Tạp chí Tin học và Điều khiển học, T. 18, S. 2 (2002), 128–134

MỘT PHƯƠNG PHÁP KIỂM TRA TÍNH NGẪU NHIÊN CỦA DÃY NHỊ PHÂN

NGUYỄN THỊ HẢI YẾN

Abstract. The new test is a combination of Ziv–Lempet algorithm and the statistic method, which is applied to test randomness of finite binary sequences. The efficiency of the new test will be identify when the test is compared which other tests. In this paper, the new test is compared on the five basic criteria.

Tóm tắt. Test mới là sự kết hợp thuật toán Ziv–Lempet với phương pháp thống kê. Nó được ứng dụng để kiểm tra tính ngẫu nhiên của một dãy nhị phân. Chúng ta sẽ được kết quả của test mới khi so sánh nó với các phương pháp khác, trong bài này, test mới được so sánh với 5 tiêu chuẩn cơ bản.

1. GIỚI THIỆU

Như chúng ta đã biết, có rất nhiều ứng dụng cần đến dãy ngẫu nhiên 0-1, chẳng hạn như ứng dụng để mã hóa, hoặc các phương pháp ngẫu nhiên. Trên thực tế, đã có rất nhiều bài viết về nghiên cứu lĩnh vực này. Trong bài này, trên cơ sở khảo sát độ phức tạp Ziv–Lempet trên một khối cỡ nhỏ và kết hợp với phương pháp thống kê, chúng tôi đưa ra một test mới để kiểm tra tính ngẫu nhiên của dãy 0-1.

2. GIỚI THIỆU THUẬT TOÁN ZIV–LEMPET

Trong các bài [1] và [4] các tác giả đã đưa ra một loại độ phức tạp Ziv–Lempet. Sau đây chúng tôi tóm tắt một số ý chính trong các bài báo đó nhằm giải thích cho phần sau.

2.1. Một số khái niệm

Bài báo [1] đã đề xuất và khai thác một hướng mới về độ phức tạp của một dãy cụ thể với việc xây dựng dần các mẫu mới trong dãy đã cho. Theo các tác giả, không tồn tại một thước đo tuyệt đối về độ phức tạp, nên họ đề xuất việc đánh giá độ phức tạp của một dãy hữu hạn theo cách nhìn của một máy tự đọc nhị phân đơn giản, khi nó quét một dãy n số đã cho $S = s_1 s_2 \dots s_n$ từ trái sang phải, thêm từ mới vào bộ nhớ của nó mỗi khi nó phát hiện một xâu con của các số liên tục chưa thấy. Kích thước của từ điển tạo ra, và tỉ lệ mà các từ mới phát hiện ra dọc theo dãy S, là những hợp phần cơ sở trong đánh giá độ phức tạp của dãy S.

Cho A^* là tập hợp gồm tất cả các dãy độ dài hữu hạn trên một bộ chữ cái hữu hạn A. Cho $l(S)$ là độ dài của $S \in A^*$ và

$$A^n = \{S \in A^* \mid l(S) = n\}, \quad n \geq 0.$$

Dãy trống $\emptyset$ có nghĩa là "dãy" có độ dài không, được coi là một phần tử của A^*. Một dãy $S \in A^*$ được chỉ rõ đầy đủ bằng cách viết $S = s_1 s_2 \dots s_n$; khi S được tạo lập từ một phần tử $a \in A$, chúng ta viết $S = a^n$. Để chỉ một xâu con của S bắt đầu từ vị trí i và kết thúc tại vị trí j, chúng ta viết $S(i, j)$, có nghĩa là khi $i \leq j$, $S(i, j) = s_i s_{i+1} \dots s_j$ và $S(i, j) = \emptyset$ đối với $i > j$.

Việc ghép $Q \in A^m$ và $R \in A^n$ tạo ra dãy mới $S = QR = q_1 q_2 \dots q_m r_1 r_2 \dots r_n \in A^{m+n}$, với $Q = S(1, m)$ và $R = S(m + 1, m + n)$. Chúng ta sử dụng kí hiệu $S^2 = SS$ để chỉ việc ghép S với chính nó. Nói chung $S^0 = \emptyset$ và $S^i = S^{i-1} S$, $i \geq 1$.

Q được gọi là *phần đầu* (tiền tố - prefix) của $S \in A^*$, và S là *phần mở rộng* của Q nếu tồn tại số nguyên i sao cho $Q = S(1, i)$; phần đầu Q và phần mở rộng S được gọi là *thật sự* nếu $l(Q) < l(S)$.

Khi độ dài của S không được chỉ rõ, một cách thuận tiện để xác định các phần đầu của S là dùng toán tử π, theo đó $S\pi^i = S(1, L(S) - i)$, với $i = 0, 1, \ldots$, cụ thể: $S\pi^0 = S$ và $S\pi^i = \emptyset$ đối với $i \geq l(S)$.

Một từ điển của dãy S, được biểu diễn bằng $v(S)$, là tập con của A^* được tạo ra từ tất cả các xâu con, hay từ $S(i, j)$ của S, ví dụ:

$$v(0010) = \{\emptyset, 0, 1, 00, 01, 10, 001, 010, 0010\}.$$

Một từ $Q \in v(S)$ được gọi là từ tận cùng phải của S nếu Q không phụ thuộc vào từ điển của bất kỳ phần đầu thật sự nào của S. Tập tất cả các từ tận cùng phải của S, được biểu diễn bằng $e(S)$, được gọi là *từ điển các từ tận cùng phải của S*, ví dụ:

$$e(0010) = \{10, 010, 0010\}.$$

Chúng ta nói rằng phần mở rộng $R = SQ$ của S là *được tái tạo* từ S và viết $S \to R$, nếu $Q \in v(R\pi)$. Nghĩa là, Q là một từ của *phần đầu thật sự* của R. Vị trí p của S sao cho $Q = R(p, l(Q) + p - 1)$ được gọi *con trỏ* để tái tạo $S \to R$. Rõ ràng S *được tái tạo* từ S vì $S = S\emptyset$ và $\emptyset \in v(S\pi)$.

Chúng ta nói rằng một dãy không trống S được *tạo ra* từ phần đầu $S(1, j)$ của nó nếu $S(1, j) \to S\pi$ và $j < l(S)$, nghĩa là $S(j + 1, l(S) - 1) \in v(S\pi^2)$. Tính tạo được của S từ $S(1, j)$ được thể hiện bằng kí hiệu $S(1, j) \Rightarrow S$, và $S(1, j)$ được coi là *cơ sở* của S. Nhận xét rằng mỗi dãy không trống đều có cơ sở (ví dụ: $S\pi$).

Sự kiện mỗi dãy không trống S có thể được tạo từ một số phần đầu đúng của S gợi ý nên lệnh $Q \Rightarrow S$ như một cơ chế tạo S từ Q. Thật ra mỗi dãy không trống S có thể được hiểu như là kết quả tạo cuối cùng của quá trình xây dựng tự phân định từ điển lặp lại với bước đầu tiên là $S(1, 0) \Rightarrow S(1, 1)$, (trong đó $S(1, 0) = \emptyset$, $S(1, 1) = s_1$) và khi tạo ra $S(1, h_i)$ từ bước i, tiếp tục thực hiện $S(1, h_i) \Rightarrow S(1, h_{i+1})$ và tiếp tục thực hiện như vậy nhiều nhất là $l(S)$ bước, toàn bộ S được tạo ra. Chúng ta gọi cơ chế từng bước tạo ra S như vậy là *quá trình tạo* của S, và kết quả $S(1, h_i)$ của bước thứ i là trạng thái thứ i của quá trình.

2.2. Khái niệm độ phức tạp Ziv–Lempet

Xem xét quá trình tạo gồm m bước của dãy S và cho $S(1, h_i)$, $i = 1, 2, \ldots, m$ là m trạng thái của quá trình. Nhớ rằng $h_1 = 1$ và $h_m = l(S)$. Phân tích S thành:

$$H(S) = S(1, h_1)S(h_1 + 1, h_2) \ldots S(h_{m-1} + 1, h_m)$$

được gọi là *lịch sử* (*tạo*) của S và m từ $H_i(S) = S(h_{i-1} + 1, h_i)$, $i = 1, 2, \ldots, m$, với $h_0 = 0$, được gọi là các *thành phần* của $H(S)$.

Thành phần $H_i(S)$ và bước tạo tương ứng $S(1, h_{i-1}) \Rightarrow S(1, h_i)$ được gọi là *toàn diện* nếu *không có* $S(1, h_{i-1}) \to S(1, h_i)$. Một lịch sử (hoặc quá trình tạo) được gọi là toàn diện nếu mỗi thành phần của nó, *ngoại trừ thành phần cuối cùng* là toàn diện. *Lịch sử toàn diện* của dãy S được thể hiện bằng $E(S)$. Ví dụ, lịch sử toàn diện của dãy $S = 0001101001000101$ là một phân tích của S thành:

$$0.001.10.100.1000.101$$

với các thành phần kế tiếp nhau được ngăn cách bằng dấu chấm và phần cuối của dãy không có dấu chấm thể hiện là thành phần cuối cùng không toàn diện.

Độ phức tạp Ziv–Lempet $c(S)$ cho dãy S được đề xuất như sau: Cho $c_H(S)$ thể hiện số các thành phần trong lịch sử $H(S)$ của S. Khi đó:

$$c(S) = \min\{c_H(S)\}$$

với sự tối thiểu hóa được thực trên tất cả lịch sử của S. Nói cách khác $c(S)$ là số nhỏ nhất có thể của các bước để tạo ra S theo quá trình tạo ở trên.

Định lý 1. *$c(S) = c_E(S)$, với $c_E(S)$ là số các thành phần trong $E(S)$, là lịch sử toàn diện của S.*

Chứng minh Định lý 1 có thể xem trong [1].

Độ phức tạp Ziv–Lempet của các dãy thường được tính dựa trên định lý này.

2.3. Thuật toán để tính độ phức tạp Ziv–Lempet

Từ khái niệm độ phức tạp Ziv–Lempet trong [1], chúng ta có thể xây dựng được thuật toán để tính độ phức tạp Ziv–Lempet [4]. Cho dãy $S = s_1 s_2 \dots s_n$ có độ dài n, độ phức tạp Ziv–Lempet (DPT) được tính theo các bước sau:

Bước 1: Đặt $i = 1$, $K[1] = 1$.

Bước 2: Đặt $m = K[i]$.

Bước 3: Nếu $m \geq n$ thì DPT $= i$, thuật toán dừng.

Bước 4: Nếu $m < n$ thì

4.1. Xác định tập $J = \{1 \leq j \leq m : S_j = S_{m+1}\}$, đặt $l = 1$.

4.2. a: Kiểm tra J ($J = \emptyset$?) có phải tập trống không?

Nếu $J = \emptyset$ thì $K[i+1] = K[i] + 1$, $i = i + 1$, quay về bước 2.

b: Nếu $J \neq \emptyset$ và $m + l + 1 \leq n$ thì:

b1: xóa mọi $j \in J$ thỏa $S_{j+1} \neq S_{m+l+1}$.

b2: Nếu $J = \emptyset$ thì đặt $K[i+1] = K[i] + l + 1$, $i = i + 1$, và quay về thực hiện bước 2.

b3: Nếu $J \neq \emptyset$ thì $l = l + 1$, quay về phần b.

c: Nếu $J \neq \emptyset$ và $m + l + 1 > n$ thì DPT $= i + 1$, thuật toán dừng.

2.4. Phân bố của độ phức tạp Ziv–Lempet trên các dãy nhị phân

Cho đến nay chưa có tài liệu nào công bố kết quả nghiên cứu về phân bố lý thuyết độ phức tạp Ziv–Lempet trên các khối độ dài n của dãy nhị phân. Trong bài [3], các tác giả đã khảo sát thống kê thực nghiệm về phân bố đó. Trong bài này, chúng tôi đưa ra phân bố chính xác của độ phức tạp Ziv–Lempet của các khối nhị phân độ dài n, với $n = 8$, 16, 24, 32 bít, trên cơ sở lập trình tính toán trên máy tính PC.

• $n = 2^8$

i	2	3	4	5
m_i	4	60	152	40
p_i	0,015	0,234	0,594	0,156

• $n = 2^{16}$

i	2	3	4	5	6	7	8
m_i	4	252	4284	23280	31836	5872	8
p_i	0,000061	0,00385	0,07	0,36	0,49	0,09	0,000122

• $n = 2^{24}$

i	2	3	4	5	6
m_i	4	572	25588	470876	3383184
p_i	0,00000024	0,000034	0,0015	0,028	0,202

i	7	8	9	10	11
m_i	7999420	4600716	296744	112	0
p_i	0,477	0,274	0,0177	0,000007	0

• $n = 2^{32}$

i	2	3	4	5	6	7
m_i	4	1020	87828	3497184	65965896	544606968
p_i	$9{,}31 \,.\, 10^{-10}$	$0{,}024 \,.\, 10^{-5}$	0,00002	0,00081	0,0154	0,127

i	8	9	10	11	12	13
m_i	1706622208	1648133316	321571628	4481200	44	0
p_i	0,397	0,384	0,075	0,001	$0{,}1 \,.\, 10^{-7}$	0

Phương pháp này có những hiệu quả nhất định song không tránh khỏi hạn chế về thời gian:

Với dãy có số lượng bít càng lớn thì thời gian thống kê (thực hiện trên máy PC) càng lâu. Cụ thể là:

Số lượng bít	2^8	2^{16}	2^{24}	2^{32}	2^{40}
Thời gian	≈0:00:00,6	≈0:00:00,28	≈0:03:34,22	≈25 h	≈10 tháng

Với trường hợp số lượng bít là 2^{40} thì lượng thời gian để thống kê được là rất lớn. Vì vậy, kết quả thống kê chưa được trình bày ở đây.

3. TEST MỚI

3.1. Năm tiêu chuẩn phổ biến

Cho $s = s_0 s_1 \dots s_{n-1}$ là dãy nhị phân độ dài n. Có 5 tiêu chuẩn phổ biến [2] được dùng để xác định xem dãy s có vi phạm một số đặc trưng đặc biệt của dãy ngẫu nhiên thật sự hay không. Tuy nhiên, việc vượt qua 5 tiêu chuẩn này không khẳng định một cách tất định rằng $s = s_0 s_1 \dots s_{n-1}$ là thật sự ngẫu nhiên mà nó chỉ có ý nghĩa một khẳng định mang tính xác suất.

3.1.1. Kiểm tra tần số lệch

Cho n_0, n_1 là số các số 0, 1 tương ứng trong dãy s. Sử dụng công thức thống kê sau:

$$X_1 = \frac{(n_0 - n_1)^2}{n}$$

xấp xỉ tuân theo phân phối χ^2 với một bậc tự do nếu $n \geq 10$.

Tiêu chuẩn này dùng để kiểm tra xem các số 0 và 1 có xuất hiện đều nhau hay không.

3.1.2. Kiểm tra các bộ đôi móc xích

Cho n_{00}, n_{01}, n_{10}, n_{11} là số lần xuất hiện của các cặp 00, 01, 10, 11 tương ứng trong dãy s. Chú ý rằng $n_{00} + n_{01} + n_{10} + n_{11} = n - 1$ vì ta tính bộ đôi móc xích. Sử dụng công thức thống kê sau:

$$X_2 = \frac{4}{n-1}(n_{00}^2 + n_{01}^2 + n_{10}^2 + n_{11}^2) - \frac{2}{n}(n_0^2 + n_1^2) + 1$$

xấp xỉ tuân theo phân phối χ^2 với 2 bậc tự do nếu $n \geq 21$.

Tiêu chuẩn này dùng để kiểm tra xem các bộ đôi nói trên có xác suất xuất hiện như nhau hay không.

3.1.3. Kiểm tra Poker

Cho m là số nguyên dương thỏa mãn $\lfloor n/m \rfloor \geq 5.2^m$, và cho $k = \lfloor n/m \rfloor$. Chia dãy s thành k phần rời nhau, mỗi phần có độ dài m, và cho n_i là số lần xuất hiện của mẫu thứ i của đoạn dài m, $1 \leq i \leq 2^m$.

Tiêu chuẩn này dùng để kiểm tra xem các đoạn có độ dài m (thường là nhỏ) nói trên có xác suất xuất hiện như nhau hay không. Sử dụng công thức thống kê sau:

$$X_3 = \frac{2^m}{k}\left(\sum_{i=1}^{2^m} n_i^2\right) - k,$$

nó xấp xỉ tuân theo phân phối χ^2 với $2^m - 1$ bậc tự do.

3.1.4. Kiểm tra các loạt

Một khối được hiểu là một dãy liên tiếp các số 1, và một gap là một dãy liên tiếp các số 0. Trong một dãy ngẫu nhiên có độ dài n, số trung bình các gap (hoặc các khối) có độ dài i là $e_i = (n - i + 3)/2^{i+2}$. Cho k là số nguyên lớn nhất trong các số i thỏa mãn $e_i \geq 5$.

Cho B_i, G_i tương ứng là các số khối, gap độ dài i trong s với mỗi i, $1 \leq i \leq k$. Sử dụng công thức thống kê sau:

$$X_4 = \sum_{i=1}^{k} \frac{(B_i - e_i)^2}{e_i} + \sum_{i=1}^{k} \frac{(G_i - e_i)^2}{e_i},$$

nó xấp xỉ tuân theo phân phối χ^2 với $2k - 2$ bậc tự do. Có thể hiểu ở đây đã xấp xỉ hai lần, những số hạng tương ứng với $e_i < 5$ có giá trị rất bé nên đã được bỏ qua.

3.1.5. Kiểm tra tự tương quan

Tiêu chuẩn này kiểm tra tương quan giữa dãy s và các dịch chuyển của nó. Cho d là một số nguyên cố định, $1 \leq d \leq \lfloor n/2 \rfloor$. Kí hiệu

$$A(d) = \sum_{i=0}^{n-d-1} s_i \oplus s_{i+d}\,.$$

Sử dụng công thức thống kê sau:

$$X_5 = 2\Big(A(d) - \frac{n-d}{2}\Big)/\sqrt{n-d},$$

nó xấp xỉ tuân theo phân phối chuẩn $N(0,1)$ nếu $n - d \geq 10$. Vì các giá trị bé cũng như giá trị lớn của $A(d)$ là không mong muốn, nên test 2-phía được sử dụng.

3.2. Test mới dựa theo độ phức tạp Ziv–Lempet

Giả sử ta có dãy với độ dài n. Ta chia dãy này thành những khối có độ dài l (= 8, 16, 32, 64, 128, 256 bít), chẳng hạn sau khi chia ta có k khối: $s_1, s_2, \ldots, s_k$ (trong đó $s_i \cap s_j = \emptyset,\ i \neq j$).

Goi m_i là số các khối độ dài l có độ phức tạp Ziv–Lempet là i.

n_0, n_1 là các chỉ số mà $0 < p_i < 1$, $\forall\, n_0 \leq i \leq n_1$ và $p_i = 0$ với $i < n_0$ hoặc $i > n_1$.

Xác suất xuất hiện: $p[i] = \frac{m[i]}{k}$

$$\chi^2 = \sum_{i=n_0}^{n_1} \frac{m[i]^2}{kp[i]} - k = \sum_{i=n_0}^{n_1} \frac{(m[i] - kp[i])^2}{kp[i]}\,.$$

Chẳng hạn bậc tự do của χ^2 được kí hiệu là t ($t = n_1 - n_0$), $\alpha = 0{,}05$.

Nếu $\chi^2 \geq \chi^2_{i-1}(\alpha)$: bác bỏ.

Ngược lại, nếu $\chi^2 < \chi^2_{i-1}(\alpha)$: chấp nhận.

Theo yêu cầu của phân phối χ^2 cần có điều kiện $k * p[i] > 5$, $\forall i$.

3.3. Kết quả thực nghiệm

Ví dụ 1. Giả sử ta có dãy độ dài $n = 240$ như sau:

11101111 00101010 10000011 00010000 10101011
11101111 00101010 10000011 00010000 10101011
11101111 00101010 10000011 00010000 10101011
11101111 00101010 10000011 00010000 10101011
11101111 00101010 10000011 00010000 10101011
11101111 00101010 10000011 00010000 10101011

• **Sử dụng năm tiêu chuẩn cơ bản:**

∗ Test tần số lệch: $n = 240$, $n_0 = 126$, $n_1 = 114$; $X_1 = 0{,}6$ ($\chi^2 = 3{,}84$) → chấp nhận.

∗ Serial test: $n_{00} = 60$, $n_{01} = 66$, $n_{10} = 66$, $n_{11} = 47$; $X_2 = 3{,}4293$ ($\chi^2 = 5{,}99$) → chấp nhận.

∗ Poker test: $m = 3$, $k = 80$; $X_3 = 9{,}6$ ($\chi^2 = 14{,}1$) → chấp nhận.

∗ Runs test: $k = 5$, $k_l = 5$; $X_4 = 30{,}6808$ ($\chi^2 = 15{,}5$) → bác bỏ.

∗ Test tự tương quan: $d = 3$, $A(d) = 143$; $X_5 = 3{,}1829$ ($\chi^2 = 3{,}84$) → chấp nhận.

• **Sử dụng test mới:**

Khi $l = 8$ thì số khối nhỏ nhất cần phải có là:

$$k > \frac{5}{p[i]} = \frac{5}{(0{,}0156 + 0{,}156)} = 29$$

(trong đó ta gộp xác suất xuất hiện độ phức tạp 2 với 5 được $0{,}0156 + 0{,}156$, chính là xác suất nhỏ nhất so với xác suất xuất hiện độ phức tạp 3 và 4).

Vậy với dãy $n = 240$ thì số khối $k = 30$; i chính là độ phức tạp Ziv–Lempet của khối; m_i chính là số khối xuất hiện có độ phức tạp i. Sử dụng thuật toán Ziv–Lempet ta được kết quả sau:

i	2	3	4	5
m_i	0	24	6	0

Sau đó áp dụng phương pháp thống kê χ^2 ta có kết quả: $X = 53{,}941051$ ($\chi^2 = 7{,}84$) → bác bỏ.

Ví dụ 2. Giả sử ta có độ dài $n = 240$ như sau:

00000000 10101000 10010011 11011000 011111111
00000000 10101000 10010011 11011000 011111111
00000000 10101000 10010011 11011000 011111111
00000000 10101000 10010011 11011000 011111111
00000000 10101000 10010011 11011000 011111111
00000000 10101000 10010011 11011000 011111111

• **Sử dụng 5 tiêu chuẩn cơ bản:**

∗ Test tần số lệch: $n = 240$, $n_0 = 132$, $n_1 = 108$; $X_1 = 2{,}4$ ($\chi^2 = 3{,}84$) → chấp nhận.

∗ Serial test: $n_{00} = 78$, $n_{01} = 54$, $n_{10} = 53$, $n_{11} = 54$; $X_2 = 5{,}0435$ ($\chi^2 = 5{,}99$) → chấp nhận.

∗ Poker test: $m = 3$, $k = 80$; $X_3 = 8{,}00$ ($\chi^2 = 14{,}1$) → chấp nhận.

∗ Runs test: $k = 8$, $k_l = 7$; $X_4 = 11{,}4595$ ($\chi^2 = 23{,}7$) → chấp nhận.

∗ Test tự tương quan: $d = 3$, $A(d) = 105$; $X_5 = -1{,}7538$ ($\chi^2 = 3{,}84$) → chấp nhận.

• **Sử dụng test mới:**

Với dãy $n = 240$ thì khối $k = 30$; i chính là độ phức tạp Ziv–Lempet của khối; m_i chính là số khối xuất hiện có độ phức tạp i. Sử dụng thuật toán Ziv–Lempet ta được kết quả sau:

i	2	3	4	5
m_i	6	6	12	6

Sau đó áp dụng phương pháp thống kê χ^2 ta có kết quả: $X = 11{,}131483$ ($\chi^2 = 7{,}84$) $\rightarrow$ bác bỏ.

Nhìn vào ví dụ trên ta có nhận xét dãy đưa vào thử là một dãy tồi (có nhiều chu kỳ lặp lại). Nhưng với năm tiêu chuẩn cơ bản thì dãy trên được chấp nhận. Trong khi đó với test mới thì dãy đó đã bị bác bỏ.

Chú ý:

$*$ Khi $l = 16$ thì số khối nhỏ nhất cần phải có là:

$$k > \frac{5}{p[i]} = \frac{5}{(0{,}00385 + 0{,}07)} = 67.$$

$*$ Khi $l = 24$ thì số khối nhỏ nhất cần phải có là:

$$k > \frac{5}{p[i]} = \frac{5}{(0{,}000034 + 0{,}0015 + 0{,}028 + 0{,}0177)} = 105.$$

$*$ Khi $l = 32$ thì số khối nhỏ nhất cần phải có là:

$$k > \frac{5}{p[i]} = \frac{5}{(24 * 10^{-7} + 2 * 10^{-5} + 0{,}00081 + 0{,}154)} = 290.$$

Vậy, dựa trên kết quả thực nghiệm tính toán được ta thấy test mới có hiệu quả trong việc kiểm tra tính ngẫu nhiên của dãy nhị phân.

4. KẾT LUẬN

Để kiểm tra tính ngẫu nhiên của dãy 0-1 có rất nhiều phương pháp được sử dụng. Ở đây chúng tôi giới thiệu một test mới trên cơ sở kết hợp phương pháp thống kê χ^2 và độ phức tạp Ziv–Lempet. Các ví dụ nêu ra cho thấy test mới có tác dụng bổ sung cho 5 tiêu chuẩn cơ bản trong việc đánh giá độ ngẫu nhiên của dãy nhị phân.

Tôi xin chân thành cảm ơn TS Nguyễn Ngọc Cương đã giúp đỡ và cho nhiều đóng góp về nội dung bài báo này. Tôi xin trân trọng cảm ơn TS Đoàn Văn Ban đã đọc và đóng góp ý kiến để bài báo được hoàn thiện.

TÀI LIỆU THAM KHẢO

[1] Abraham Lempet, On the complexity of finite sequences, *IEEE Transactions on Information Theory*, Vol. IT-22, No. 1 (1976).
[2] Benezes A., Van Oorschot P. C., Vanstole S., *Handbook of Applied Cryptografy*, CRC press, 1977.
[3] John M. Carroll and Sri Nurdiati, Weak keys and weak data foiling the two nemeses, *Cryptologia*, Volume XVIII, Number 3 (1994).
[4] Mund S, *Ziv–Lempet Complexity for Periodic Sequences and its Cryptographic Application*, Eurocrypt'91.

Nhận bài ngày 10 - 11 - 2001
Nhận lại sau khi sửa ngày 8 - 1 - 2002

Số 162, ngõ 205, đường Giải Phóng,
Quận Hai Bà Trưng, Hà Nội.