

考虑历史数据的软件可信演化分析方法

赵 倩^{1,2} 冯光升² 李 莉¹

(哈尔滨商业大学计算机与信息工程学院 哈尔滨 150028)¹

(哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)²

摘 要 针对软件可信演化中对软件可信性相关属性的需求,提出一种考虑历史数据的软件可信演化分析方法。通过收集软件演化的历史数据,提取软件可信演化相关属性并进行量化,建立软件可信演化着色原则,构建软件可信演化分析图。实验结果表明,通过对软件可信演化分析图中演化单元的观察,可实时、有效、直观地得到内嵌在演化数据中的隐藏信息,以此分析出软件演化是否为可信演化,从而为演化数据的分析提供支持。

关键词 软件,软件可信演化,软件可信性,分析

中图法分类号 TP302.7 **文献标识码** A

Software Dependable Evolution Analysis Method Considering Software Historical Behavior Data

ZHAO Qian^{1,2} FENG Guang-sheng² LI Li¹

(School of Computer and Information Engineering, Harbin University of Commerce, Harbin 150028, China)¹

(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China)²

Abstract According to the dependability need in software dependable evolution, a software dependable evolution analysis method considering software historical behavior data was proposed. By collecting historical behavior data in software evolution, the method extracts and computes related properties of software dependable evolution, establishes coloring principle and builds software evolution analysis chart. Hidden information embedded in evolution data is gotten real-time, effectively and intuitively through observing software unit in software evolution analysis chart. It can analyze whether evolution is dependable or not to support evolution data analysis.

Keywords Software, Software dependable evolution, Software dependability, Analysis

1 引言

在当今信息技术社会中,各个行业的运行都非常依赖软件,如政府、工业、运输、商业、制造和服务行业。而软件生产和软件质量一直不能满足人们的期望,而且软件一直受到衰退问题的困扰。解决这一困扰的有效途径就是对软件进行修改,使软件发生演化,以适应用户需求和环境的不断变化。演化是软件生命周期普遍存在的现象,其本质是一种由用户预期偏差所驱动的软件修改过程,是软件系统从简单到复杂、从初步可用到成熟可信的渐变过程^[1]。而软件可信性在演化过程中是非常重要的属性,任何时候可信性的降低都可能给软件系统带来灾难性的后果。在演化过程中,如果软件可信性随着演化的发生而不断降低,这种演化并不是用户所能接受的演化,可以说是失败的软件演化。因此,为了强调软件可信性在演化中的作用,一些学者将能够有效提升软件可信性的演化过程称为软件可信演化^[2]。近年来,鉴于网络对软件技术产生的深刻影响,人们从不同角度和不同层面开展了诸多促进软件可信演化的探索和研究^[3-6]。这些研究都为软

件可信演化的发展提供新的切入点。

在软件演化过程中将产生很多数据,相对于当前时刻,将以往演化过程中的数据简称为软件的历史数据。软件的历史数据包含了很多软件变化的有用信息,从已经发生过的情况中,根据目前收集到的情况分析已经发生和未来将要发生的行为是符合生物认知规律的,同时也能为软件演化提供指导,促进软件演化的进行。本文正是使用所提出的软件可信演化分析图,从这些软件的历史数据中找寻那些能够保持和提高软件可信性的演化操作,为未来的可信演化提供支持。

2 国内外研究现状

经过多年的研究,国外已经有软件演化可视化技术。这些可视化分析技术使得人们能从源代码版本、缺陷报告或者维护日志中找寻埋藏在历史数据中的隐藏信息,这些隐藏信息对软件系统演化有极大用处。

Harald Gall 等人^[7]提出一个三维直观表示方法来检测一个软件版本的历史信息。三维信息分别是:时间、软件系统结构、一种属性值。每个结构用版本序列号确定,并且拥有自

到稿日期:2012-07-24 返修日期:2012-12-10 本文受国家自然科学基金资助项目(60873036,60973027),中央高校基本科研业务费专项资金(HEUCF100601),黑龙江省自然科学基金(F201037),哈尔滨市科技创新人才研究专项基金(2011RFQXG007),黑龙江省教育厅科学技术研究项目(12521135)资助。

赵 倩(1980—),女,博士,讲师,主要研究方向为自律计算、软件可信性、软件演化,E-mail:zhaoqian@hrbeu.edu.cn。

己的属性值,这些属性值用颜色来表示,每个颜色模块与抽象分解模块连接。这种三维图形表示方法较为复杂,不利于表示和观察。Michele Lanza^[8]提出一个基于软件可视化和软件度量的演化矩阵方法,其使用一个矩形来表示软件的类,矩形的长和宽表示要度量类的属性值。在软件运行过程中,类相关属性发生变化,则矩形也跟着变化。通过观察矩形的变化来分析软件运行过程中的软件演化。方法的目的主要是寻找两个版本软件之间的差异,如果软件使用的是面向对象编程中类继承,则不能够得到软件的演化矩阵图。Wu 等人^[9,10]使用光谱来研究软件演化,采用光谱来实现软件的可视化分析,根据软件历史数据的可视化记录,对不同版本的软件发生哪些改变进行了分析。其由于在记录过程中没有反映软件性能方面的参数,因此不能表示不同版本之间软件其余性能指标。Tudor Girba 等人^[11]根据软件在运行过程中的一系列需求提出一个演化分析方法 Hismo,其通过可视化的手段对软件历史行为进行建模,以表示软件演化的过程。这种分析方法更多是描述和表示历史演化,而对软件演化分析的支持不够充分。Robbes^[12]为软件中操作和信息的变化进行建模,以便从程序环境中快速找到影响系统的操作,这样就能够对软件的运行进行准确地监控,可以使开发者和研究者更好地利用和管理更为复杂的系统。

根据软件演化分析的国内外现状可知,有些演化方法在记录过程中没有反映软件性能方面的参数,不适合用于分析和表示软件可信演化;有些演化方法只是描述和表示历史演化,对软件演化分析的支持不够充分,另外一些的可信程度不足。如何建立具有良好可视化且能有效反映软件可信性变化的软件可信演化分析方法成为当前软件可信演化的研究热点。本文使用考虑历史数据的演化分析方法记录软件演化过程,通过对可视化记录的分析,研究软件演化策略是否得当,指导软件演化,约束演化行为,保障软件在演化过程中可信性的增长。

3 考虑历史数据的软件可信演化分析方法

3.1 软件可信演化分析方法建立过程

演化是软件生命周期普遍存在的现象,其本质是一种由用户预期偏差所驱动的软件修改过程。本文通过建立软件可信演化分析图的方法来对软件可信演化进行实时、可视化的记录与分析,软件可信演化分析图的建立过程如图 1 所示。

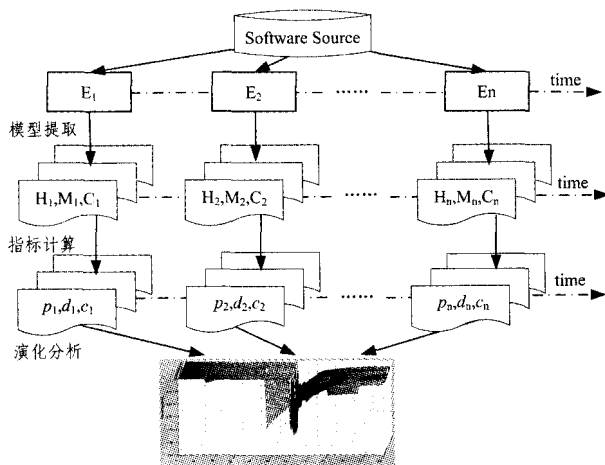


图 1 软件可信演化分析图的建立过程描述

首先对待分析的软件进行建模,软件可信演化分析模型是一个三维结构,由矩阵 M 和 H 组成。矩阵 M 是一个二维矩阵,对于给定的软件,其上的软件单元可根据属性 p 来计算。矩阵中的行向量存储的是软件单元演化历史值的向量值,列向量存储的是特定时间点上所有软件单元演化属性和策略等的演化状态。如果研究的是具有 m 个软件单元 $(u_0, u_1, \dots, u_m)$ 的软件在 n 个时间点 $(t_0, t_1, \dots, t_n)$ 中的演化,则矩阵 M 就是一个 $m \times n$ 的矩阵。矩阵 H 也是一个二维矩阵,对于给定的软件单元,其上的软件单元可根据可信性 d 来计算。矩阵中的行向量存储的是软件单元可信性演化历史值的向量值,列向量存储的是一段时间内所有软件单元可信性的演化情况。将这种矩阵视为基于属性的表达,它能定量地刻画软件系统的状态和可信性演化历史。使用颜色来表示演化矩阵 M 的值,产生软件可信演化分析图。然后根据软件实际运行的数据统计和计算各项指标值。最后,软件可信演化分析图中的颜色可使人们从大量的演化数据中找出内嵌在演化数据中的软件演化模式。这些演化模式体现了软件根据内外部环境所做出的反应,一方面可以体现软件的内外部环境变化,另一方面可以使人们从纷繁芜杂的演化数据中对演化数据有直观、整体的了解,更有利于演化数据的分析。

3.2 软件可信演化分析图各维定义

演化图为软件的内容和变化提供了一个实时的可视化表示,演化图中各维数可用一个四元组 $\langle T, S, EI, D \rangle$ 来表示,其元素分别表示时间、软件单元、演化信息、可信属性。

$T: t_i \in T (i < N_0)$ 表示所观察软件系统的演化时间。 T 有两种表示方法:一种以软件可信演化事件为单位时间,如软件释放或提交相关命令;另一种使用固定的时间单元为单位时间,如分、小时。研究目的不同,单位时间选取也不相同。如果系统分析软件结构演化,就需要以软件释放命令为单位时间进行演化图的绘制,因为系统结构会因为软件释放相关命令而发生很大变化。如果需要研究软件开发者的行为,选取固定时长的时间作为单位时间将是较为合理的。

$S: s_j \in S (j < N_0)$, 其中 s_j 可以是组件、线程等任何人们想要考察软件的单位。在分析演化过程中,会以研究 S 的变化来分析软件整体上的演化情况, s_j 以一种有序的位置放置在演化图中。通常情况下,以 s_j 创建和修改的时间作为软件放置软件单元的次序。

$EI: e_{s_j}^{t_i} (e_{s_j}^{t_i} \in EI)$ 表示 s_j 在 t_i 时刻软件的演化信息值。演化信息主要反映了除软件可信性相关属性外 s_j 中所发生的变化,如代码的行数、扇入和扇出的依赖性、失效密度以及 s_j 中增加或减少的组件数。

$D: d_{s_j}^{t_i} (d_{s_j}^{t_i} \in D)$ 表示 s_j 在 t_i 时刻软件可信性相关属性的度量值。假设 $d_{s_j}^{t_i}$ 和 $d_{s_j}^{t_k}$ 分别表示 s_j 在 t_i 和 t_k 时刻软件的可信属性度量值。若 $t_i < t_k$ 且 $d_{s_j}^{t_i} > d_{s_j}^{t_k}$, 则说明演化后的软件比没有演化的软件在可信性上还要差,这样的演化是失败的;相反,若 $t_i < t_k$ 且 $d_{s_j}^{t_i} \leq d_{s_j}^{t_k}$, 则说明演化是成功的,是可信演化。可信属性可以是软件可信性的度量值,也可以是可信性相关子属性的值,如可靠性、可用性、可生存性。

3.3 模型中的着色原则

四维属性中,使用颜色来表示一维数据的值,以产生最后的三维图。图中的颜色可使人们从大量的演化数据中直观地找出内嵌在演化数据中的隐藏信息,更有利于人们对演化数

据的分析。当形成这些颜色数据后,就能从这些颜色中观察到系统中复杂的软件可信演化过程。着色函数不是统一的,需要针对不同对象和目标来选取和建立。通常有线性映射、指数衰退映射等。本文的实验中将要使用指数衰退映射,因此对其进行相应的介绍。

指数衰退染色原则需要用到3种颜色: C_{Hot} , C_{Cold} 和 C_{BG} , 及一个梯度染色函数 g 。设矩阵 M 中 s_i 在 t_j 时刻的演化值为 $d_{ij}^{t_j}$, 其相关颜色为 $c_{ij}^{t_j}$ 。通过式(1)来计算所着的颜色。

$$c_{ij}^{t_j} = \begin{cases} C_{BG}, & \text{if } d_{ij}^{t_j} \text{ empty} \\ C_{Hot}, & \text{else if } d_{ij}^{t_j} \geq T \\ C_{Hot}, & \text{else if } d_{ij}^{t_j-1} = \text{empty} \\ g(c_{ij}^{t_j-1}), & \text{otherwise} \end{cases} \quad (1)$$

式中, T 是阈值, empty 表示单元中没有值。对于分析图中每个单元,如果单元没有值,将图中的背景颜色 C_{BG} 赋予没有值的单元。如果值大于等于 T ,或者之前的单元没有值,给这个单元填颜色 C_{Hot} ; 否则,根据函数值 g 来决定颜色。函数值 g 的定义如式(2)所示:

$$\begin{aligned} g(c) &= [r', g', b'] \\ r' &= r(c) + (r(C_{Cold}) - r(c)) \times \text{decay} \\ g' &= g(c) + (g(C_{Cold}) - g(c)) \times \text{decay} \\ b' &= b(c) + (b(C_{Cold}) - b(c)) \times \text{decay} \end{aligned} \quad (2)$$

使用 $[r(c), g(c), b(c)]$ 来表示颜色 c , 其中, $r(c)$, $g(c)$ 和 $b(c)$ 分别表示 RGB 颜色模式下的红色、绿色和蓝色数值。函数 g 以指数衰退改变色彩直到 C_{Cold} 。式(2)中的衰退因子 $\text{decay} \in [0, 1]$ 。

4 软件可信演化分析方法实验

取 AschoalSAE 软件的演化数据作为软件可信演化分析方法实验的分析数据,计算分析图中相关维数上的数据数值,确定合适的着色原则,根据所做准备进行软件可信演化分析图的绘制。最后将本方法与 Wu 的基于光谱软件演化方法进行对比,以说明本方法在反映软件演化与软件可信性上较借鉴基于光谱软件演化方法的有效性。

实验环境:5 台测试主机和 1 台被测主机的配置均为 CPU 为 Pentium4, 2GHz, 双核处理器, 1G 内存, 安装 Windows XP 系统。被测主机上安装 Tomcat, 为测试主机提供服务。使用项目组开发的 Process Manager 软件对被测主机进行测试, SystemInfor 软件对测试信息中相关内容进行提取。测试主机安装 JMeter 测试软件, 向被测主机的发起 HTTP 请求。

实验过程:4 台测试主机使用 JMeter 模拟用户向被测主机上的 AschoalSAE Web 软件系统发起 HTTP 请求, 模拟服务器向用户提供 HTTP 请求服务。线程产生时间为 5s, 每个线程包含 5 个 HTTP 请求, 线程间的时间延迟为 3s。线程变换情况如下: 600 min. 100 线程; 600 min. 500 线程; 300 min. 1000 线程; 300 min. 500 线程; 300min. 100 线程; 100 min. 0 线程。在实验过程中分别 5 次向被测主机发起 SQL 注入攻击。注入时间分别为第 300、900、1300、1500 和 1900min., 每次持续时间为 5min。

根据软件可信演化分析图建立过程中各个步骤, 得到软件可信演化分析图, 根据分析图中的结果进行演化分析。选

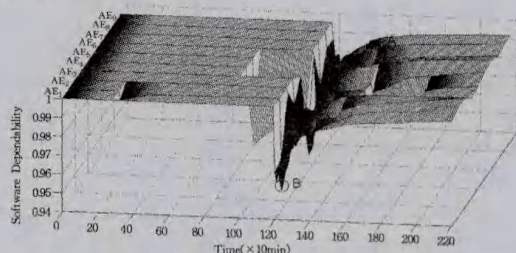
取 AE 作为指标计算的基本单元, 则矩阵是由软件运行时间内所有 AE 的指标组成的。使用 X 轴来表示时间, Y 轴来表示 AE 个数的变化情况, Z 轴来表示软件的可信性值。

统计软件运行时间, 共计 2200min.。将 10min 作为演化研究单元, 分别考察各个 AE 在 10min 内 Component 个数的最大值, 如果 AE 中并无相应的 Component, 则使用 -1 来表示当时 AE 的情况; 如果 AE 被删除了, 也使用 -1 来表示, 统计 AE 的变化情况, 刻画矩阵 M 。同时计算软件可信性度量方法得到矩阵 H 。着色原则依然选取指数衰退型的着色原则, 根据软件可信演化矩阵 M , 计算相应的色彩值 C , 映射到不同的颜色, 用来表示矩阵 M 。选取指数衰退型着色方法, 根据式(2)可得 $c_{ij}^{t_j}$, 其中 C_{BG} 为白色, C_H 为黑色。函数 g 如式(3)所示。

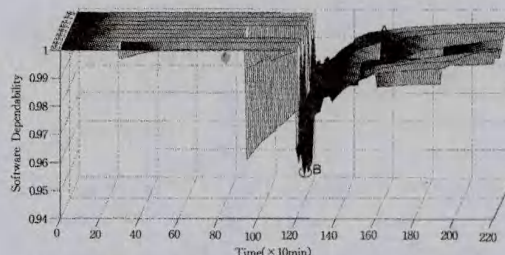
$$\begin{aligned} g(C[i][j]) &= [r(C[i][j]), g(C[i][j]), b(C[i][j])] \\ r(C[i][j]) &= \lceil 256 - (256 - r(C[i][j-1]) \times 0.95) \rceil \\ g(C[i][j]) &= \lceil 256 - (256 - g(C[i][j-1]) \times 0.95) \rceil \\ b(C[i][j]) &= \lceil 256 - (256 - b(C[i][j-1]) \times 0.95) \rceil \end{aligned} \quad (3)$$

式中, $r(c)$, $g(c)$ 和 $b(c)$ 分别表示 RGB 颜色模式下的红色、绿色和蓝色数值。

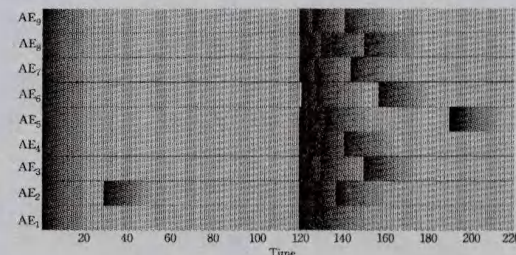
根据定义的分析图各维含义和相关矩阵可得软件可信演化分析图, 图 2(a)、图 2(b) 为使用本方法描绘的软件可信演化分析图, 图 2(c) 为根据 Wu 研究的基于光谱的软件演化方法所绘制的分析图。通过比较本方法与基于光谱的软件演化方法说明, 本方法在可视化显示软件可信与软件演化相关方面的有效性。



(a) AschoalSAE考虑历史数据的软件可信演化分析图(俯视)



(b) AschoalSAE考虑历史数据的软件可信演化分析图(平视)



(c) AschoalSAE基于光谱的软件演化分析图

图2 两种分析图的比较

Aschoal_{SAE}共有 28 个 AE,其中有 9 个 AE 参与了 HT-TP 的响应。演化图中只记录了参与演化的 AE 的情况,这样能集中反映与演化有关 AE 的相关情况,更有利于对软件本次运行的分析。图 2 分别利用本文提出的软件可信演化方法和基于光谱的软件演化方法描绘了 Aschoal_{SAE}软件在实验中的数据。

从图 2(a)和(b)可以看出,整个 A 和 B 区域都被深色填充着,这说明在此阶段中 AE($i=1\sim 9$)中的 Component 不断进行变化,不断地增加和删除。同时可以看到,A 区域中软件可信性呈上升趋势,这说明此时软件的自配置采取的演化手段有利于软件可信性的保持和增长。而 B 区域中,软件可信性在实施自配置操作后则发生了软件可信性降低的结果,这说明配置策略选择不当,需要着重观察此时进行演化的 AE 自配置策略和演化方法,以便对其进行调整。但是从图 2(c)中只能看到 A 和 B 两个区域发生了演化,演化频率比较高,不能说明演化与软件可信性之间的关系,不能对演化后软件可信性的相关情况进行控制,无法对演化策略进行指导。

结束语 本文通过对软件行为的分析和对软件行为历史数据的刻画,选取软件可信演化的策略,使软件可信性保持在一个稳定且局部非降的水平上,满足用户对软件可信性的需求。使用软件自律化方法使软件中的组件具有一定的自我管理能力,通过改变组件之间的关系进行演化操作,以软件可信演化历史反映软件可信演化对软件可信性的影响,帮助软件可信演化选择适当策略,最终在无人工干预或少人工干预情况下完成软件可信演化,以达到软件可信性的保持和增长。

参考文献

- [1] 金海,李克秋.可信计算[J].中国计算机学会通讯,2010,6(2):6-7
- [2] 王怀民,尹刚.网络时代的软件可信演化[J].中国计算机学会通讯,2010,6(2):28-36
- [3] 曾晋,孙海龙,刘旭,等.基于服务组合的可信软件动态演化机制[J].软件学报,2010,21(2):261-276
- [4] 李季,刘春梅.基于本体的可信软件演化框架模型[J].计算机应用研究,2010,27(12):4551-4554
- [5] Oreizy P, Medvidovic N, Taylor R. Architecturebased runtime software evolution[C]//Proceedings of the International Conference on Software Engineering, Kyoto, Japan, April 1998:177-186
- [6] Oreizy P, Medvidovic N, Taylor R N. Runtime software adaptation: framework, approaches, and styles[C]//Proceedings of the International Conference on Software Engineering. ACM Press, 2008:899-910
- [7] Gall H, Jazayeri M, Riva C. Visualizing software release histories: The use of color and third dimension[C]//Proceedings of the International Conference on Software Maintenance. IEEE Press, 1999:99-108
- [8] Lanza M. The evolution matrix: recovering software evolution using software visualization techniques[C]//Proceedings of the International Workshop on Principles of Software Evolution. ACM Press, 2001:37-42
- [9] Wu J, Holt R, Hassan A. Exploring software evolution using spectrographs[C]//Proceedings of 11th Working Conference on Reverse Engineering. IEEE Press, 2004:80-89
- [10] Wu J, Spitzer C, Hassan A, et al. Evolution spectrographs: visualizing punctuated change in software evolution[C]//Proceedings of 7th International Workshop on Principles of Software Evolution. ACM Press, 2004:57-66
- [11] Grba T, Ducasse S. Modeling history to analyze software evolution[J]. Journal on Software Maintenance and Evolution: Research and Practice, 2006,18(3):207-236
- [12] Robbes R, Lanza M. A change-based approach to software evolution[J]. Electronic Notes in Theoretical Computer Science, 2007,166:93-109
- [13] Berger R L. A necessary and sufficient condition for reaching a consensus using DeGroot's method[J]. Assoc. , 1981,76(374):415-418
- [14] DeMarzo P, Vayanos D, Zwiebel J. Persuasion Bias, Social Influence, and Unidimensional Opinions[J]. Quarterly Journal of Economics, 2003,118(3):909-968
- [15] Donetti L, Hurtado P I, Munoz M A. Entangled networks, synchronization, and optimal network topology[J]. Physical Review Letters, 2005,95(18):1-4
- [16] Golub B, Jackson M O. Naive learning in social networks and the wisdom of crowds[J]. Amer. Econ. J. Microecon, 2010,2(1):112-149
- [17] Pan Z. Trust, influence, and convergence of behavior in social networks[J]. Mathematical Social Sciences, 2010,60(1):69-78
- [18] 陈友,程学旗,杨森.面向网络论坛的高质量主题发现[J].软件学报,2011,22(8):1785-1804
- [19] Hegselmann, Rainer, Krause U. Opinion dynamics and bounded confidence: models, analysis, and simulation[J]. Journal of Artificial Sciences and Social Simulation, 2002,5(3):1-33
- [20] Slanina F. Dynamical phase transitions in Hegselmann-Krause model of opinion dynamics and Consensus[J]. European Physical Journal B-Condensed Mater, 2011,79(1):99

(上接第 152 页)

- [7] He M F, Li B, Luo L D. Sznajd model with "social temperature" and defender or small-world networks[J]. Int. J. Mod. Phys. C, 2004,15(7):997-1003
- [8] Aleksiejuk A, Holyst J A, Stauffer D. Ferromagnetic phase transition in Barabasi-Albert networks[J]. Physica A, 2002,310(1/2):260-266
- [9] Schulze C. Long-range interactions in Sznajd consensus [J]. Physica A, 2003,324(3/4):717-722
- [10] Schulze C. Advertising in the Sznajd marketing model[J]. Int. J. Mod. Phys. C, 2003,14(1):95-98
- [11] Sznajd W K, Sznajd J. Who is left, who is right? [J]. Physica A, 2003,351(2-4):593-604
- [12] Sabatelli L, Richmond P. Non-monotonic spontaneous magnetization in a Sznajd-like consensus model[J]. Physica A, 2004,334(25):274-280