

# 基于受限生成过程模型的计算涌现分析

张海粟<sup>1</sup> 张松林<sup>2</sup> 陈桂生<sup>3</sup>

(解放军理工大学指挥自动化学院 南京 210007)<sup>1</sup> (解放军理工大学通信工程学院 南京 210007)<sup>2</sup>  
(中国电子设备系统工程研究所 北京 100141)<sup>3</sup>

**摘要** 涌现描述了特定系统在超过某阈值时突然出现的现象,中间没有明显的过渡过程。提出了图灵机计算模型在时空复杂度上所表现出的计算涌现现象,引入了受限生成过程(Constrained Generating Procedure,CGP)模型来描述图灵机的计算过程,通过CGP模型刻画了机制参与次数、参与深度和平均参与度等3个涌现数字特征,提出了计算涌现的CGP分析方法并在3类典型图灵机计算过程中进行了验证分析。

**关键词** 计算涌现,受限生成过程,图灵机,复杂性分析

中图分类号 TP301.5 文献标识码 A

## Computational Emergence and its Constrained Generating Procedure Model

ZHANG Hai-su<sup>1</sup> ZHANG Song-lin<sup>2</sup> CHEN Gui-sheng<sup>3</sup>

(School of Command Automation,PLA University of Science and Technology,Nanjing 210007,China)<sup>1</sup>

(School of Telecommunication,PLA University of Science and Technology,Nanjing 210007,China)<sup>2</sup>

(Institute of Electronic System Engineering,Beijing 100141,China)<sup>3</sup>

**Abstract** The complexity of computational models can also put up some emergence properties around several certain critical values. This paper surveyed some computational emergence behaviors, and built their Constrained Generating Procedure(CGP) model. We introduced the CGP mechanic participating times, participating depth and average participating degree, and then analyzed the critical values of emergence properties on 3 typical Turing machine computation processes.

**Keywords** Computational emergence, Constrained generating procedure, Turing machine, Computing complexity

## 1 引言

特定系统所产生的涌现现象往往随着行为机制的数量增加和积累而“出乎意料”地表现出来。因此,如何描述、再现甚至控制特定系统中这种看似难以重复的过程已经成为目前复杂性研究的重点<sup>[1]</sup>。在可计算性研究领域,基于图灵机模型的计算实现过程往往也会表现出某种涌现特性——一个用来实现 $x+y$ 功能的、设计合理的图灵机计算过程是简单的,而计算 $x \cdot y$ 的过程必须在 $x+y$ 的简单设计上使其进一步复杂化(包括增加状态、跳转和步骤等)才能得到结果。如果再将难度提高,考虑一个计算 $x^n$ 的图灵机过程,则计算的复杂性将可能会成倍增加。可是,在图灵机所有这些成倍增加的计算复杂性后面,起支撑作用的不过仅仅是4条简单指令:写0,写1,左移和右移。随着计算任务的多样化和复杂化,通过简单指令的组合所得到的动态的、不断增长的复杂性表现出一种具有涌现特点的复杂演进过程。这种与计算模型相关、由计算模型产生的时空复杂性上的涌现现象可称之为“计算涌现”<sup>[3,4]</sup>。涌现理论本身还处于起步阶段,关于计算涌现的

具体研究结论目前更加少见,文献[3,4,6,8]从不可计算性的角度宏观地讨论了不同计算任务的涌现的差异问题,如通过振荡机方式所构造出的停机函数不可判的计算复杂性和 $xy$ 任务的计算复杂性之间的涌现区别等,但是均没有从涌现机理上建立完整的分析模型。

基于主体(Agent)的受限生成过程模型(Constrained Generating Procedure Model, CPG)最早由Holland提出<sup>[1]</sup>。在CGP模型中,主体是对系统中产生涌现现象的各要素的概括,通过给主体赋予机制(Mechanism)来刻画其能动性,而机制间的相互作用则用于描述主体间的相互影响。文献[1,7,9]等详细分析了与神经网络、西洋跳棋和元胞自动机相关的几个实例,并通过CGP模型描述了其特征和涌现现象。

不论在可计算性理论还是在计算涌现中,诸如图灵机、棋盘机和递归函数等计算模型及其计算过程的研究都是最基础的。本文通过抽象图灵机的4条基本指令,将其映射为主体的状态装换函数,建立了CGP模型,然后通过确定主体之间相互作用的接口函数,形成了多个CGP模型的复合;最后,基于CGP模型抽象了3个与计算涌现相关的特征参数;机制的

到稿日期:2010-11-16 返修日期:2011-02-26 本文受国家重点基础研究发展计划(973计划)(2007CB310804),国家自然科学基金重点项目(61035004),国家自然科学基金面上项目(69120912)资助。

张海粟(1982—),男,博士生,主要研究方向为复杂系统分析和复杂网络挖掘,E-mail: zhanghaisu@139.com;张松林(1986—),男,硕士生,主要研究方向为复杂系统分析;陈桂生(1966—),男,高级工程师,主要研究方向为复杂系统分析。

参与次数、参与程度和平均参与度,并完整描述和初步分析了图灵机模型的计算涌现过程。

本文第2节建立了图灵机计算过程的CGP模型;第3节在CGP模型的基础上分析了计算复杂度的涌现临界;第4节是相关工作简介。

## 2 图灵机计算过程的CGP模型

### 2.1 图灵机计算过程

图灵机的理论模型<sup>[2]</sup>把推理化作一系列简单的机械动作。如图1所示,图灵机由一条双向可无限延长且被分成一个个方格的磁带(格里写有符号0或1)、一个有限状态控制器和一个读写磁头组成。图灵机的动作由四元组 $\langle 0, 1, p, q \rangle$ 确定。其中,0和1为方格中的写0和写1动作; $p$ 和 $q$ 分别指示磁头移动方向,或左或右。

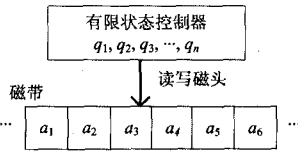


图1 图灵机模型

由状态和符号确定的工作过程称为图灵机程序,各类可计算问题均可以转化为图灵机程序来表达,由一个有限状态机的执行过程来表示,如图2(a)、图2(b)和图2(c)分别是写数值 $x$ 、计算 $2 \cdot x$ 和 $x \cdot y$ 的图灵机计算过程<sup>[2]</sup>。在此机械的自动工作过程中,不需要人的任何干预或指导。

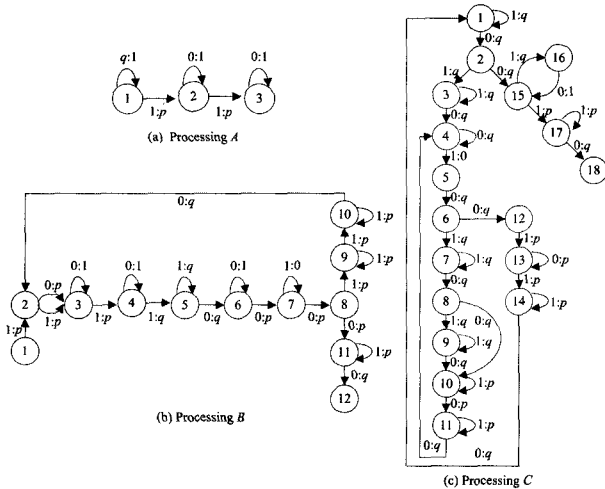


图2 3个典型的图灵机计算过程<sup>[2]</sup>

### 2.2 CGP模型

在图灵机模型中,用虚拟的磁头以及磁头在磁带上各空格处所进行的动作来完成各种计算。通过CGP模型来刻画图灵机,主体可选择空格或者磁头。由于图灵机磁带上的空格是两两相邻排列的,若选择空格,则在CGP模型中也可定义左右相连的主体及其交互,因此可定义计算主体为空格。

**定义1**(计算主体 $A_i$ 及状态集合 $S$ ) 图灵机磁带上的空格 $i$ 对应为计算主体 $A_i$ ;  $A_i$ 存在0和1两种状态,即状态集合 $S = \{S_1, S_2\} = \{0, 1\}$ 。

主体的机制是指其可能具备的状态变换过程,计算主体 $A_i$ 的机制即表示在状态 $S_1$ 和 $S_2$ 之间相互变换的过程。正是这种机制导致了系统中主体行为的动态性,进而引起了系统整体的状态变化。描述主体 $A_i$ 自身的机制所形成的状态

变换的方法,可以引入转换函数 $f_i$ 。需要考虑下面两个因素:

a)  $f_i$ 的输入 $I_i$ 为主体 $A_i$ 的可能的动作,因此, $I_i = \{i_{i1}, i_{i2}, i_{i3}, i_{i4}\} = \{0, 1, p, q\}$ ,其中, $p$ 和 $q$ 分别代表磁头左移和右移,0,1则分别代表写0和写1。通过转换函数之后, $A_i$ 将由上一个状态进入下一个状态,故转换函数 $f_i$ 所完成的映射为 $I_i \times S_i \rightarrow S_i$ 。

b) 图灵机随着时间 $t$ 不断运转,因此,对 $I_i$ 和 $S_i$ 还应该引入参数 $t$ 以刻画其随 $t$ 变化的特性,即 $S_i(t+1) = f_i(I_i(t), S_i(t))$ 。在不断地输入组合序列 $I_i(t+k)$ ( $k \in 1, 2, \dots, +\infty$ )的情况下,可以得到 $S_i(t+k)$ ( $k \in 1, 2, \dots, +\infty$ )。

据此,可定义转换函数 $f_i$ ,用来完整表征主体 $A_i$ 随着时间而变化的机制。

**定义2**(主体 $A_i$ 转换函数 $f_i$ )  $f_i(t+1): I_i(t) \times S_i(t) \rightarrow S_i(t+1)$ 。

$f_i$ 描述了主体通过相应的机制由一个状态转换到另一个状态,类似主体 $A_j$ 的转换函数 $f_j$ 为 $f_j(t+1): I_j(t) \times S_j(t) \rightarrow S_j(t+1)$ 。

分析标准的图灵机模型,在起始状态下,除磁头停留在一个空格起始位置上以及 $n$ 个初始设定的输入参数(空格上的字符为1)之外,其他的均为0,因此可得定理1。

**定理1** 在 $t=0$ 的起始状态,除 $n$ 个主体的起始状态为1(对应于图灵机 $n$ 个单元的输入参数)之外,其余的机制状态必然均为0。

在其他的普通CGP模型中,自由输入是指在某一时刻的输入值由CGP的外部因素(如人的参与)来提供。在计算涌现的CGP模型中应避免自由输入的引入,这一点和图灵机理想模型中强调“不需要人的任何干预或指导”的原则是一致的。

### 2.3 CGP模型的复合

最简单的CGP模型即可定义为一个状态转换函数 $f_i$ 。但是磁带上具有无限多个空格,随着图灵机程序的运行,空格之间会发生相应的状态转换,因此需要确定多个简单CGP模型的复合。首先从磁头的起始位置出发,按照左右交替的方式为其编码,得到所有主体的编码 $x, x \in \{1, 2, \dots\}$ ,如图3(a)所示。为表示每一个主体的转换函数之间的结合,我们引入接口函数的概念。

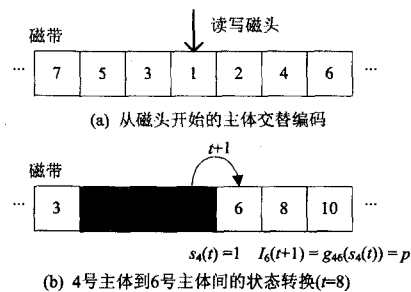


图3 主体的交替编码以及主体间的状态转换示例

主体之间的结合通过将某主体在 $t$ 时刻的状态转换为另一个主体在 $t+1$ 时刻的合法输入来实现。根据主体 $A_i$ 在时刻 $t$ 的状态 $S_i(t)$ ,通过接口函数 $g_{ij}$ 所完成的映射功能,就可以确定在 $t+1$ 时刻将要参与运算的主体 $A_j$ 的输入 $I_j(t+1)$ 。

定义 3(接口函数  $g_{ij}$ )  $g_{ij}:S_i(t) \rightarrow I_j(t+1)$  也可以表示成函数形态:  $I_j(t+1) = g_{ij}(S_i(t))$ 。

示例 图 3(b)是图 2(c)中的一个执行片段,阴影空格表示已被写 1,其实现了将空格 4 上的值 1 转移到空格 6 上的功能。接口函数描述了从  $t=8$ (即图灵机计算过程的第 8 步)时在 4 号主体和 6 号主体之间的转换过程,即 6 号主体以 4 号主体在  $t$  时刻的状态 1(写 1)为参数,根据  $g_{46}$  得到  $t+1$  时刻的输入为  $p$ 。

### 3 基于 CGP 模型的计算涌现分析

分析产生计算涌现的特定系统,可以从完成典型计算任务入手。计算涌现所揭示的是计算模型在何时、何种情况下会出现涌现,以及这种涌现的判据(即临界)。本节根据 CGP 模型及其复合,通过定义与计算复杂性相关的 3 个指标得到典型计算任务的涌现出现临界,并且进一步得到了在可计算和不可计算的情况下,各类任务的计算涌现临界特征。

基于 CGP 模型的讨论与单独对计算模型进行讨论相比将会更加简洁:其一,通过指定转换函数,可以任意模拟在图灵机计算过程中的 4 种动作,而且可以同时考虑时间  $t$  的因素(与图灵机的执行步骤有关);其二,通过 CGP 模型的复合,将复杂的计算过程通过接口函数组合起来,就能以统一的方法来分析计算涌现的临界。

#### 3.1 参与次数和参与深度

从计算过程 CGP 模型的输入集合  $I$  和状态集合  $S$  来看,随着系统复杂性的增加, $I$  和  $S$  组合空间的也将增加。 $I$  和  $S$  组合空间的增加带来的第一个变化是,每一个机制在运算中“参与”次数的变化,即磁头所反复扫描的次数的变化;第二是“参与”的深度,即磁头所扫描的最远距离的变化。

定义 4(主体  $A_i$  的参与次数  $Pt$ ) 引入距离序列  $C = \{c_i\}$ ,其中  $i$  为主体  $A_i$  的编码,主体  $A_i$  的状态序列  $S$  中反复出现的非 0 的计数  $c_i$  即为主体  $A_i$  的参与次数  $Pt$ 。

为定义参与深度,必须有一个变量能够记录下磁头最远扫描的位置与其初始位置之间的距离。只要区分磁头起始位置的左右两部分,就可以用主体的编码  $x$  表示。在交替编码的基础上引入允许负值的距离序列  $D = \{d_i\}$ ,其中  $d_i$  表示与初始位置的距离,向右一格为 1,向左一格则为 -1。

定义 5(主体  $A_i$  的参与深度  $Pd$ ) 主体  $A_i$  对应于距离序列  $D = \{d_i\}$  中的项  $d_i$ 。

#### 3.2 计算涌现临界

从系统随机性程度与计算复杂性的关系、不可计算过程对应的 CGP 判据和参与次数、参与深度之间的约束关系等 3 个角度来分析计算涌现的临界。

##### (1) 系统随机性程度与计算复杂性的关系

针对每一主体  $A_i$ ,为其 4 类不同的动作赋予权重  $WI_{ij} = \{w_{i1}, w_{i2}, w_{i3}, w_{i4}\} = \{1, 1, 2, 2\}_i$ ,其中  $i$  对应于主体  $A_i$  的编号,也代表了时刻  $t=i$  的状态  $S$ 。权重确定的原则为:0 和 1 分别和机制的某一个时刻状态有关,因此赋值为 1; $p$  和  $q$  不仅与当前状态有关,而且是下一步机制的输入,标识了下一时刻将要“激活”的机制,因此赋值为 2。

定义 6(主体  $A_i$  的平均参与度  $P$ ) 记

$$WH_i = \sum_{j=0}^n WI_{ij}$$

一个图灵机参与过程中所有  $WH_i$  的序列为  $\{WH_1,$

$WH_2, WH_3, \dots, WH_n\}$ ,将主体  $A_i$  的平均参与度  $P$  定义为  $WH_i$  序列的方差  $P = D(WH_1, WH_2, WH_3, \dots, WH_n)$ 。

考察图 2 所示的 3 个图灵机计算过程,其平均参与度  $P$  如图 4 所示。

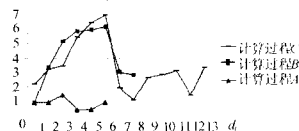


图 4 平均参与度  $P$

分析平均参与度  $P$  的统计特性,可以看出,距离磁头原点为  $d_i$  的各主体的  $P$  值仅仅在步骤执行中期的时候达到了最大值。若将  $d_i$  视为图灵机计算过程随机性的标志,则  $d_i$  的值越大,其随机性越强;而其平均参与深度并不是随着随机性的增加而显著增强,同样,也并不是在  $d_i=1$  的时候达到最深复杂度。通过 CGP 模型的分析,图灵机各主体的平均参与度变化规律与一般复杂系统<sup>[2]</sup>的结论基本相似,在系统演进的开始和结束的两端,复杂度或平均参与度均有所下降。

##### (2) 不可计算过程对应的 CGP 判据

根据转换函数  $f_i(t)$  的定义,平均参与度  $P$  即是其参数  $I_i(t)$  的方差。若以  $P$  和权重  $WI_{ij}$  的乘积  $P \cdot WI_{ij}$  作为参数输入  $f_i(t)$ ,则在序列  $WH_i \cdot WI_{ij}$  不收敛的情况下,对  $f_i(t)$  来说函数值将趋于无穷大。因此, $WH_i \cdot WI_{ij}$  的物理意义反映了计算模型的求解能力,可得定理 2。

定理 2 序列  $WH_i \cdot WI_{ij}$  不收敛代表不可计算的情况。

根据距离序列  $D = \{d_i\}$  的定义,其值是  $WH_i$  在所有时间  $t$  上的展开,所以两者具有同样的收敛性,因此  $D = \{d_i\}$  是否收敛同样也刻画了可计算性,可得定理 2 的推论。

推论 序列  $D = \{d_i\}$  不收敛也代表了不可计算的情况。

对于参与深度  $Pd$  来说,考虑死循环的情况:其参与深度并没有变化,甚至可以仅仅为 1(在两个状态循环不停的情况下),但是其不能计算出有意义的结果。此时参与深度  $Pd$  组成的序列中始终循环为 0 和 1 两项。从这个意义上来说, $Pd$  反映了计算的有效性。对于不可计算的情况来说,这种循环是否存在是难以判定的,因此可得定理 3。

定理 3 参与深度  $Pd$  组成的序列中若出现循环,则其对应的计算模型中也一定存在循环。

##### (3) 参与深度、次数之间的约束关系

一个图灵机计算过程中 CGP 模型的所有主体  $A_i$  的参与次数平均值为

$$Pt' = \frac{1}{k} \sum_{i=\min}^{\max} Pt$$

式中,取  $k = \max - \min$ 。类似可得到参与深度与平均参与度的平均值  $Pd'$  和  $P'$ 。

$$Pd' = \frac{1}{k} \sum_{i=\min}^{\max} Pd \quad P' = \frac{1}{k} \sum_{i=\min}^{\max} P$$

3 者之间有一定的约束关系。图灵机计算过程 A、B 和 C 的 CGP 模型中 3 者的数据列举如表 1 所列。

表 1 参与次数、参与深度和平均参与度

计算过程	主体数	执行步骤 $t_{\max}$	$Pt'$	$Pd'$	$P'$
A	6	8	1.3	3.6	1.5
B	8	50	3.8	7.5	4.2
C	13	287	6.3	10.3	7.6

从表 1 可以看出,1)通常随着图灵机的主体数增加, $Pt'$ ,  $Pd'$ 和  $P'$ 均会有所增加;2) $P'$ 的数值一般位于  $Pd'$ 和  $Pt'$ 之间,且近似靠近于  $Pt'$ 较小的端。

第二个结论目前不能简单推广至其他计算过程,因为模型经过复合之后需要重新计算相关的数值,复合关系并不能保证这种数值关系的一致性。从这个角度来说,尽管复杂的 CGP 模型是由简单的 CGP 模型迭代产生的,但其整体规律性也可能并不是简单的相加。

#### 4 相关工作

文献[3,4]等从计算系统状态的角度进行了递归、差分方程、后效性等几种情况的分析,讨论了一些不同计算任务复杂性涌现的差异问题,包括通过振荡机方式所构造出的停机函数不可判的计算复杂性和  $x^y$  任务等。其提出关于计算涌现的数学建模下一步研究的重点应在于抓住其中主体及其聚集、系统的数学特性之间的关系,并且强调了其中不会存在突变,但是并没有从涌现机理上建立一个完整的、自治的分析模型。

文献[6,8,10]从函数的可计算性分析、可解析性分析等角度讨论了计算涌现问题。其从传统时空复杂性分析理论出发,对于计算涌现的临界分析主要停留在单一的统计值上,因此,这些工作都没有将计算过程从执行步骤的角度来进行分析和建模,这与本文根据每一个图灵机磁带上空格的动作建立模型相比有所不同。

除了上述这些早期较有影响的理论研究之外,从特定的应用系统、计算环境或者网络交互等角度,也有研究者提出了部分结果[5,7,11-13]。这些研究主要针对分布式系统、网络交互、自主体交互等,提出了一些理想的计算模型,用以讨论复杂系统的特点。如文献[5]针对虚拟现实的设计问题,讨论了复杂度的变化动力学;文献[7]讨论了代理(Agent)设计中的交互复杂性问题;文献[11]提出了以交互为基本要素的通用计算模型“交互积”,并在此基础上提出了算法的交互复杂度指标,研究了交互复杂度和时间复杂度之间的关系;文献[12]提出了跨层次网络仿真的概念,关注微观个体的自组织集成方式,从社会经济系统的网络仿真中的主体交互进行了研究;文献[13]则将互联网看作一个不断演化的开放、复杂的系统总体,研究了网络协议层上的涌现统计现象;文献[14,15]等对多自主体代理之间的交互进行了涌现现象统计。总体来看,这些针对特定系统或者特定计算环境的复杂度原理分析,可以视作图灵机计算模型更高层次的讨论,因此本文的工作与其形成了互补,也可在下一步工作中讨论这些不同计算层面(如单机和网络)上的计算涌现之间的定量约束关系。

**结束语** 计算涌现的客观存在是计算机复杂性不断提高、计算模型不断改进的重要反映,但据 Church 假设<sup>[2]</sup>,可以认为迄今的计算能力并没有超出图灵机所划定的可计算范围。尽管计算模型本身的发展并没有得到本质上的突破,但不断复杂化的计算模型显然还是给我们准确理解其计算步骤

带来了困难。即使模型实现中通过处理器能力的提高和存储器空间的扩大能获得能力的提高,但我们能否通过分析复杂计算过程中的涌现现象,最终得到相应的临界(包括复杂性与时空程度之间的关系以及特殊过程,如不可计算等临界),在目前还是未知的。基于这种认识,本文初步探索了计算涌现的建模和分析。

运算的复杂性随着问题的复杂而增加的特点既可以通过计算模型本身的递归次数增加来体现,也可以通过描述 CGP 模型中不同机制的参与深度和参与次数来表现。机制的平均参与度体现了运算过程中各机制的参与程度和参与次数的分布。不可计算与平均参与度、参与深度和参与次数之间具有紧密的联系,而且这三者之间出现的相互约束关系是任何单个机制都无法表现出来的整体特性,这正是涌现的重要特征之一,也将是本文下一步的工作重点。可见,随着问题的难度陡然增加,通过机制的参与深度和参与次数的讨论方式将更容易抓住计算过程演化的特点。

#### 参考文献

- [1] Holland J. Emergence: From chaos to order [M]. New York: Perseus Books Group, 1998: 10-19
- [2] Boolos G, Burgess J, Jeffrey R. Computability and Logic, Fourth Edition: 影印版[M]. 北京: 电子工业出版社, 2003: 23-34
- [3] Boschetti F, Gray R. Emergence and Computability[J]. Emergence, Complexity and Organization, 2007, 9(1): 120-130
- [4] Cotsaftis M. Integrated control for high accuracy complex system[J]. Applied Mathematics and Computation, 2008, 206(2): 555-560
- [5] Crutchfield J. The Calculi of Emergence: Computation, Dynamics, and Induction[J]. Physica D, 1994, 75(1): 11-54
- [6] Bar-Yam Y. A Mathematical Theory of Strong Emergence Using Multi-scale Variety[J]. Complexity, 2004, 9(6): 15-24
- [7] Odell J. Agents and Emergence [J]. Distributed Computing, 1998, 13(10): 1-3
- [8] Testa B, Kier L. Emergence and Dissolution in the Self-organization of Complex Systems[J]. Entropy, 2000, 2(1): 1-25
- [9] Schaeffer J. Checkers Is Solved[J]. Science, 2007, 317(5844): 1518-1522
- [10] Fromm J. The emergence of complexity[M]. Kassel: Kassel university press, 2004: 18-57
- [11] 刘兴武, 徐志伟. 交互复杂度: 面向网络计算的复杂度指标[J]. 计算机研究与发展, 2004, 41(12): 2088-2094
- [12] 黄祥钊, 费奇, 欧阳敏, 等. 跨层次仿真中网络形成的涌现机制研究综述[J]. 计算机科学, 2009, 36(2): 26-29
- [13] 唐红, 黄鼎, 吴渝. 因特网突现行为研究综述[J]. 计算机科学, 2010, 37(5): 34-39
- [14] Prokopenko M, Boschetti F, Ryan A. An information-theoretic primer on complexity, self-organization, and emergence [J]. Complexity, 2009, 15(1): 11-28
- [15] 金士尧, 黄红兵, 范高俊. 面向涌现的多 Agent 系统研究及其进展[J]. 计算机学报, 2008, 31(6): 881-895