzheng@pku.edu.cn)

control software and power grid detection software as examples. The length of the code in such software ranges from hundreds of thousands to millions, and in general there are over 100 functions and multiple layers of structure provided [10–12]. Currently, the majority of research on trustworthiness modeling and measurement uses discrete methods, including the stratified estimate method, the sub-attributes modeling method, and the key-redundancy degree method [13–15]. These methods judge trustworthiness by separate analysis or simple synthesis of the subsections of the entire software life cycle. As a software system evolves dynamically, certain kinds of behavioral characteristics and relationships of trustworthiness attributes appear to be dynamic, stochastic and nonlinear, which may lead to extremely high complexity of software trustworthiness whose analysis is beyond the applicability of the existing static and discrete methods.

In recent years, a number of dynamic continuous models and statistical analysis methods for software systems have been developed (see, e.g., [16,17]). For trustworthiness of large-scale software [18,19], dynamical evolution models were initially studied by using the characteristics of the evolution process during the whole software life cycle, and exact statistical analysis methods, dynamical quantitative indices and assessment mechanisms for software trustworthiness have since been developed.

Although there are generic evolution models and trustworthiness measurement methods, trustworthiness assessment for large-scale software is still nontrivial in practical applications. This is because all of the attributes need to be calculated and analyzed in the measurement process. Based on the dynamical evolutionary characteristics of software trustworthiness attributes (STAs) and using nonlinear relations between large-scale STAs, this paper proposes a new approach for optimizing trustworthiness measurement in terms of the kernel trustworthiness attributes, and establishes a downsize-optimized statistical analysis method for STAs, which improves on the invariant measurement method [18,19]. Our method considerably simplifies the trustworthiness assessment of large-scale software.

2 Characters of kernel trustworthiness attributes

Software trustworthiness often involves a great number of attributes. This is true, for instance, for large-scale software systems dealing with enormous amounts of data, complex functional components, frequent transfer of resources and high performance standards. Extensive dynamical interactions among STAs may lead to structural complexity and nonlinear coupling relations, thus increasing the evolution complexity of software systems. Therefore, it becomes difficult to use current measurement methods for involved complexity analysis. For example, in the invariant-measure based statistical analysis method [19], trustworthiness assessment relies on the calculation of the average return frequency:

$$\mu_F = \lim_{n\to\infty} \frac{1}{n^m} \sum_{v_1=1}^{n} \cdots \sum_{v_m=1}^{n} \delta_{\xi_{(v_1,\ldots,v_m)}}(x_1,\ldots,x_m). \tag{1}$$

For this method, the multiple sum of characteristic functions and the weak limit need to be calculated, so the level of computational complexity is exponential with respect to the number of attributes [20]. When there are more than two attributes, no direct method is available for STA measurement.

As software is an integrated body of complex systems, different attributes may exhibit different evolution behaviors and lead to different degrees of trustworthiness complexity during the software life cycle. As STAs expand, interact and develop constantly, their trustworthiness complexity displays convergence towards some key representative attributes. For large-scale software systems, many experimental results and much theoretical research indicate that a few special attributes occupy the majority of trustworthiness complexity [21,22]. Such attributes, which have high behavioral complexity, are more sensitive to environmental interference, and play a definitive role in the limit of trustworthiness evolution, are called kernel trustworthiness attributes (KTA). They are characterized as follows.

1) Active evolutions. KTAs cause the greatest trustworthiness complexity in forms of various patterns of dynamic behavior. Since the function states and performance states of software systems keep changing in different life stages, KTAs have corresponding complexity and multiple dynamical activities compared with the stable states of other attributes.

2) Sensitive to interference. As dynamical and open environments are an important trustworthiness factor, KTAs can adequately respond to internal or external interference. Especially when the intensity of interference is in the critical range, the stability of the evolution will fluctuate sharply to adapt.

3) Complex coupling relations. KTAs lead to a large proportion of nonlinear relations, which are the main cause of the increase in evolution complexity of software. Due to the dynamical nonlinear relations, the evolution complexity will inevitably converge to the KTAs marked with strong nonlinearity.

4) Declaration ability. According to dynamical system theory [21,22], the variation of KTAs is consistent with the evolution trends of the whole system, and therefore the behavior of KTAs determines the limiting states of software trustworthiness evolutions.

The characteristics discussed above suggest that KTAs may be distinct from other attributes owing to distinguishing trustworthiness behaviors. In addition, dynamic research shows that the evolution complexity of systems will converge to a few scales of attribute directions [21,22]. For the existence of complexity convergence characteristics, some dynamic research results, such as the theory of central manifolds and normal forms, solve the vector field in some lower-dimensional form that is as simple as possible [23–25]. Therefore, KTA extraction is available, and the ideal for optimizing trustworthiness models based on KTAs that follows could effectively decrease the scale of trustworthiness models and considerably reduce the complexity of structure and nonlinear relations in trustworthiness assessment. To obtain consistent trustworthiness along with complexity reduction, the dynamic kernel characteristics should be preserved during KTA extraction. Consequently, we propose an effective optimization method for trustworthiness measurement based on rational KTA extraction.

3 Optimization method based on software evolution and KTAs

Finding KTAs explicitly is a key step for optimization, but there are few effective methods to achieve this. KTAs determined by the internal and external factors are the inner or natural aspects of software systems. Hence, in this paper we will extract KTAs based on software evolution characteristics, and optimize the statistical analysis method from a dynamical point of view, using analysis tools for complex behavior such as higher dimensional and non-hyperbolic dynamical systems.

Let $\dot{x} = F(x; a)$ be the trustworthiness evolution model of a large-scale software system with $x \in \mathbb{R}^n$ and $a \in \mathbb{R}^m$. According to the construction principle for trustworthiness evolution models [18], x reflects the STAs which are internal factors, and a reflects external factors such as human interaction and survival. The dimension n is the number of STAs under consideration, and m is the number of parameters. Thus, the life cycle of software trustworthiness evolution can be represented by the behavior of the dynamical system F.

Step 1. Optimization by FR-Map.

By the analysis in Section 2, KTAs possess the characteristics of active evolution, sensitivity to interference and complex coupling relations. Hence the stability of model $\dot{x} = F(x; a)$ will fluctuate sharply in the corresponding directions, which suggests an approach to extracting KTAs by stability analysis. In general, stability analysis of dynamical systems starts with analysis around critical points [25]. Let $\varphi_a(t; x_0)$ be a solution of $\dot{x} = F(x; a)$ with x_0 as the initial state. A point x^* is called a critical point if $\varphi_a(t; x^*) = x^*$ for some t. The set L of all critical points is called a critical set. If there exists some T which satisfies $\varphi_a(T; x^*) = x^*$, $\varphi_a(t; x^*) \neq x^*$ for $0 < t < T$, the point $x^*(\in L)$ is called a periodic point of period T. The set of all periodic points $\{\varphi_a(t; x^*) : 0 \leqslant t \leqslant T\}$ is called a periodic orbit. In particular, a point of period 1 is called a singularity, for which all the STAs remain balanced and software evolves in the same state. Given the system $\dot{x} = F(x; a)$, we can obtain the critical set according to the procedure above.

As software keeps developing in the life cycle and complex coupling relations are important characteristics of KTAs, global stability analysis is necessary. This means that to extract KTAs exactly, we should analyze not only the stability around the critical set but also the dynamic behavior of orbits among critical points. Nevertheless, global stability analysis is not trivial, and analysis methods are often

used to solve the global stability problem of a certain system. Thus, we cannot directly analyze the trustworthiness of software. Considering the complexity and difficulty of global stability analysis, our method tries to find some regularity in the global system evolution instead of solving the entire global stability problem. By some special constructions under certain evolution regularity assumptions, we can remove some non-KTAs and consequently gain an optimized system.

As periodicity is an obvious form of evolution regularity, we take trustworthiness models that include periodic orbits as examples to explain our ideal optimization. Firstly, we introduce the definitions of a transversal and an FR-Map as optimization tools.

Consider a trustworthiness evolution model $\dot{x} = F(x; a)$, and a state point x^* for which one of the coordinate functions $F_k(x^*; a) \neq 0$, with $1 \leqslant k \leqslant n$. The hyperplane through x^* formed by setting the kth coordinate equal to a constant,

$$\Omega = \{x : x_k = x_k^*\} \tag{2}$$

is called a transversal, because trajectories cross it around x^*.

Let $\varphi_a(t; x_0)$ be a solution of $\dot{x} = F(x; a)$ with x_0 as the initial state. Assume that $\varphi_a(\tau^*; x^*)$ is in a transversal Ω for some $\tau^* > 0$, and also assume that there are no other intersections of $\varphi_a(t; x^*)$ with Ω near x^*. For x near x^*, there is a nearby time $\tau(x)$ such that $\varphi_a(\tau(x); x)$ is in Ω. Then

$$P(x) = \varphi_a(\tau(x); x) \tag{3}$$

is called the first return map (FR-Map).

Assume that x^* is on a periodic orbit of period T. Once a transversal is constructed through x^*, orbits which start near x^* will come back to the transversal because of the periodicity of x^*. Thus, the definition of the FR-Map makes sense. We now focus on the relationship between the original orbit and the FR-Map from the aspect of stability.

Theorem 1. Consider a trustworthiness evolution model $\dot{x} = F(x; a)$, $x \in \mathbb{R}^n$ and denote by $\varphi_a(t; x_0)$a solution of $\dot{x} = F(x; a)$ with x_0 as the initial state. Assume that x^* is on a periodic orbit of period T. If there is a transversal though x^*, then the n eigenvalues of $D_x\varphi_a(T; X^*)$ consist of the $n-1$ eigenvalues of $\mathrm{DP}(x^*)$, together with 1, where $P(x) = \varphi_a(\tau(x); x)$ is the FR-Map on Ω.

Proof. Assume that the FR-Map $P(x)$ is formed for a time $\tau(x)$ such that $\varphi_a(\tau(x); x)$ is back in Ω.

It is obvious that

$$D_x\varphi_a(T; X^*)F(x^*; a) = F(\varphi_a(T; x^*)) = F(x^*; a). \tag{4}$$

Therefore, 1 is an eigenvalue for the eigenvector $F(x^*; a)$.

If we take a vector v lying in Ω, we have

$$\begin{aligned}\mathrm{DP}(x^*)v &= D_x\varphi_a(\tau(x^*); x^*)v + \left(\left.\frac{\partial\varphi_a}{\partial t}\right|_{(\tau(x^*);x^*)}\right)\left(\left.\frac{\partial\tau}{\partial x}\right|_{x^*}\right)v \\ &= D_x\varphi_a(T; x^*)v + F(x^*; a)(D\tau(x^*))v.\end{aligned} \tag{5}$$

In the second term,

$$(D\tau(x^*))v = \left(\left.\frac{\partial\tau}{\partial x}\right|x^*\right)v \tag{6}$$

is a scalar multiplied by the vector field $F(x^*; a)$ and represents the change in time to return from Ω back to Ω. Thus, if we take a vector v lying in Ω, we get

$$D_x\varphi_a(T; x^*)v = \mathrm{DP}(x^*)v - F(x^*)(D\tau(x^*))v. \tag{7}$$

Therefore, using a basis of $F(x^*; a)$ and $n-1$ vectors along Ω, we have

$$D_x\varphi_a(T; x^*) = \begin{pmatrix} 1 & -D\tau(x^*) \\ 0 & \mathrm{DP}(x^*) \end{pmatrix} \tag{8}$$

and the eigenvalues are as stated in the theorem. Note that the eigenvalue 1 results from the periodicity of the orbit.

In dynamics research, eigenvalues are a useful approach to stability analysis. Judging from the absolute value of eigenvalues, we can consequently estimate the stability of evolution models [23,24]. For example, assume that x^* is on a periodic orbit of period T. Thus, if all the eigenvalues of $D_x\varphi_a(T;x^*)$ have absolute value less than one except for the eigenvalue equal to 1, then the orbit is asymptotically stable [26]. As shown in Theorem 1, the eigenvalues of $D_x\varphi_a(T;x^*)$ consist of 1 and all the eigenvalues of $\mathrm{DP}(x^*)$. This means that the stability of a periodic orbit is equal to that of the FR-Map.

It is obvious that the FR-Map is of $n-1$ dimensions while the original model is in n-dimensional space. By constructing the FR-Map and transversal, the periodic dimension is removed while the stability is preserved. Since we aim to extract KTAs without changing the dynamic characteristics of the evolution models, the method using the FR-Map and transversal construction is helpful when there is a periodic orbit.

In fact, periodicity is not a necessary condition for constructing proper FR-Maps and transversals for non-KTA elimination. The main function of periodicity is to take the orbit back into the transversal. Therefore, if there is a hyperplane which is a transversal for all the orbits passing through it, and all the orbits starting from an initial point cross it and come back to it, then the FR-map method makes sense for optimizing the trustworthiness model, no matter whether a periodic orbit exists. According to the discussion above, we propose a general method for KTA extraction and model reduction in the following form.

Theorem 2. Consider a trustworthiness evolution model $\dot{x} = F(x;a)$, $x \in \mathbb{R}^n$ and denote by $\varphi_a(t;x_0)$ a solution of with x_0 as the initial state. If there is a hyperplane Ω which is a transversal for all the orbits passing through it and all the orbits starting from an initial point cross it and come back to it, then an FR-Map can be defined that preserves the dynamic characteristics and stability of the original model F.

Furthermore, it is not difficult for the conditions on the hyperplane Ω in this theorem to be satisfied. Given a trustworthiness evolution model, we can construct a proper hyperplane either by phase graph analysis or numerical simulation. Consider, for example, the following popular model [21,22]:

$$\begin{cases} \dot{x} = y, \\ \dot{y} = -x^2 - x^3. \end{cases} \tag{9}$$

We can draw the phase graph (shown in Figure 1) by local stability analysis around the critical set, and consequently find that $\Omega = \{x = -1\}$ is the proper hyper plane for orbits starting from any initial point no matter whether the orbits are near periodic orbits.

Furthermore, we can simulate orbits starting with different initial points as shown in Figure 2, where orbits starting with different initial points are drawn in distinct colors. It is obvious that we have much more than a single choice for the proper transversal construction.

Thus, for the majority of models, trustworthiness analysis of the n-dimensional model F can be turned into the assessment of the $(n-1)$-dimensional FR-Map by the optimization method proposed above.

Step 2. Optimization by partial differential operator.

The contribution of Step 1 to optimization is indispensable but limited. Considering the computational complexity of current trustworthiness assessment methods, the one-dimensional downsize operation is far from the optimization demands for large-scale software. Nevertheless, the optimization in Step 1 has turned the original differential model, which is continuous in time t, into a function that iterates in discrete time. This considerably decreases the complexity of the analysis and leads to an approach for further optimization by discrete systems research.

According to the discussion in the above section, to extract KTAs exactly we should carry out global stability analysis and focus on the domain where stability of the system varies sharply. As global stability analysis is also incalculable for most models in discrete time, we only try to distinguish some non-KTAs from the STAs rather than identifying every KTA.

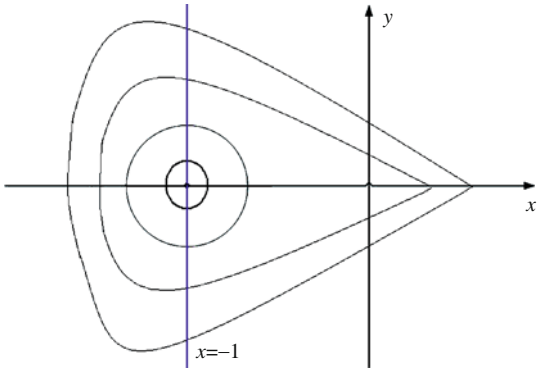

Figure 1 Phase graph of Eq. (9) around the critical set and corresponding transversal.

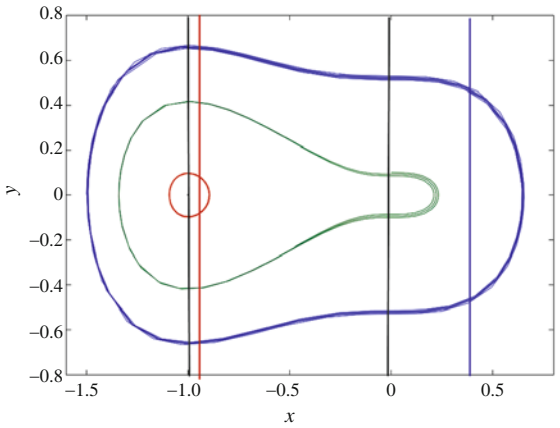

Figure 2 Orbits starting with different initial points (drawn in distinct colors red, green and blue).

In discrete time dynamics research, partial differential operators are useful tools for stability research near singularities, and we discuss some important stability criteria below. Learning from existing stability analysis tools based on partial differential operators, we generalize part of the stability criteria to fulfill the global condition, and accordingly achieve optimization of the FR-Map derived in Step 1.

Let P be a nonlinear map from $\mathbb{R}^n$ to $\mathbb{R}^n$ with coordinate functions P_i. The partial differential operator at a point $\bar{x}$ is the $n \times n$ matrix

$$\mathbf{DP}(\bar{x}) = \left(\frac{\partial P_i}{\partial x_j}(\bar{x})\right) = \begin{pmatrix} \frac{\partial P_1}{\partial x_1}(\bar{x}) & \cdots & \frac{\partial P_1}{\partial x_n}(\bar{x}) \\ \vdots & & \vdots \\ \frac{\partial P_n}{\partial x_1}(\bar{x}) & \cdots & \frac{\partial P_n}{\partial x_n}(\bar{x}) \end{pmatrix}. \tag{10}$$

Each row corresponds to a coordinate function and each column to the variable used for calculation.

For a nonlinear map with a singularity or periodic point $\bar{x}$, the stability criterion is illustrated assuming that all the eigenvalues λ_j of $D(P^k)(\bar{x})$ satisfy $|\lambda_j| < 1$, so that the orbit through $\bar{x}$ is an attractor.

To distinguish the non-KTAs, the dynamically stable directions that are dissimilar from that of the KTAs should be removed. Referring to the local criterion for periodic points, we propose a global one for stability determination in a certain direction.

Theorem 3. Consider an FR-Map P from $\mathbb{R}^n$ to $\mathbb{R}^n$ of a trustworthiness evolution model. Define the directional differential operator as an n-dimensional vector

$$\mathbf{DP}_k(x) = \left(\frac{\partial P_i}{\partial x_k}\right) = \left(\frac{\partial P_1}{\partial x_k}, \ldots, \frac{\partial P_n}{\partial x_k}\right)^{\mathrm{T}}. \tag{11}$$

If all the components of the directional differential operator satisfy $|\partial P_i/\partial x_k| < 1$, $1 \leqslant i \leqslant n$ for any point x on the transversal, then the FR-Map is stable in the kth direction.

As this theorem can be proved directly from the definition of the differential operator, we omit a detailed proof.

Using Theorem 3, the stability of the FR-Map can be tested in each direction, and all the stable directions will be distinguished fewer than n times. If all the directions are stable, then the evolution of the original model is stable, and trustworthiness of this software can be predicted and measured. We can then declare that the software is trustable and it is not necessary to analyze its STA by a statistical method or to optimize the KTAs. In addition, since these stable directions may be correlated, linearly independent directions should be selected.

Compared with the characteristics of KTAs, the linearly independent stable directions indicate non-KTAs. Removing these directions from the FR-Map, the partial differential operator based method considerably decreases the dimension of the trustworthiness models.

Step 3. Statistical analysis of optimized model.

KTAs of large-scale software can be accurately extracted or confined in an area shrunk by the FR-Map and partial differential operator based criterion, thus using theory to shed light on applications of computer science. A downsize model can then be constructed based on the extracted KTAs, which obviously decreases the dimensions and the analysis complexity.

The reduced simplest system can be viewed as an optimized new trustworthiness evolution model while its trustworthiness behavior is the same with the initial one. The trustworthiness of the new models can be assessed using current statistical analysis methods. As the dimensions of the trustworthiness model have been greatly decreased, statistical analysis methods that are infeasible for large-scale software may be of use.

Take the invariant-measure based statistical analysis method [19] as an example. When the number of STAs is three or more, the method is inefficacious as the computational complexity is huge. When there are fewer than three attributes in the new model, trustworthiness can be assessed by calculating the average return frequency μ_F in Eq. (1), with the computation only involving two variables, a double sum and a planar weak limit at most. Furthermore, the level of computational complexity decreases exponentially.

4 Optimized statistical analysis of trustworthiness for large-scale software

In this section, by modeling two typical software trustworthiness examples, we study the optimized method for statistical analysis of the trustworthiness of the models and compare the computational complexity of the KTA extraction based method with the current result.

4.1 An example for general optimized analysis of trustworthiness

Consider an example of the trustworthiness model for software running in a Linux environment. In this condition, the amount of thread, memory consumption, and CPU occupation are the most important state quantities, which are the attributes for analyzing and predicting the software trustworthiness. Denote by $x(t)(\geqslant 0)$ the thread at time t, by $y(t)(\in [0, D])$ the memory consumption at time t, and by D the upper bound on the physical memory. Also, denote by $z(t)(\in [0, 1])$ the CPU occupation at time t.

When the software is running separately on the system, it will perform its calculation and storage function by setting up new thread to allocate memory and CPU resources. As the task proceeds towards finishing, the attributes showing the task and resource situation will return to their original state at their own speed. Assume that the relative amounts of those attributes (i.e., $x\prime(t)/x(t)$) are respectively same at different times and that the ratios are denoted by the parameters a, b and c.

When there is other software running collaterally or the system is attacked by a virus or malicious code, the system will postpone some tasks to conserve resources. Obviously, some threads will be in wait and three attributes will interact and restrict each other: the more threads there are, the more tasks should be treated, so more memory and CPU resources are needed; simultaneously, the more memory is consumed, the slower is the performance speed, so more threads will be in wait and the amount of thread and CPU will increase with the memory; meanwhile, the CPU occupation will almost or already achieve the capacity load when there is some thread in wait, so changes to the CPU will play a small part in the variety of thread; however, the memory consumption is more sensitive with greater CPU occupation, so the more CPU that remains, the more thread that is in wait can be processed, and thus the memory increases. Referring to the memory management model for the Linux system [26], the superposition of attribute relations is respected by multiplication. Denote by m, n and w the intensity of effect caused by the attribute relations, and assume them to be constant. Set the initial state as $(\bar{x}, \bar{y}, \bar{z})$. From the discussion above, the nonlinear relationship between the attributes can be represented by the following equations:

$$\begin{cases} \dot{x}(t) = -a(x(t) - \bar{x}) + m(y(t) - \bar{y}), \\ \dot{y}(t) = -b(y(t) - \bar{y}) + n(x(t) - \bar{x})(1 - z(t)), \\ \dot{z}(t) = -c(z(t) - \bar{z}) + w(x(t) - \bar{x})(y(t) - \bar{y}). \end{cases} \tag{12}$$

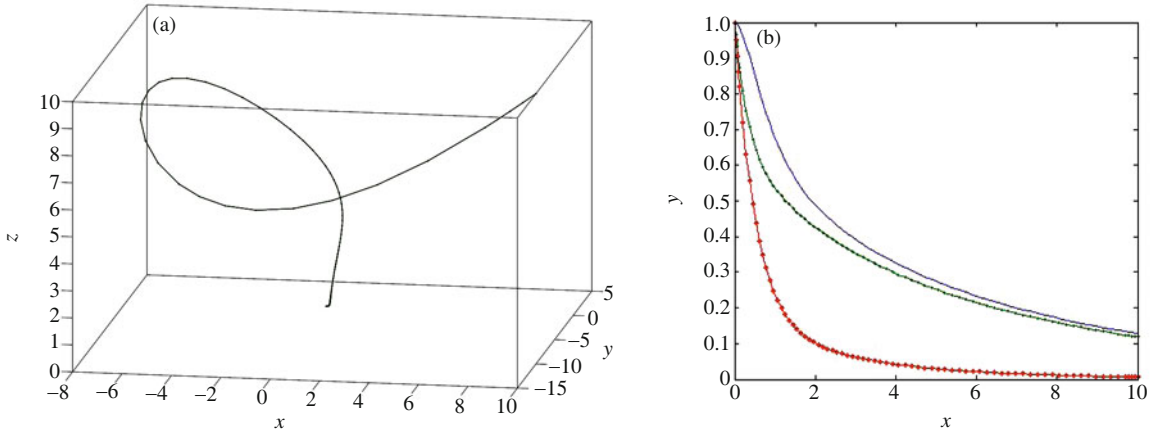

Figure 3 Numerical solution of equation set (13) for $p < 1 (a = 10, p = 0.8, c = 3)$. (a) Phase portrait; (b) STA evolution.

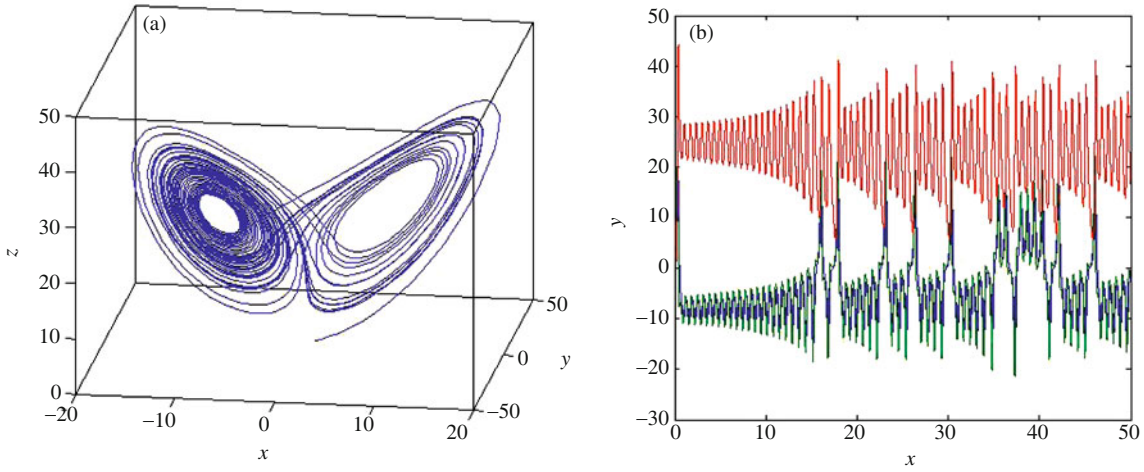

Figure 4 Numerical solution of equation set (13) for $p > 1 (a = 10, p = 26, c = 3)$. (a) Phase portrait; (b) STA evolution.

We will assess and analyze the trustworthiness of model (12) using the KTA based optimization method proposed in this paper.

By simple linear transformation, Eq. (12) reduces to

$$\begin{cases} \dot{X}(t) = -aX(t) + aY(t), \\ \dot{Y}(t) = pX(t) - bY(t) - X(t)Z(t), \\ \dot{Z}(t) = -cZ(t) + lX(t)Y(t), \end{cases} \tag{13}$$

where $a > 0$, $b > 0$, $c > 0$, $p = (mn/a) > 0$ and $l = (w/n) > 0$. $X(0)$, $Y(0)$ and $Z(0)$ all equal zero in equation set (13), and the system is symmetric from the viewpoint of the Z-axis.

To extract KTAs, we start with stability analysis around critical points. When $0 < p < 1$, equation set (13) has only one singularity at $(0, 0, 0)$, with corresponding eigenvalues $\lambda = (\lambda_1, \lambda_2, \lambda_3)$ each of which is negative. Thus, the whole solution space is stable according to the discussion in Step 2 and thus the system is stable (as shown in Figure 3), which means that the software system is trustable [1,18].

When $p > 1$, the two new critical points $(\pm\sqrt{l^{-1}c(p-1)}, \pm\sqrt{l^{-1}c(p-1)}, p-1)$ emerge alongside the original one, and the repeated evolution circles around the new critical points is obvious (Figure 4), which is similar to the condition in Theorem 2.

Actually, it can be proved that each orbit starting at the initial point will be pushed out along the X or Y direction and pulled back in the Z direction when it goes around the two new singularities. Hence the orbits will cross the region between the two new singularities repeatedly, which is consistent with the simulation result in Figure 4. Therefore, we can define a hyperplane as

$$\Sigma = \{(X, Y, Z) : |X|, |Y| \leqslant \alpha < l^{-1}C(p-1), Z = p-1\}. \tag{14}$$

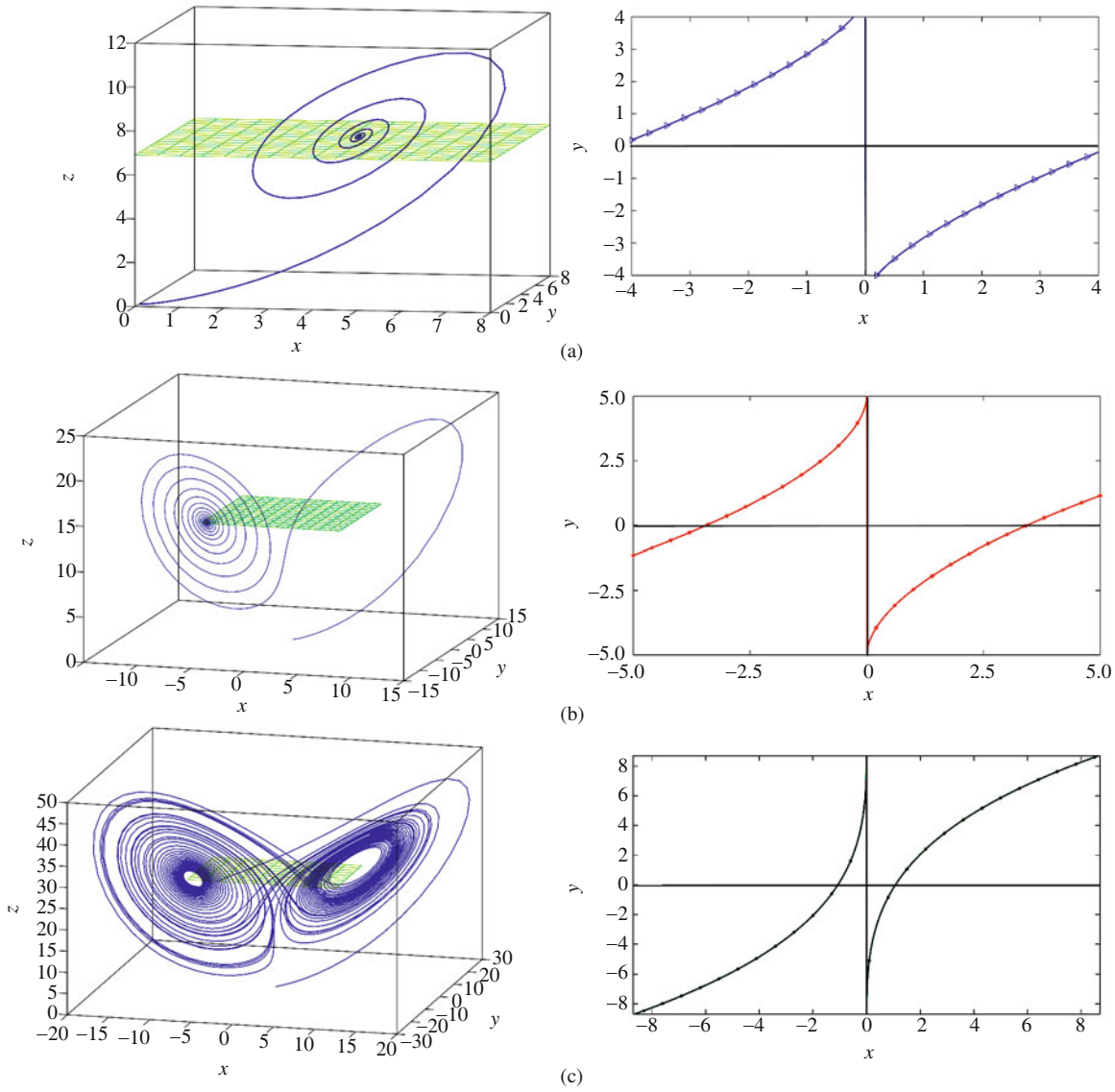

Figure 5 Optimal solution of equation set (3) and the corresponding optimized model with respect with the parameter p=8,15,28. (a) $p = 8$; (b) $p = 15$; (c) $p = 28$.

Thus, all the orbits starting at an initial point will cross Σ and return to it. Moreover, Σ is a transversal of the system. According to Theorem 2, we get the following FR-Map for equation set (13):

$$(f(u), g(u, v)) = \begin{cases} (-\alpha + (p-1)\beta u^{-\lambda_1/\lambda_3}, \beta u^{-\lambda_2/\lambda_3} v), & u > 0, \\ (\alpha - (p-1)\beta |u|^{-\lambda_1/\lambda_3}, \beta |u|^{-\lambda_2/\lambda_3} v), & u < 0, \end{cases} \tag{15}$$

where u is the direction between the two new critical points, v is orthogonal to u, and α and β are constants depending on the parameters of Eq. (13). Meanwhile, one of the vectors u or v is a KTA, or both are. The analysis of Eq. (15) is identical to that for Eq. (13) in the sense of trustworthiness assessment. However, the 3-dimensional model has been reduced to a 2-dimensional one.

Further, we now try to optimize Eq. (13) to a simpler form using Step 2 of our optimization method. As $f(u)$ (shown in Figure 5) is independent of v, so $\partial f(u)/\partial v = 0 < 1$. Together with $0 \leqslant \partial g(u, v)/\partial v < 1$, Eq. (15) converges strongly in direction v according to Theorem 3. This means that the v-direction is a non-KTA for Eq. (15). By removing the non-KTA direction u, we finally obtain the equation $f(u)$ as the simplest form and the variable u as the unique kernel trustworthiness attribute.

Remark 1. By the above discussion, the trustworthiness of the system corresponding to equation set

(13) is determined by a one-dimension function, which greatly reduces the complexity of the software trustworthiness assessment.

Remark 2. Notice that the direction u reflects the KTA and is the superposition of directions X and Y, which suggests that the software trustworthiness is greatly influenced when thread and memory consumption are in some proportional relationship.

4.2 An example for directly optimized analysis of trustworthiness

For special large-scale software in certain fields of application, the simplest result from optimized analysis of trustworthiness can be achieved directly using the method introduced in this paper. In this case, the characteristics of KTAs are usually more prominent than in general. Since our optimization method is based on the evolution characteristics of KTAs, we may reduce the dimension of the model directly, rather than using a dimension-by-dimension method. In the following, we study an example of a trustworthiness model for software that controls a power system, for which trustworthiness plays an important role. In this model of power-system controlling software, the high efficiency of our optimized method is obvious.

In the classical power-system model [27], we focus on three modules of software to control the transient stability of the power system: the difference in rotor angle δ between different electric generators, the difference in velocity ω between different electric generators, and the efficiency of the controllability σ. Consider a power system consisting of n electric generators with uniform damping only. The evolution model of the STA can be represented as follows:

$$\begin{cases} \delta_{in}(t_{k+1}) = \dfrac{1}{n}\omega_{in}(t_k), \\ \omega_{in}(t_{k+1}) = \dfrac{1}{M_i}(P_{mi} - P_{ei}) - \dfrac{1}{M_i}(P_{mn} - P_{en}) - \kappa\omega_{in}(t_k), \\ \sigma_{in}(t_{k+1}) = \dfrac{1}{2}\sigma_{in}(t_k) + q\omega_{in}(t)\delta_{in}(t_k), \end{cases} \tag{16}$$

where

$$P_{ei} = {E_i}^2 G_{ii} + \sum_{l=1, l\neq i}^{n} E_i E_l G_{il}\cos\delta_{il} + E_i E_l G_{il}\sin\delta_{il}. \tag{17}$$

κ is the damping coefficient; M_i is the inertia constant for generator i; δ_{in} is the difference in rotor angle between generator i and generator n; ω_{in} is the difference in velocity between generator i and generator n; q is the inverse of the greatest value of δ_{in}; P_{mi} is the output power of generator i; E_i is the inner potential of generator$i$; G_{ii} is the self-conductance of generator i; and G_{il} is the transsusceptance between generator i and generator l.

Since the model is given in the form of iterative equations, it is difficult to find the proper transversal for constructing an FR-Map for the differential model. In this case, we try to extract KTAs directly using Step 2 of the optimization method. It is easy to obtain the partial differential operator of equation set (16):

$$\mathbf{DF} = \left(\frac{\partial F_i}{\partial\delta_{in}}, \frac{\partial F_i}{\partial\omega_{in}}, \frac{\partial F_i}{\partial\sigma_{in}}\right) = \begin{pmatrix} 0 & \frac{1}{n} & 0 \\ P(\delta_{in}) & -\kappa & 0 \\ q\omega_{in} & q\delta_{in} & \frac{1}{2} \end{pmatrix}. \tag{18}$$

We then get the directional differential operators along ω_{in} and σ_{in} which satisfy $|\partial F_i/\partial\omega_{in}| < 1$, $|\partial F_i/\partial\sigma_{in}| < 1$ for $i = 1, 2, 3$, when the domain is in any state. According to Theorem 3, the model is stable along these two directions, so that δ_{in} is the unique direction that could indicate a KTA. Removing ω_{in} and σ_{in} from equation set (16), the optimized system corresponding to the classical power system model can be obtained as $\dot{\delta} = f_2(\delta)$, where f_2 is a characteristic function defined in Liu [27]. Consequently, the original model is directly reduced to a 1-dimensional form with δ_{in} as the variable.

Remark 3. By the above discussion, the trustworthiness of the system corresponding to equation set (16) is determined by a 1-dimensional function $f_2(\delta)$, which is demonstrated directly by the optimized

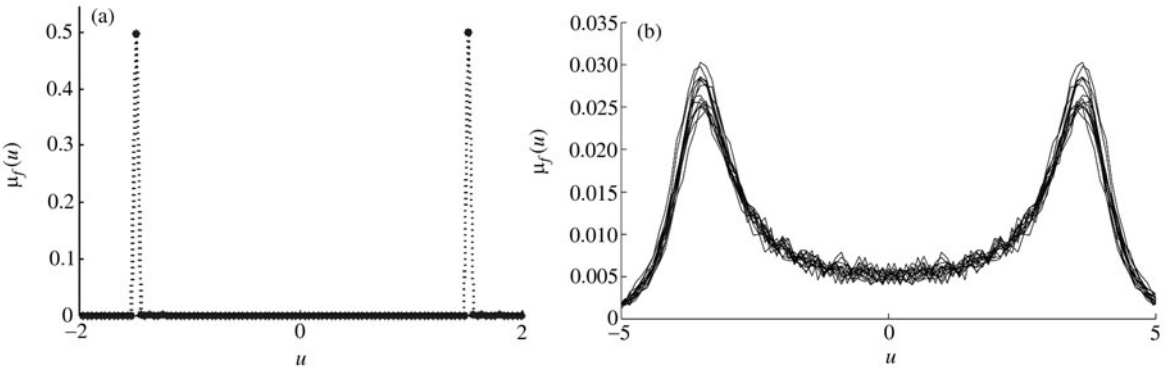

Figure 6 Numerical simulation of $\mu_f(u)$. (a) $p < 13$; (b) $p > 13$.

method and greatly reduces the complexity of the software trustworthiness assessment for the original model.

Remark 4. Notice that the direction δ reflects the KTA, and the software trustworthiness is greatly influenced when the parameter for the difference in rotor angle δ fluctuates considerably.

4.3 Statistical analysis of trustworthiness and comparison of complexity

Taking the model in Subsection 4.1 as an example, we study the reduced model by a statistical analysis method. For a 1-dimensional function $f(u)$, by introducing the Frobenius-Perron operator [28,29], $\mu_f(u)$ should satisfy

$$\mu_f(u) = \frac{m}{\beta(p-1)}\left(\frac{u+\alpha}{p-1}\right)^{m-1}\mu_f\left(\beta^{-1}\left(\frac{u+\alpha}{p-1}\right)^m\right) + \frac{m}{\beta(p-1)}\left(\frac{\alpha-u}{p-1}\right)^m\mu_f\left(-u_1^{-1}\left(\frac{\alpha-u}{p-1}\right)^m\right). \tag{19}$$

As shown in Figure 6, $\mu_f(u) = 0$ except for a finite set of points when $p < 13$, and thus the corresponding software system is trustable [19]. This verifies our theoretical result in Subsection 4.1. When $p > 13$, $\mu_f(u)$ is ergodic in a finite interval, so each attribute reaches all states in this interval with positive probability. During this time, the process is chaotic and completely unpredictable, which suggests that self-defense ability cannot resist security threats from the external environment such as viruses. That is, the software is extremely distrustful. Finally, quantitative statistical analysis of software systems with multiple attributes is realized by optimization.

In this section, we demonstrate the performance of the optimized statistical analysis method proposed in this paper through two typical examples. Each large-scale software trustworthiness model in the examples is successfully reduced to a low-dimensional form or even a simple 1-dimensional model, which decreases the complexity of trustworthiness assessment. Not only is the feasibility of both critical sets verified, but also the ability to alternately apply the two steps from the methods and directly reduce models into the simplest form are explicitly demonstrated.

To illustrate the decreased computational complexity for the optimized method, we calculate and compare the actual complexity of the optimized and original statistical methods. Consider the software with n STAs. To assess the improved method, both the computational complexity for optimization and statistical analysis should be taken into account. In the optimization, the computational complexity for solving the critical set is $O(n \log n)$. Furthermore, the logic consumption of the FR-Map construction in Step 1 and the directional differential operator based computation are $O(nm)$ and $O(n^2)$, where $m(< n)$ is the reduced dimension. In addition, the complexity of the invariant-measure based statistical analysis is $O(\mathrm{e}^n)$. As the optimized model is an m-dimensional system, the statistical analysis complexity for the reduced model is $O(\mathrm{e}^m)$. As a result, the total complexity of the improved trustworthiness statistical analysis method is $O(n(m+\log n)+m^2+\mathrm{e}^m)$. Considering the $O(\mathrm{e}^n)$ complexity of the original method, the reduction in computational complexity is significant and exponential with respect to the number of STAs.

5 Conclusion

As software is a class of complex systems with a life cycle, extensive dynamical interactions among the attributes leads to structural complexity and nonlinear coupling relations, which seriously aggravate the evolution complexity of software systems and increase the difficulty of calculations for the current trustworthiness measurement methods. According to the properties of KTAs, which perform as active evolutions, sensitivity to interference, complex coupling relations and declaration ability, an advanced optimized statistical analysis methodology is proposed using the nonlinear correlations between attributes, which greatly reduces the computational complexity of the original method. Taking two concrete trustworthiness models of software as typical examples, this paper studies the practicality of the optimized method and proposes a framework for calculating the trustworthiness of large-scale complex software systems.

Acknowledgements

This work was supported by International Cooperation Project of China (Grant No. 2010DFR00700), and Innovation Foundation of BUAA for PhD Graduates.

References

1 Kuhlmann D, Landfermanny R, Ramasamy H V. An Open Trusted Computing Architecture–Secure Virtual Machines Enabling User-Defined Policy Enforcement. IBM Reserch Report RZ3655. 2006
2 Pearson S, Balacheff B. Trusted Computing Platforms: Tcpa Technology in Context. New Jersey: Premtice Hall PTR Prees, 2003
3 Delaune S, Kremer S, Ryan M D, et al. A formal analysis of authentication in the TPM. LNCS, 2011, 6151: 350–365
4 Sadeghi A R. Trusted computing: special aspects and challenges. In: Proceedings of the 34th Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM'08). LNCS, Vol 4910. Heidelberg: Springer-Verlag Verlin, 2008. 98–117
5 ISO/IEC 11889-3. Information Technology-Trusted Platform Module-Part 3: Structures. 2009
6 Schmidt H. Trustworthy components compositionality and prediction. J Syst Software, 2003, 65: 215–225
7 Kirovski D, Drinic M, Potkonjak M. Enabling trusted software integrity. Oper Syst Rev, 2003, 36: 108–120
8 Berger B. Trusted computing group history. Inform Secur Tech Rep, 2005, 10: 59–62
9 Peter G N. Reflections on system trustworthiness. Adv Comput, 2007, 70: 269–310
10 Jitender K C, Yogesh S. Code and data spatial complexity: two important software understandability measures. Inform Software Tech, 2003, 45: 539–546
11 Erman C, Martha G. Software complexity and its impacts in embedded intelligent real-time systems. J Syst Software, 2005, 78: 128–145
12 Pate-Cornell E, Dillon R. Probabilistic risk analysis for the NASA space shuttle: a brief history and current work. Relab Eng Syst Safe, 2001, 74: 345–352
13 Jose L S, Ines H. An AHP-based methodology to rank critical success factors of executive information systems. Comp Stand Inter, 2005, 28: 1–12
14 Littlewood W, Wright D. The use of multilegged arguments to increase confidence in safety claims for software-based systems: a study based on a BBN of an idealised example. IEEE Trans Software Eng, 2007, 33: 347–365
15 Stéphane L P, Michael B. A trust analysis methodology for pervasive computing systems. In: Trusting Agents for Trusting Electronic Societies. LNCS, Vol 3577. Heidelberg: Springer-Verlag, 2005. 129–143
16 Aarthi N, Vijay V. Dynamic trust enhanced security model for trusted platform based services. Future Gener Comp Sy, 2011, 27: 564–573
17 Daniele C R, Andrea B, Montani S, et al. A dynamic Bayesian network based framework to evaluate cascading effects in a power grid. Eng Appl Artif Intel, 2012, 25: 683–697
18 Zheng Z M, Ma S L, Li W, et al. Dynamical characteristics of software trustworthiness and their evolutionary. Sci China Ser F-Inf Sci, 2009, 52: 1328–1334
19 Zheng Z M, Ma S L, Li W, et al. Complexity of software trustworthiness and its dynamical statistical analysis methods. Sci China Ser F-Inf Sci, 2009, 52: 1651–1657
20 Christopher J B, Rua M. Dynamical conditions for convergence of a maximum entropy method for Frobenius Perron operator equations. Appl Math Comput, 2006, 182: 210–212

21 John G, Philip H. Nonlinear Oscillations, Dynamical Systems, and Bifurcation of Vector Fields. Berlin: Springer-Verlag, 1983
22 Lasota A, Mackey M. Chaos, Fractals and Noise. Berlin: Springer-Verlag, 1994
23 Robinson R C. An Introduction to Dynamical Systems: Continuous and Discrete (in Chinese). Beijing: China Machine Press, 2005
24 Beck C, Schlogl F. Thermo Dynamics of Chaotic Systems. New York: Combindye University Press, 1993
25 Zhang Z F, Li C Z, Zheng Z M, et al. Bifurcation Theory in Vector Fields (in Chinese). Beijing: Higher Education Press, 1997
26 Brendan D G. Digital forensics of the physical memory. Dig Invest, 2007, 4: 62–64
27 Liu H, Min Y. Calculation of two-dimensional characteristic invariant manifolds on the boundary of transient stability region in power system. Power Syst Technol, 2009, 33: 5–10
28 Ding J, Zhou A H. The projection method for computing multi-dimensional absolutely continuous invariant measures. J Stat Phys, 1994, 77: 899–908
29 Gary F. Approximating physical invariant measures of mixing dynamical systems in higher dimensions. Nonlinear Anal, 1998, 32: 831–860