第 57 卷 第 6 期 2008 年 6 月
1000-3290/2008/57(06)/3359-08
物 理 学 报
ACTA PHYSICA SINICA
Vol.57,No.6,June,2008

TD-ERCS 离散混沌伪随机序列的复杂性分析*

孙克辉† 谈国强 盛利元
(中南大学物理科学与技术学院,长沙 410083)

(2007 年 10 月 13 日收到;2007 年 11 月 18 日收到修改稿)

采用相空间直接观察法和行为复杂性算法,系统地分析了新型 TD-ERCS 离散混沌系统产生的伪随机序列的复杂性,得出了其复杂性变化规律.在 Kolmogorov 复杂性基础上,应用经典的 Limpel-Ziv 算法,ApEn 算法和 PE 算法,从一维时间序列到多维相空间重构两方面计算了 TD-ERCS 离散混沌伪随机序列的复杂度大小.计算结果表明,TD-ERCS系统的行为复杂性高,而且该系统的复杂性大小随系统参数改变的变化范围小,是一个复杂性非常稳定的全域性离散混沌系统,其产生的混沌伪随机序列适合于信息加密或扩频通信.

关键词:混沌,混沌伪随机序列,TD-ERCS 系统,复杂度
PACC:0545

1.引 言

混沌在信息安全领域的应用已成为非线性科学研究的热点,而混沌系统复杂性分析是系统安全性能的一个非常重要研究方面.混沌系统的复杂性大小,直接关系到混沌密码系统的密码学性能.为了保证扩频通信的最大通信容量,实用的伪随机码应具有尽可能大的序列复杂度,因此,混沌伪随机序列的复杂度分析是混沌伪随机序列应用于信息加密和扩频通信的一个重要研究内容.

从 20 世纪 70 年代开始,复杂性问题的研究引起了国内外学者的广泛关注,并与非线性科学及其混沌动力学的复杂性研究交错在一起,在国际上形成了非线性科学和复杂性问题的研究热潮.1965 年,Kolmogorov 提出了复杂性的度量方法[1];1976 年,Lempel 和 Ziv 将 Kolmogorov 复杂性在计算机中实现[2];1991 年,Steven 和 Pincus 提出了计算序列的复杂性近似熵(approximate entropy,ApEn)算法[3].2002 年,Bandt 和 Pompe 提出了复杂性度量的排列熵(permutation entropy,PE)算法[4].目前,这些算法已经广泛应用于密码学[5]、医疗[6]和信号检测[7]等领域.

多年来,人们采用信息熵、Lyapunov 指数、分维数等参数来度量系统的混乱程度或无序程度,但这些方法都不能刻画系统的复杂性本质.因此,本文采用相空间观察法和行为复杂性算法分析由切延迟椭圆反射腔系统(tangent delay-ellipse reflecting cavity map system,TD-ERCS)产生的混沌伪随机序列的复杂性,为 TD-ERCS 系统在密码学与保密通信中的应用提供理论依据.

2.TD-ERCS 离散混沌系统的空间复杂性分析

2.1.TD-ERCS 离散混沌系统

2004 年,盛利元等人提出了 TD-ERCS 离散混沌系统[8],其映射关系为

$$
\begin{aligned}
x_n &= -\frac{2k_{n-1}y_{n-1}+x_{n-1}(\mu^2-k_{n-1}^2)}{\mu^2+k_{n-1}^2},\\
k_n &= \frac{2k'_n-k_{n-1}+k_{n-1}k_n'^2}{1+2k_{n-1}k'_n-k_n'^2} \quad n=1,2,3,\cdots, \qquad (1)\\
k'_{n-m} &= -\frac{x_{n-m}}{y_{n-m}}\mu^2 \quad (m\leqslant n),\\
y_n &= k_{n-1}(x_n-x_{n-1})+y_{n-1},
\end{aligned}
$$

其中,系统参数 $\mu\in(0,1]$,$|x_n|\leqslant 1$,$|y_n|\leqslant 1$;m 为整数,代表切线延迟;k'_{n-m} 为延迟 m 后椭圆切线的斜率;k_0 可由入射角 α 确定.显然,给定系统参数值 μ 和 m,初值 x_0 和 α,就可以求出 y_0,k_0 和 k'_0,从而

* 国家自然科学基金(批准号:60672041)资助的课题.

† E-mail:kehui_csu@hotmail.com

可以得到一组混沌序列 $\{x_n, k_n\}$. 当 $m=0$ 时,系统为 ERCS 系统;当切延迟 $m\geqslant 1$ 时,该系统称为 TD-ERCS 系统,此时,系统处于混沌状态.

2.2. TD-ERCS 系统的空间复杂性

TD-ERCS 系统的相空间结构及迭代点分布情况如图 1 和 2 所示. 系统初值 $x_0=0.7654$, $\alpha=0.9876$, 系统参数 $\mu=0.7123$, 迭代次数 $N=10000$.

图 1(a)可见,ERCS 有着独特的相空间结构,类似一个方形,且轨线呈不连续状态,而由图 2(a)可见,迭代点呈现明显的规律性,系统复杂性显然比较小. 由图 1(b)可知,切延迟 $m=1$ 的 TD-ERCS 系统相轨线为双峰结构,类似正弦曲线,由图 2(b)可见,迭代点分布与 ERCS 明显不同,遍布整个平面区域,无明显规律性. 由图 1(c)可见,切延迟 $m=2$ 的 TD-ERCS 系统的相轨线已经由 $m=1$ 时的双峰结构转化为离散点,系统演化更加复杂,系统产生的序列的复杂性也更大,由图 2(c)可见系统的迭代点遍历整个平面区域,具体的复杂性将通过复杂性计算进行讨论.

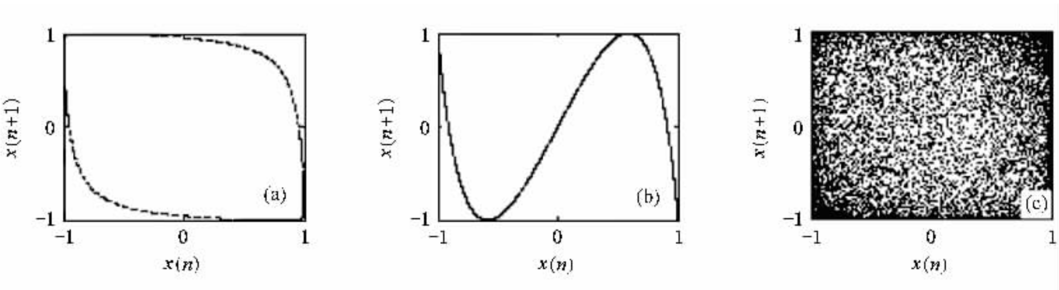

图 1 TD-ERCS 系统相空间结构 (a) $m=0$; (b) $m=1$; (c) $m=2$

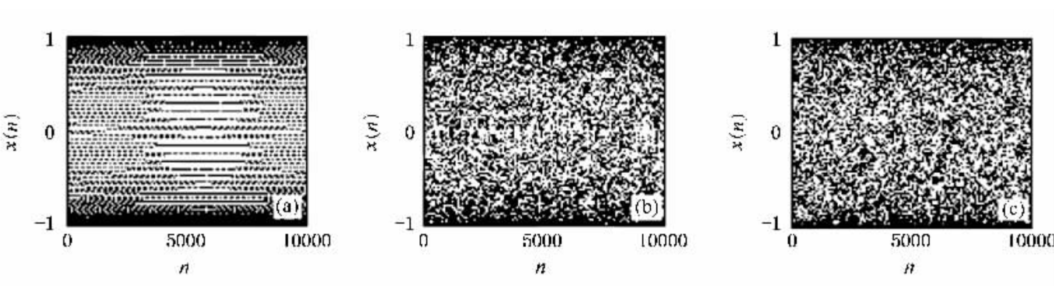

图 2 TD-ERCS 系统迭代点分布 (a) $m=0$; (b) $m=1$; (c) $m=2$

2.3. TD-ERCS 系统伪随机序列的产生

由混沌系统方程迭代产生的序列经过量化和判决后得到的序列称为混沌伪随机序列. 目前,多数文献采用的混沌序列粗粒化一般都是二次粗粒化方法,由于只考虑了大于平均值和小于平均值两种情况,二次粗粒化方法很可能会丢失原混沌动力学系统的一些有用信息. 为此本文采用多次粗粒化量化方法,其判决公式为[9]

$$\sigma_c(x) = j,$$
$$\sin^2\left[\frac{j\pi}{2K}\right] < x \leqslant \sin^2\left[\frac{(j+1)\pi}{2K}\right]$$
$$j = 0,1,2,\cdots,K-1. \tag{2}$$

对序列 $\{x_n\}$ 进行判决,即可以得到 $K=2^n$ 进制的混沌伪随机序列 $\{\sigma_c(x_n)\}_{n=0}^{\infty}$. 对于 TD-ERCS 系统产生的序列 $\{x_n\}\in(-1,1)$,必须先对其做线性变化,令 $x'_n = x_n/2+1/2$,使其值域变为 $(0,1)$. 由于此变换过程只有压缩和平移,故不会影响原混沌序列的性质.

3. TD-ERCS 离散混沌系统的行为复杂性

系统的行为复杂性是指系统产生的伪随机序列与随机序列的相似程度. 为了分析 TD-ERCS 系统混沌序列的复杂性,采用 Limpel-Ziv 算法,近似熵算法和排列熵算法,对 TD-ERCS 混沌伪随机序列的复杂性进行分析和讨论.

3.1. 基于 Limpel-Ziv 算法的复杂性

3.1.1. Limpel-Ziv 算法描述

给定时间序列 $S\{s_1, s_2, \cdots, s_n\}$，首先对它进行粗粒化处理，将其变为伪随机序列. 本文采用多次粗粒化量化算法. 量化后，不妨设重构的伪随机序列也为 $S(s_1, s_2, \cdots, s_n)$，对重构序列形成的字符串按一定的规则进行子串划分，规则如下：

在一个字符串 $S(s_1, s_2, \cdots, s_n)$后再加一个字符串 $Q(q_1, q_2, \cdots, q_n)$得到一个新字符串 SQ_V，令 SQ_V 是 SQ 减去最后一个字符所得字符串，再判断 Q 是否是 SQ_V 的一个子串，如果 Q 是 SQ_V 的一个子串，则把这个字符 Q 加到 S 后面，继续增长，再判断. 如果 Q 不是 SQ_V 的一个子串，则用'·'前后分开，下一步把'·'前的所有字符看成 S，重新构造 Q，重复以上过程直到结束. 序列的复杂性定义为由'·'确定的 S 的子串数目.

文献[2]的研究表明，几乎所有的复杂性 $c(n)$ 都趋向于一常数 b，即

$$\lim_{n\to\infty} c(n) = b = \frac{N}{\log_2 N}, \tag{3}$$

其中 N 代表序列的长度. 对于 k 个符号的序列，复杂性 $c(n)$收敛于

$$\lim_{n\to\infty} c(n) = b = \frac{N}{\log_k N}. \tag{4}$$

对 $c(n)$进行归一化，即

$$C(n) = c(n)/b. \tag{5}$$

用归一化的 $C(n)$来测度伪随机序列的复杂性变化，完全随机的序列 $C(n)$值趋向于 1，而周期性序列的 $C(n)$趋向于 0，其余情况(如混沌状态等)介于 0—1. 相对复杂度 $C(n)$反应了一个伪随机序列与随机序列的接近程度，某序列的 $C(n)$趋向于 1，则表明该序列越趋近随机序列，序列越复杂.

3.1.2. Limpel-Ziv 计算结果分析与讨论

1) 基于 Limpel-Ziv 算法的复杂性

应用 Limpel-Ziv 算法，对 ERCS 系统和 TD-ERCS 系统产生的伪随机序列的复杂性进行计算，取系统参数 $\mu = 0.7123$，计算结果如表 1 所示.

由表 1 可见，ERCS 系统的复杂性接近于 0；对于 $m = 1$ 的 TD-ERCS 系统，其复杂性大约为 0.55，复杂性相对较大；而对 $m = 2, 3$，TD-ERCS 系统复杂性达到了 0.90 以上，系统复杂性大. 与相空间直接观察法相比，对于 ERCS 系统，系统的相轨线呈明显的规律性，故复杂性非常小；对于 $m = 1$ 的 TD-ERCS 系统，其相轨线为类正弦曲线，复杂性较大；对于 $m = 2, 3$的 TD-ERCS 系统，其相轨线为离散无序状的点，相应的 Limpel-Ziv 复杂性最大. 这也验证了相空间分析的结论.

表 1 基于 Limpel-Ziv 算法的 TD-ERCS 伪随机序列复杂性

TD-ERCS 系统	Limpel-Ziv 复杂性			
	$N = 1000$	$N = 2000$	$N = 3000$	$N = 5000$
ERCS($m = 0$)	0.0864	0.0567	0.0359	0.0319
TD-ERCS($m = 1$)	0.5946	0.5739	0.5574	0.5513
TD-ERCS($m = 2$)	0.9335	0.9230	0.9189	0.9158
TD-ERCS($m = 3$)	0.9335	0.9321	0.9393	0.9371

此外，随着序列长度 N 的增长，复杂性的计算结果有减小的趋势，随着序列的不断增长，复杂性的计算结果逐渐趋于一个稳定值；在序列长度 4000 增大到 5000 的过程中，复杂性大小变化不大. 故当取 $N = 4000$ 时，Limpel-Ziv 算法计算结果趋于稳定.

2) 不同系统参数对伪随机序列复杂性大小的影响

为了更好地分析 TD-ERCS 混沌伪随机序列的复杂性，研究了混沌系统随系统参数变化时，系统复杂性变化情况. 计算结果分别如图 3 和 4 所示. 序列的长度均取 $N = 1000$，切延迟 m 为整数.

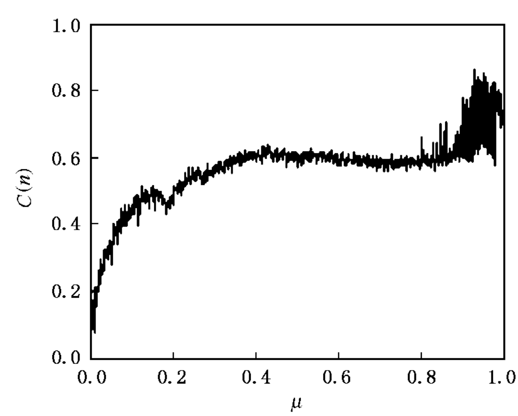

图 3 TD-ERCS 系统 Limpel-Ziv 复杂性随 μ 变化

图 3 中，对于切延迟 $m = 1$ 的 TD-ERCS 系统，随着压缩因子 μ 的增加，系统的复杂性呈现增大的趋势，但中间有一段相对平稳的过程，在靠近 1 附近，系统的复杂性又有着一个突然增大的趋势. 在图 4 中，对于 $\mu = 0.7123$ 的 TD-ERCS 系统，随着切延迟 m 的变化，系统的复杂性非常稳定，除了 $m = 1$ 的复杂性相对较小，大约为 0.6 外，其余点复杂性稳定在

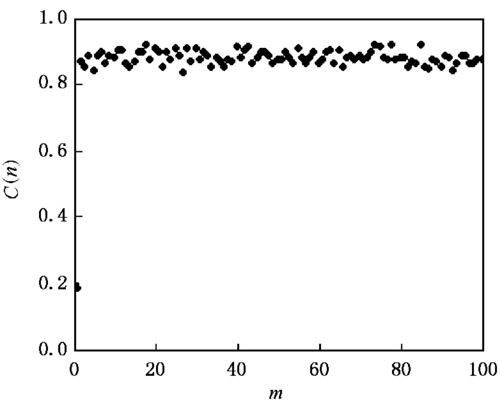

图 4 TD-ERCS 系统 Limpel-Ziv 复杂性随 m 变化

0.92 左右,这说明当系统参数 m 变化时,TD-ERCS 系统的复杂性大,且稳定.

3.2. 基于近似熵算法的复杂性研究

1991 年,Steven 和 Pincus 提出了一种度量序列的复杂性的近似熵(ApEn)算法,文献[9]提出了用近似熵算法计算混沌运动的测度熵,作为衡量混沌伪随机序列复杂性的标准.近似熵算法从 Limpel-Ziv 算法发展而来,Limpel-Ziv 算法从一维角度直接统计序列的随机程度来衡量整个序列的复杂性,而 ApEn 算法则从多维空间来讨论序列的复杂性.近似熵算法利用边缘条件概率的统计方式统计序列的随机程度,用相邻轨道的变化程度体现整个序列的复杂性,所以与 Limpel-Ziv 算法相比,ApEn 算法更能分析序列的内在复杂性.该算法的特点是可用较短的观察序列有效地判断混沌伪随机序列的复杂度.

3.2.1. ApEn 算法描述

近似熵算法描述如下:

第 1 步 对于一个长度为 N 的序列样本空间 $\{x_1, x_2, x_3, \cdots, x_N\}$,构造 p 维向量 $\{X_1, X_2, \cdots, X_i, \cdots, X_{N-p+1}\}$,其中

$$X_i = [x_i, x_{i+1}, \cdots, x_{i+p-1}], 1 \leqslant i \leqslant N - p + 1.$$

第 2 步 定义 p 维向量 X_i 和 X_j 的最大距离:

$$d[X_i, X_j] = \max_{k=1,2,\cdots,p} \{|x_{i+k-1} - x_{j+k-1}|\}. \tag{6}$$

第 3 步 计算满足与第 i 个 p 维向量 X_i 的最大距离小于 r 的概率

$$C_i^p(r) = \frac{\text{满足 } d[X_i, X_j] \leqslant r \text{ 的 } j \text{ 的个数}}{N - p + 1}. \tag{7}$$

第 4 步 利用 $C_i^p(r)$ 求出 $\Phi^p(r)$,有

$$\Phi^p(r) = (N - p + 1)^{-1} \sum_{i=1}^{N-p+1} \ln C_i^p(r). \tag{8}$$

第 5 步 ApEn 定义为

$$\mathrm{ApEn}(p, r, N) = \Phi^p(r) - \Phi^{p+1}(r). \tag{9}$$

计算近似熵时,先确定两个参数 p 和 r,p 为嵌入维数,r 为分辨率参数,且在整个计算过程中固定不变.嵌入维数 p 的最大值由观察空间的长度 N 决定,p 越大,ApEn 越接近测度熵,但 p 越大,要使计算结果的精确度高,需要的序列长度 N 也非常大,计算比较困难,实际计算中,一般选择 $p = 2$.分辨率参数 r 决定了该算法的分辨率,r 越小,ApEn 的分辨率就越高,在实际计算中,r 应该合理取值.Pincus 等人[3]建议取 $p = 2$,$r = 0.1$—0.25 SD(SD 是原始数据 $\{x_i\}$($i = 1, 2, \cdots, N$)的标准偏差).

对于任意的 p,K 进制序列 $X_N = \{x_1, x_2, \cdots, x_i, \cdots, x_N\}$ 满足 $0 \leqslant \lim_{n\to\infty} \mathrm{ApEn}(p, r, N) \leqslant \ln K$[9].如,对于 8 进制,ApEn 的最大值为 2.079.

3.2.2. ApEn 计算结果分析与讨论

1) TD-ERCS 系统的近似熵

对 ERCS 系统和不同参数的 TD-ERCS 进行 ApEn 值计算,系统参数 $\mu = 0.7123$,舍弃前 200 个点,选取不同的序列长度 N 和分辨率参数 r,得到计算结果如表 2 所示.

表 2 TD-ERCS 混沌伪随机序列的 ApEn 复杂性

TD-ERCS 系统	ApEn(2, r, N)值			
	r	$N = 500$	$N = 1000$	$N = 3000$
Logistic($\mu = 4$)[10]	0.15			0.69
ERCS($m = 0$)	0.3	0.5324	0.5293	0.5221
TD-ERCS($m = 1$)	0.3	0.9448	0.9348	0.9213
TD-ERCS($m = 2$)	0.5	0.9706	0.9751	0.9592
TD-ERCS($m = 3$)	0.5	0.9439	0.9560	0.9541

表 2 中,对于 ERCS 系统,ApEn = 0.5221,与前面的计算结果有些差异,根据 Limpel-Ziv 算法的分析并与相空间结构图比较,该系统是一个比较规则的系统,产生的序列也明显具有周期性,但是,计算结果为什么会大于 0?文献[11]已证明了该系统是长周期系统,系统较为复杂,此外,在 Limpel-Ziv 算法中,只对序列的一维状态进行了分析,故体现出 ERCS 序列的规律性,而 ApEn 算法,在多维状态下分析序列的复杂程度,体现出 ERCS 序列的内在复杂性.由表 2 可见,对于 TD-ERCS 序列簇,其复杂性高,而且切延迟 $m = 2$ 以上的 TD-ERCS 系统的复杂程度要明显高于 $m = 1$ 的 TD-ERCS 系统.对于 Logistic 映射,$N = 3000$ 时,ApEn 值为 0.69[10],比

TD-ERCS系统的复杂性低.

2）不同系统参数对系统 ApEn 大小的影响

计算中，取 $p=2$，$r=0.3$，序列迭代点的长度 $N=1000$. 计算结果如图 5 和 6 所示.

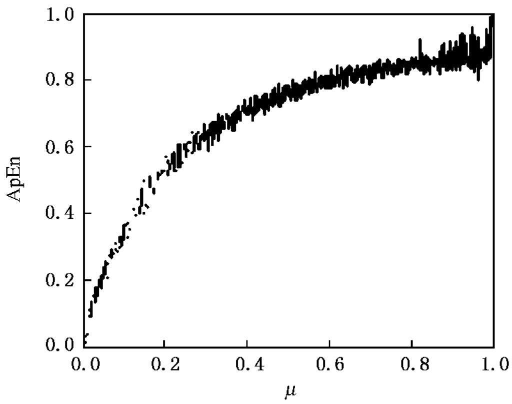

图 5　TD-ERCS 系统 ApEn 值随 μ 变化

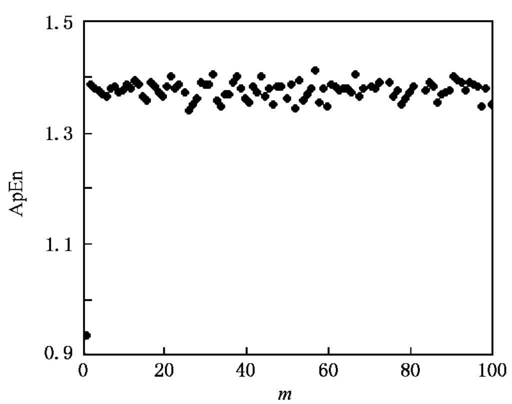

图 6　TD-ERCS 系统 ApEn 值随 m 变化

图 5 中，TD-ERCS 切延迟参数 $m=1$，压缩因子 $\mu\in(0,1]$. 当 μ 在 0.2 附近变化时，ApEn 复杂性出现有模糊的点，出现了计算结果不存在的情况，这是由系统结构所致，还是预示着新的现象，尚不得而知，有待进一步研究. 当 μ 进一步增大时，TD-ERCS 系统的复杂性慢慢增加，当 $\mu>0.6$ 时，TD-ERCS 系统的 ApEn 复杂性趋于稳定，当 $\mu=1$ 时，TD-ERCS 系统的复杂性达到最大值.

图 6 中，系统压缩因子 $\mu=0.7123$，切延迟 m 为 1—100 的整数. 由图 6 可见，只有 m 较小时的 ApEn 复杂性在 0.93 左右，而对于其他的切延迟 m，系统的复杂性在 1.38 到 1.42 之间波动. 可见，混沌伪随机序列的复杂性稳定.

3.3. 基于排列熵算法的复杂性

排列熵(PE)复杂性算法也是建立在 Kolmogorov 复杂性基础上，运用 Shannon 信息熵的概念，来统计和计算序列的复杂性. 与前面介绍的 Limpel-Ziv 算法和 ApEn 算法相比，排列熵复杂性与近似熵类似，是在多维空间中考虑系统复杂性变化情况，不同的是，该算法是利用多维重构空间的相似程度来衡量整个序列的复杂性，分析所有嵌入维的相似特性. 该算法能更好的检测复杂动力学系统的复杂性.

3.3.1. PE 算法描述

PE 算法描述如下：

第 1 步　由系统方程迭代得到长度为 N 的离散时间序列 $\{x_i, i=1,2,\cdots,N\}$，对 $\{x_i\}$ 进行相空间重构，得到重构后的序列

$$X(i)=[x(i),x(i+\tau),\cdots,x(i+(p-1)\tau)]\quad 1\leqslant i\leqslant N-p+1, \tag{10}$$

式中 p 和 τ 分别为嵌入维数和延迟时间. 这里使用最大重叠情形，令 $\tau=1$，即将每个子序列向后移动一个数据点得到下一个子序列.

第 2 步　将 $X(i)$ 的第 p 个重构分量 $[x(i), x(i+\tau),\cdots,x(i+(p-1)\tau)]$ 按照升序重新进行排列，得到

$$[x(i+(j_1-1)\tau)\leqslant x(i+(j_2-1)\tau)\leqslant\cdots\leqslant x(i+(j_p-1)\tau]\quad 1\leqslant j\leqslant N-P+1. \tag{11}$$

若存在序列某两个值的 $x(i)$ 相等，就按照 j 值的大小来进行排序. 所以，任意一个向量 $X(i)$ 都可以得到一组符号序列

$$A(g)=[j_1,j_2,\cdots,j_p]\quad 1\leqslant g\leqslant N-p+1. \tag{12}$$

第 3 步　p 个不同的符号 $[j_1, j_2,\cdots,j_p]$ 一共有 p！种不同的排列，也就是一共有 p！种不同的符号序列，符号序列 $A(g)$ 是其中的一种. 将所有排列相同的符号序列 $A(g)$ 归为一组，在 $N-p+1$ 组序列中一共有 k 组不同的符号序列，设每组序列的个数分别为 $\mathrm{Num}_1, \mathrm{Num}_2,\cdots,\mathrm{Num}_k$，则每一种符号序列出现的概率 $P_1, P_2,\cdots,P_k$ 为

$$P_k=\frac{\mathrm{Num}_k}{N-p+1}. \tag{13}$$

第 4 步　时间序列 $\{x_i, i=1,2,\cdots,N\}$ 的 k 种不同符号序列的排列熵就可以按照 Shannon 信息熵的形式定义为

$$H(p)=-\sum_{i=1}^{k}P_k\ln P_k. \tag{14}$$

第 5 步 理论上，当 $P_k = 1/p!$ 时，$H(p)$ 达到最大值 $\ln(p!)$，实际当中，$H(p) \leqslant \ln(N-p+1)$. 为了方便，通常将 $H(p)$ 用 $\ln(N-p+1)$ 进行归一化处理，即

$$0 \leqslant h(p) = \frac{H(p)}{\ln(N-p+1)} \leqslant 1. \tag{15}$$

若计算混沌伪随机序列的排列熵，应在第 1 步中加入对混沌序列的量化，使 $\{x_i, i = 1,2,\cdots,N\}$ 变成混沌伪随机序列. 这里，仍然采用多次粗粒化量化方法对原序列进行量化. 然后，对混沌伪随机序列进行重构. 因为混沌伪随机序列本身已有一定的大小关系，故在第 2 步中只需要统计出数目相同的序列个数 Num_k，直接进入第 3 步.

显然，$h(p)$ 变化体现了序列的随机性. $h(p)$ 越小，序列越规则，序列的复杂性越小；$h(p)$ 越大，序列越随机，序列复杂性越大. 文献[6]中讨论了序列长度 N，以及 p 选取时对计算结果的影响. N 的选取不能太小，否则会失去其统计学意义，一般 $1000 \leqslant N \leqslant 10000$；$p$ 的取值范围一般为 $3 \leqslant p \leqslant 15$.

3.3.2. PE 计算结果分析与讨论

1) 固定参数的 PE 值

对混沌序列进行 8 进制粗粒化量化，得到混沌伪随机序列，然后计算其 PE 复杂性. 设嵌入维数 $p = 5$，系统参数 $\mu = 0.7123$，分别计算长度为 $N = 1000$，2000，4000，5000 的序列的复杂性，所得计算结果如表 3 所示.

表 3 TD-ERCS 混沌伪随机序列的 PE 值

TD-ERCS 系统	PE 值			
	$N = 1000$	$N = 2000$	$N = 4000$	$N = 5000$
Logistic($\mu = 4$)[12]		0.51		
ERCS($m = 0$)	0.5548	0.5103	0.4702	0.4582
TD-ERCS($m = 1$)	0.8348	0.7804	0.7212	0.7043
TD-ERCS($m = 2$)	0.9964	0.9886	0.9792	0.9757
TD-ERCS($m = 3$)	0.9930	0.9916	0.9868	0.9841

表 3 中，各混沌伪随机序列的复杂性值都比较大，各个系统的区分度也变大. 这是因为，当序列经过量化后，所得到的不再是算法中描述的大小顺序关系，而是确定的数字排列顺序关系. 也就是说，在统计 Num_k 时，对相同序列的统计更加准确. 所以，计算得到的 P_k 分布更加均匀，计算结果也就更大. 这也说明，PE 算法对量化后的结果计算更加准确. 对于 Logistic 映射，$N = 2000$ 时，PE 值为 0.51[12]，比 TD-ERCS 系统的复杂性低.

2) 不同系统参数的 PE 值的变化规律

为了更好的对比分析系统的 PE 值随系统参数的变化规律，计算了 TD-ERCS 系统排列熵值随系统参数变化情况. 混沌伪随机序列的 PE 复杂性随参数变化情况如图 7 和图 8 所示，混沌序列的长度 $N = 4000$，嵌入维 $p = 5$.

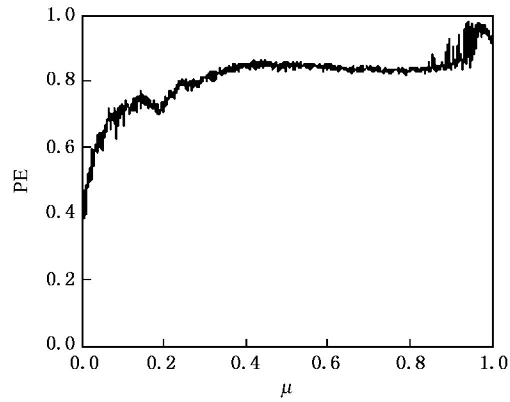

图 7 TD-ERCS 序列的 PE 值随参数 μ 的变化

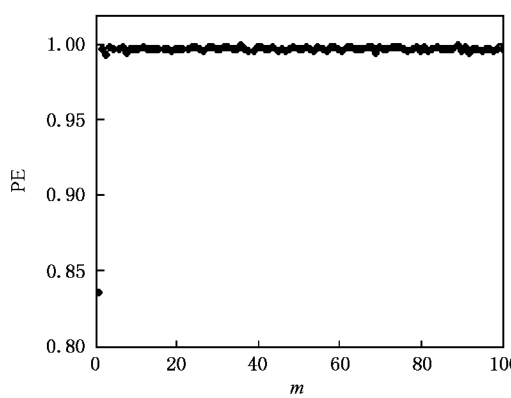

图 8 TD-ERCS 序列的 PE 值随参数 m 的变化

图 7 所示的参数 $m = 1$ 为的 TD-ERCS 系统，随着参数 μ 的变化，在 $\mu = 0.2$ 附近复杂性有一个先下降后增加的现象，类似于一个"大雁"的形状. 图 8 所示的 $\mu = 0.7123$ 的 TD-ERCS 系统的复杂性相当稳定. 可见，量化后的混沌伪随机序列保持了原混沌序列的复杂性，并使系统局部区域的复杂性得到放大.

3.4. 三种复杂性算法比较

3.4.1. 三种复杂性算法的异同

通过对三种算法的计算原理和物理意义上的分析可知，每种算法都有相似之处，也有其各自的特点.

相似之处在于三种算法都建立在 Kolmogorov 复

杂性的基础上，都是对序列的随机程度的描述. 序列的随机程度越大，产生序列的计算机程序越长. 另外，都使用了信息熵的概念对计算机程序的长度、出现的概率大小进行统计，所得到的信息量进行统计，从而得到了复杂性的具体数值. 对 Limpel-Ziv 算法，序列越复杂，序列要不断产生"新词"，所需要的计算机程序也越长，序列的复杂性也越大；对 ApEn 算法，序列越复杂，"相似"的重构序列也越少，必然要求增长序列的长度和放大分辩率参数 r 来得到合理的计算结果，所需的程序也越长；对于 PE 算法，序列越复杂，所得到的排列的个数越多，进行统计的次数也越多，从而得到的熵值也越大. 因此，这三种算法都是建立在 Kolmogorov 复杂性和信息熵的基础上的复杂性算法.

三种算法也各有特点. 对于 Limpel-Ziv 算法，只是在一维时间尺度上对系统的复杂性进行统计，没有体现不同嵌入维之间序列的复杂性大小关系，除了序列长度外，算法中没有涉及到主观选择的参数. 对于 ApEn 算法，体现的是不同嵌入维变化时序列的复杂性大小，由于计算过程中涉及到嵌入维和分辩率参数的选取，计算结果随主观因素而有所变化. 而对于 PE 算法，则是确定维数时的序列的复杂性大小关系，计算中也要对嵌入维数进行选取，这与 ApEn 算法中的嵌入维是同一个概念，只不过这里是对一维进行计算，选取的嵌入维可以稍大(在本文中，计算 ApEn 的嵌入维为 2，而 PE 复杂性计算的嵌入维为 5).

3.4.2. 计算结果对比

应用 Limpel-Ziv 算法、ApEn 算法和 PE 算法，计算了 TD-ERCS 系统产生的混沌伪随机序列的复杂性，系统参数 $\mu = 0.7123$，计算结果如表 4 所示.

表 4　不同算法的 TD-ERCS 伪随机序列的复杂性

TD-ERCS 系统	Limpel-Ziv	ApEn	PE
	$N = 5000$	$N = 3000$	$N = 4000$
ERCS($m = 0$)	0.0319	0.5221	0.4582
TD-ERCS($m = 1$)	0.5513	0.9213	0.7043
TD-ERCS($m = 2$)	0.9158	0.9592	0.9757
TD-ERCS($m = 3$)	0.9371	0.9541	0.9841

表 4 中 Limpel-Ziv 算法识别了系统的长周期现象，所以得到的计算结果比较小. 而在 ApEn 算法和 PE 算法中，运用了时间序列的重构方法，将序列进行了重构，体现了序列在多维情况下的复杂程度，因此得到的计算结果偏大. 由表 4 可见，应用不同复杂性算法计算的结果是正确的.

4. 结　论

分别采用 Limpel-Ziv 算法、ApEn 算法和 PE 算法三种复杂性算法分析了 TD-ERCS 离散混沌系统产生的混沌伪随机序列的复杂性；讨论了 TD-ERCS 系统中系统参数变化对系统序列复杂性的影响. 研究表明，这三种算法都是有效的复杂性算法，计算出由 TD-ERCS 系统产生的混沌伪随机序列的复杂性大；混沌伪随机序列的复杂性随系统参数 μ 的增大而增加，在 $\mu = 1$ 附近趋于最大值；随延迟系数 m 的增加而保持稳定. 计算中发现，当 $m = 1$，$\mu = 0.2$ 时，TD-ERCS 伪随机序列的 Limpel-Ziv 和 PE 复杂性出现"大雁"状波动，而 ApEn 值不存在现象，其具体原因尚不清楚，作者将另文研究. 总之，TD-ERCS 系统是一个复杂性大的新型离散混沌系统，在密码学、保密通信等领域有着很好的应用前景.

[1] Kolmogorov A N 1965 *Problem of Information Transission* **35** 1546

[2] Lempel A, Ziv J 1976 *IEEE Trans*. **IT-22** 75

[3] Steven M, Pincus S 1991 *Mathematics* **88** 2297

[4] Bandt C, Pompe B 2002 *Phys*. *Rev*. *Lett*. **88** 174102

[5] Amigó J M, Kocarev L, Szczepanski J 2006 *Phys*. *Lett*. A **355** 27

[6] Zhang H X, Zhu Y S, Niu J H, Tong S B 2000 *Acta Phys*. *Sin*. **49** 1416 (in Chinese) [张红煊、朱贻盛、牛金海、童善保 2000 物理学报 **49** 1416]

[7] Wu W L, Zhu N 2003 *J*. *Elect*. *Info*. *Tech*. **25** 678 (in Chinese) [吴为麟、朱　宁 2003 电子与信息学报 **25** 678]

[8] Sheng L Y, Sun K H, Li C B 2004 *Acta Phys*. *Sin*. **53** 2871 (in Chinese) [盛利元、孙克辉、李传兵 2004 物理学报 **53** 2871]

[9] Cai J P, Li Z, Song W T 2003 *Acta Phys*. *Sin*. **52** 1871 (in Chinese) [蔡觉平、李　赞、宋文涛 2003 物理学报 **52** 1871]

[10] Wang Y X, Weng Y F, Zheng D L 2006 *J*. *Beijing Tech*. *Business Univ*. (*Natural Science Edition*) **24** 38 (in Chinese) [王云雄、翁贻方、郑德玲 2006 北京工商大学学报(自然科学版) **24** 38]

[11] Sheng L Y, Jia W Y 2006 http://www.paper.edu.cn/ (in Chinese) [盛利元 2006 中国科技论文在线]

[12] Hou W, Feng G L, Dong W J 2006 *Acta Phys*. *Sin*. **55** 2663 (in Chinese) [侯　威、封国林、董文杰 2006 物理学报 **55** 2663]

The complexity analysis of TD-ERCS discrete chaotic pseudo-random sequences*

Sun Ke-Hui† Tan Guo-Qiang Sheng Li-Yuan

(*School of Physics Science and Technology*, *Central South University*, *Changsha* 410083, *China*)

(Received 13 October 2007; revised manuscript received 18 November 2007)

Abstract

By observing the phase diagram and using the behavior complexity algorithm, the complexity of chaotic pseudo-random sequences generated by the new TD-ERCS discrete chaotic system is analyzed in detail, and the rules of complexity variety are investigated. Based on the Kolmogorov complexity, from one-dimensional time series to multidimensional phase space restructure, the complexity values of TD-ERCS discrete chaotic pseudo-sequences are calculated by using the Limpel-Ziv algorithm, ApEn algorithm and PE algorithm, respectively. The results show that the behavior complexity of TD-ERCS system is high, and the complexity value changes a little with the change of the parameters of TD-ERCS system. TD-ERCS system is a discrete chaotic system with the steady complexity, and the pseudo-random sequences generated by TD-ERCS are suitable for use in information encryption and spread spectrum communications.

Keywords: chaos, chaotic pseudo-random sequence, TD-ERCS system, complexity

PACC: 0545

* Project supported by the National Natural Science Foundation of China (Grant No.60672041).

† E-mail: kehui_csu@hotmail.com