

TCVN

TIÊU CHUẨN QUỐC GIA

TCVN 9600: 2013

ISO 24153:2009

Xuất bản lần 1

LẤY MẪU NGẪU NHIÊN VÀ QUY TRÌNH NGẪU NHIÊN HÓA

Random sampling and randomization procedures

HÀ NỘI - 2013

Mục lục

	Trang
Lời nói đầu.....	5
Lời giới thiệu.....	6
1 Phạm vi áp dụng.....	7
2 Tài liệu viện dẫn.....	7
3 Thuật ngữ, định nghĩa và ký hiệu	8
3.1 Thuật ngữ và định nghĩa	8
3.2 Ký hiệu.....	11
4 Quy định chung	12
5 Lấy mẫu ngẫu nhiên – Phương pháp dùng thiết bị cơ.....	13
5.1 Phương pháp bình.....	13
5.2 Phương pháp đồng xu hay xúc sắc	13
6 Lấy mẫu ngẫu nhiên giả độc lập – Phương pháp bảng số.....	15
6.1 Bảng số ngẫu nhiên.....	15
6.2 Phương pháp cơ bản.....	15
7 Lấy mẫu ngẫu nhiên giả độc lập – Phương pháp máy tính.....	15
7.1 Tổng quan	15
7.2 Thuật toán khởi tạo.....	16
7.3 Thuật toán tạo số ngẫu nhiên	17
7.4 Hồ sơ đánh giá	19
8 Áp dụng cho tình huống lấy mẫu phổ biến.....	19
8.1 Khái quát	19
8.2 Số nguyên ngẫu nhiên trong dải	19
8.3 Hoán vị ngẫu nhiên	20
8.4 Trộn ngẫu nhiên.....	20
8.5 Lấy mẫu ngẫu nhiên có hoàn lại	20
8.6 Lấy mẫu ngẫu nhiên không hoàn lại	21
8.7 Lấy mẫu ngẫu nhiên cho phương án lấy mẫu liên tục (CSP).....	21
8.8 Lấy mẫu ngẫu nhiên phân tầng.....	22
8.9 Lấy mẫu ngẫu nhiên một lần từ cỡ lô ban đầu chưa biết.....	22
8.10 Lấy mẫu ngẫu nhiên một lần theo thứ tự, không hoàn lại	22
8.11 Lấy mẫu chùm	24
8.12 Lấy mẫu ngẫu nhiên với xác suất tỷ lệ thuận với cỡ mẫu	24

TCVN 9600:2013

8.13 Lấy mẫu nhiều tầng.....	24
8.14 Ngẫu nhiên hóa trong thiết kế thực nghiệm.....	25
8.15 Hình vuông La Tinh ngẫu nhiên	26
Phụ lục A (quy định), Bảng số ngẫu nhiên	27
Phụ lục A (tham khảo), Mã máy tính thuật toán tạo số ngẫu nhiên	32
Phụ lục A (tham khảo), Lấy mẫu ngẫu nhiên và mã máy tính ngẫu nhiên hóa	35
Thư mục tài liệu tham khảo	41

Lời nói đầu

TCVN 9600:2013 hoàn toàn tương đương với ISO 24153:2009;

TCVN 9600:2013 do Ban kỹ thuật tiêu chuẩn quốc gia TCVN/TC 69
Ứng dụng các phương pháp thống kê biên soạn, Tổng cục Tiêu chuẩn
Đo lường Chất lượng đề nghị, Bộ Khoa học và Công nghệ công bố.

Lời giới thiệu

Lấy mẫu ngẫu nhiên và quy trình ngẫu nhiên hóa là cơ sở để xác nhận giá trị của nhiều phương pháp thống kê sử dụng trong thử nghiệm, dù là cho mục đích kiểm soát chất lượng công nghiệp và cải tiến hay cho thiết kế thực nghiệm trong lĩnh vực y tế, sinh học, nông nghiệp hay lĩnh vực cụ thể khác. Nhiều tiêu chuẩn thống kê đề cập đến việc tiến hành thử nghiệm như vậy. Đặc biệt, tất cả các tiêu chuẩn lấy mẫu chấp nhận dưới đây đều được thiết kế trên tiền đề là lấy mẫu ngẫu nhiên được sử dụng để lựa chọn đơn vị lấy mẫu cần thiết ứng với bố trí lô:

TCVN 7790 (ISO 2859) (tất cả các phần), Quy trình lấy mẫu để kiểm tra định tính

TCVN 8243 (ISO 3951) (tất cả các phần), Quy trình lấy mẫu để kiểm tra định lượng

TCVN 9601 (ISO 8422), Phương án lấy mẫu liên tiếp để kiểm tra định tính

ISO 8423, Phương án lấy mẫu liên tiếp để kiểm tra định lượng phần trăm không phù hợp (đã biết độ lệch chuẩn)

ISO 13448 (tất cả các phần), Quy trình lấy mẫu chấp nhận dựa trên phân bố nguyên tắc ưu tiên (APP)

ISO 14560, Quy trình lấy mẫu chấp nhận định tính – Mức chất lượng qui định về số cá thể không phù hợp trên một triệu

ISO 18414, Quy trình lấy mẫu chấp nhận định tính – Hệ thống lấy mẫu chấp nhận không dựa trên nguyên tắc tin cậy đối với kiểm soát chất lượng đầu ra

ISO 21247, Hệ thống lấy mẫu chấp nhận không kết hợp và qui trình kiểm soát quá trình để chấp nhận sản phẩm

Ngoài ra, TCVN 7790-3 (ISO 2859-3) và ISO 21247 bao gồm các qui định đối với lấy mẫu ngẫu nhiên được áp dụng để xác định xem lô có cần kiểm tra hoặc không áp dụng qui trình lấy mẫu lô cách quãng hay không, đồng thời để quyết định đơn vị nào cần kiểm tra từ một quá trình sản xuất trong phương án lấy mẫu liên tục, một cách tương ứng. Do đó, điều quan trọng là hiệu lực của tất cả các tiêu chuẩn trên để áp dụng lấy mẫu ngẫu nhiên hiệu quả.

Mặc dù các nguyên tắc của tiêu chuẩn này có thể áp dụng chung trong trường hợp yêu cầu lấy mẫu ngẫu nhiên và có thể xác định rõ đơn vị lấy mẫu, ưu tiên trên cơ sở cá thể riêng rẽ, nhưng có nhiều tình huống trong đó vật liệu quan tâm không được xác định là cá thể riêng rẽ, như trong trường hợp vật liệu dạng đồng. Trong tình huống này, người sử dụng cần tham khảo tiêu chuẩn ISO sau đây để có hướng dẫn thích hợp:

ISO 11648 (tất cả các phần), Khía cạnh thống kê của lấy mẫu từ vật liệu dạng đồng

Lấy mẫu ngẫu nhiên và quy trình ngẫu nhiên hóa

Random sampling and randomization procedures

1 Phạm vi áp dụng

Tiêu chuẩn này quy định các quy trình lấy mẫu ngẫu nhiên và ngẫu nhiên hóa. Nhiều phương pháp được đưa ra, bao gồm cả các phương pháp dựa trên thiết bị cơ khí, bảng số ngẫu nhiên và thuật toán trên máy tính xách tay.

Tiêu chuẩn này áp dụng trong trường hợp quy định, hợp đồng hay tiêu chuẩn khác yêu cầu sử dụng lấy mẫu ngẫu nhiên hay ngẫu nhiên hóa. Các phương pháp áp dụng cho các trường hợp như

- a) lấy mẫu chấp nhận các đơn vị riêng rẽ để kiểm tra theo lô,
- b) lấy mẫu cho mục đích khảo sát,
- c) đánh giá kết quả của hệ thống quản lý chất lượng, và
- d) chọn đơn vị thử nghiệm, phân bổ cách xử lý chúng và xác định thứ tự đánh giá trong tiến hành các thiết kế thực nghiệm.

Ngoài ra, tiêu chuẩn còn có các thông tin để hỗ trợ việc đánh giá hay xem xét bên ngoài khác các kết quả lấy mẫu ngẫu nhiên hay ngẫu nhiên hóa theo yêu cầu của nhân sự quản lý chất lượng hay cơ quan chế định.

Tiêu chuẩn này không cung cấp hướng dẫn như quy trình lấy mẫu ngẫu nhiên hay ngẫu nhiên hóa thích hợp để sử dụng cho tình huống thực nghiệm cụ thể bất kỳ hoặc đưa ra hướng dẫn về việc lựa chọn chiến lược lấy mẫu hay xác định cỡ mẫu. Cần tham khảo các tiêu chuẩn khác (như liệt kê trong phần Lời giới thiệu) hoặc các tài liệu viện dẫn hợp lệ hướng dẫn về vấn đề này.

2 Tài liệu viện dẫn

Tài liệu viện dẫn dưới đây rất cần thiết cho việc áp dụng tiêu chuẩn này. Đối với các tài liệu ghi năm công bố thì áp dụng bản được nêu. Đối với các tài liệu không ghi năm công bố thì áp dụng bản mới nhất, bao gồm cả các sửa đổi.

TCVN 9600:2013

TCVN 7870-2 (ISO 80000-2), Đại lượng và đơn vị – Phần 2: Dấu và ký hiệu toán học sử dụng trong khoa học tự nhiên và công nghệ

TCVN 8244-1 (ISO 3534-1), Thống kê học – Từ vựng và ký hiệu – Phần 1: Thuật ngữ chung về thống kê và thuật ngữ dùng trong xác suất

TCVN 8244-2 (ISO 3534-2), Thống kê học – Từ vựng và ký hiệu – Phần 2: Thống kê ứng dụng

ISO 3534-3, *Statistics – Vocabulary and symbols – Part 3: Design of experiments* (Thống kê học – Từ vựng và ký hiệu – Phần 3: Thiết kế thực nghiệm)

3 Thuật ngữ, định nghĩa và ký hiệu

Tiêu chuẩn này áp dụng các thuật ngữ và định nghĩa trong TCVN 8244-1 (ISO 3534-1), TCVN 8244-2 (ISO 3534-2), ISO 3534-3 và các thuật ngữ, định nghĩa dưới đây.

3.1 Thuật ngữ và định nghĩa

3.1.1

Chùm (cluster)

Bộ phận của **tổng thể** (3.1.6) chia tách thành các nhóm **đơn vị mẫu** (3.1.13) tách biệt nhau theo một cách thức nhất định.

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.2.28]

3.1.2

Lấy mẫu chùm (cluster sampling)

Phép **lấy mẫu** (3.1.12) trong đó **mẫu ngẫu nhiên** (3.1.8) của các **chùm** (3.1.1) được chọn và tất cả các **đơn vị mẫu** (3.1.13) tạo thành chùm bao gồm trong **mẫu** (3.1.11).

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.3.9]

3.1.3

Xáo trộn (derangement)

Hoán vị hoàn toàn (complete permutation)

Hoán vị các thành phần trong đó không một thành phần nào còn giữ nguyên vị trí ban đầu trong tập hợp (ví dụ {3, 1, 2} là hoán vị của {1, 2, 3}).

3.1.4

Lô (lot)

Bộ phận xác định của **tổng thể** (3.1.6) cấu thành trong các điều kiện về cơ bản giống với tổng thể xét về mục đích **lấy mẫu** (3.1.12).

CHÚ THÍCH: Ví dụ, mục đích lấy mẫu có thể để xác định khả năng chấp nhận lô hoặc để ước lượng giá trị trung bình của một đặc trưng cụ thể.

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.2.4]

3.1.5

Lấy mẫu nhiều tầng (multistage sampling)

Phép lấy mẫu (3.1.12) trong đó mẫu (3.1.11) được chọn theo tầng, các đơn vị mẫu (3.1.13) tại mỗi tầng được lấy từ đơn vị mẫu lớn hơn được chọn ở tầng trước đó.

CHÚ THÍCH: Lấy mẫu nhiều tầng khác với lấy mẫu nhiều lần. Lấy mẫu nhiều lần là lấy mẫu theo nhiều tiêu chí đồng thời.

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.3.10]

3.1.6

Tổng thể (population)

<tham chiếu> toàn bộ các cá thể được xem xét.

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.2.1]

3.1.7

Lấy mẫu ngẫu nhiên giả độc lập (pseudo-independent random sampling)

Lấy mẫu (3.1.12) trong đó mẫu (3.1.11) gồm n đơn vị mẫu (3.1.13) được lấy từ tổng thể (3.1.6) theo bảng số ngẫu nhiên hoặc thuật toán trên máy tính được thiết kế sao cho mỗi tổ hợp có thể có của n đơn vị mẫu có một xác suất được lấy cụ thể (xem thêm 4.4).

3.1.8

Mẫu ngẫu nhiên (random sample)

Mẫu (3.1.11) chọn bằng phép lấy mẫu ngẫu nhiên (3.1.9).

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.2.25]

3.1.9

Lấy mẫu ngẫu nhiên (random sampling)

Phép lấy mẫu (3.1.12) trong đó một mẫu (3.1.11) gồm n đơn vị mẫu (3.1.13) được lấy từ một tổng thể (3.1.6) sao cho mỗi tổ hợp có thể có của n đơn vị mẫu có một xác suất được lấy cụ thể.

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.3.5]

3.1.10

Ngẫu nhiên hóa (randomization)

Quá trình nhờ đó tập hợp các cá thể được đặt theo một trật tự ngẫu nhiên.

CHÚ THÍCH: Nếu từ một tổng thể (3.1.6) gồm các số tự nhiên từ 1 đến n , các số được lấy ngẫu nhiên (nghĩa là theo cách thức tất cả các số đều có cơ hội được lấy như nhau), từng số một, lần lượt, không hoàn lại, cho đến khi

TCVN 9600:2013

hết tổng thể, các số này được gọi là lấy "theo thứ tự ngẫu nhiên".

Nếu n số này được liên kết trước với n đơn vị riêng biệt hay n xử lý riêng biệt mà sau đó được sắp xếp lại theo thứ tự các số được lấy thì thứ tự của các đơn vị hay xử lý đó được gọi là được ngẫu nhiên hóa.

3.1.11

Mẫu (sample)

Tập hợp con của **tổng thể** (3.1.6) gồm một hoặc nhiều **đơn vị mẫu** (3.1.13).

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.2.17]

3.1.12

Lấy mẫu (sampling)

Hoạt động lấy hoặc thành lập một **mẫu** (3.1.11).

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.3.1]

3.1.13

Đơn vị mẫu (sampling unit)

Đơn vị (unit)

Một trong các phần riêng lẻ hợp thành **tổng thể** (3.1.6).

CHÚ THÍCH 1: Đơn vị mẫu có thể gồm một hoặc nhiều cá thể, ví dụ một bao diêm, nhưng sẽ bao hàm một kết quả thử.

CHÚ THÍCH 2: Đơn vị mẫu có thể gồm các cá thể rời rạc hoặc một lượng vật liệu dạng đồng xác định.

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.2.14]

3.1.14

Lấy mẫu có hoàn lại (sampling with replacement)

Lấy mẫu (3.1.12) trong đó mỗi **đơn vị mẫu** (3.1.13) được lấy ra, quan trắc, rồi được trả về **tổng thể** (3.1.6) trước khi lấy đơn vị mẫu tiếp theo.

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.3.15]

3.1.15

Lấy mẫu không hoàn lại (sampling without replacement)

Lấy mẫu (3.1.12) trong đó mỗi **đơn vị mẫu** (3.1.13) chỉ được lấy ra từ **tổng thể** (3.1.6) một lần mà không được trả về tổng thể đó.

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.3.16]

3.1.16

Giá trị xuất phát (seed)

Trị số hoặc tập hợp các giá trị để khởi đầu một thuật toán **lấy mẫu ngẫu nhiên giả độc lập** (3.1.7) hoặc để thiết lập điểm bắt đầu trong bảng số ngẫu nhiên.

3.1.17

Mẫu ngẫu nhiên đơn giản (simple random sample)

Mẫu (3.1.11) được chọn bằng cách lấy mẫu ngẫu nhiên đơn giản (3.1.18).

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.2.24]

3.1.18

Lấy mẫu ngẫu nhiên đơn giản (simple random sampling)

Phép lấy mẫu (3.1.12) trong đó một mẫu (3.1.11) gồm n đơn vị mẫu (3.1.13) được lấy từ một tổng thể (3.1.6) sao cho tất cả các tổ hợp có thể có của n đơn vị mẫu có cùng xác suất được lấy ra.

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.3.4]

3.1.19

Lấy mẫu phân lớp (stratified sampling)

Lấy mẫu (3.1.12) sao cho các phần mẫu (3.1.11) được lấy từ lớp (3.1.21) khác nhau và mỗi lớp được lấy ít nhất là một đơn vị mẫu (3.1.13).

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.3.6]

3.1.20

Lấy mẫu ngẫu nhiên đơn giản phân lớp (stratified simple random sampling)

Lấy mẫu ngẫu nhiên đơn giản (3.1.18) từ mỗi lớp (3.1.21).

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.3.7]

3.1.21

Lớp (stratum)

Tổng thể con tách biệt và đầy đủ được xem là đồng nhất hơn về các đặc trưng nghiên cứu so với toàn bộ tổng thể (3.1.6).

[TCVN 8244-2:2010 (ISO 3534-2:2006), 1.2.29]

3.2 Ký hiệu

Tiêu chuẩn này áp dụng dấu và ký hiệu toán học trong TCVN 7870-2 (ISO 80000-2) và các ký hiệu sau:

d_i	con số thứ i hoặc giá trị mật của đồng xu hay xúc sắc
N	cỡ lô
n	cỡ mẫu
n_i	cỡ của mẫu thứ i
U	biến thực ngẫu nhiên phân bố đều trên phạm vi mở (0, 1)
x_i	giá trị thứ i của biến x
$j!$	j giai thừa

- [z] hàm trần nguyên của z (trả về số nguyên nhỏ nhất lớn hơn hoặc bằng giá trị thực z)
 [z] hàm sàn nguyên của z (trả về phần nguyên của giá trị thực z)

4 Quy định chung

4.1 Lấy mẫu ngẫu nhiên là điều kiện tiên quyết để áp dụng đúng hầu hết các phương án lấy mẫu trong ứng dụng công nghiệp. Tương tự, ngẫu nhiên hóa, sử dụng các nguyên tắc lấy mẫu ngẫu nhiên, là không thể thiếu được khi thực hiện các thiết kế thực nghiệm, vì nó làm tăng hiệu lực nội tại của thực nghiệm, cho phép sử dụng các phương pháp thống kê trong giải thích các kết quả thực nghiệm. Mục tiêu của lấy mẫu ngẫu nhiên là cung cấp phương tiện áp dụng các kết quả của lý thuyết xác suất vào các vấn đề thực tiễn, trong khi tránh được mọi dạng độ chệch bất kỳ. Mục tiêu này không đạt được khi sử dụng các loại lấy mẫu nhất định khác. Ví dụ, lấy mẫu dựa trên các khái niệm như trực giác hay đánh giá của cá nhân, tính lộn xộn, hoặc theo hạn ngạch đều có độ chệch vốn có và kết quả dẫn đến sai lỗi nghiêm trọng trong quá trình ra quyết định mà không có điều kiện đánh giá rủi ro. Lấy mẫu ngẫu nhiên đồng xác suất nhằm loại trừ độ chệch như vậy bằng cách đảm bảo rằng mỗi đơn vị trong lô có cùng một xác suất được chọn (lấy mẫu có hoàn lại) hay, nói cách khác, mọi mẫu có thể trong cỡ mẫu cho trước lấy từ một lô có cùng xác suất được chọn (lấy mẫu không hoàn lại).

4.2 Trong phương án lấy mẫu ngẫu nhiên đồng xác suất có hoàn lại, xác suất một đơn vị cụ thể trong lô gồm N đơn vị được chọn ở lần lấy cho trước bất kỳ luôn là $1/N$. Có N^n mẫu ngẫu nhiên theo thứ tự có thể có của n đơn vị từ N đơn vị và, đầy đủ là, có $(N + n - 1)! / [n! (N - 1)!]$ mẫu ngẫu nhiên không theo thứ tự khác nhau có thể có gồm n đơn vị từ N đơn vị (xem chú thích bên dưới).

Theo phương án lấy mẫu ngẫu nhiên đơn giản không hoàn lại, xác suất một đơn vị trong lô được chọn ở lần lấy cho trước là $1/N$ đối với lần lấy đầu tiên, $1/(N - 1)$ đối với lần lấy thứ hai, $1/(N - 2)$ đối với lần lấy thứ ba, và cứ tiếp tục như vậy. Nếu n đơn vị được chọn ngẫu nhiên từ lô gồm N đơn vị mà không hoàn lại, khi đó mỗi tổ hợp n đơn vị có cùng một xác suất lựa chọn với mọi tổ hợp N đơn vị khác lấy n đơn vị mỗi lần. Số lượng mẫu gồm n đơn vị ngẫu nhiên không theo thứ tự xác suất khác nhau từ lô gồm N đơn vị là $N! / (N - n)!$, tương đương với số lượng tổ hợp gồm N đơn vị lấy n đơn vị mỗi lần. Điều đáng lưu ý khác là số lượng mẫu ngẫu nhiên theo thứ tự gồm n đơn vị được lấy không hoàn lại từ lô gồm N đơn vị là $N! / (N - n)!$ tương đương với số hoán vị có thể có gồm N đơn vị lấy n đơn vị mỗi lần. Cần chú ý là việc lấy mẫu ngẫu nhiên không hoàn lại là phương án lấy mẫu phổ biến nhất được sử dụng trong các ứng dụng lấy mẫu chấp nhận.

CHÚ THÍCH: Trong lấy mẫu có hoàn lại trên cơ sở mẫu gồm 3 đơn vị trong 5 đơn vị, danh mục {1, 1, 2}, {1, 2, 1} và {2, 1, 1} sẽ khác khi xét về thứ tự (và về kỹ thuật gọi là đa tập hợp) nhưng giống nhau nếu không xét đến thứ tự.

4.3 Mục tiêu của lấy mẫu ngẫu nhiên chỉ có thể đạt được bằng việc gắn với các quy trình chặt chẽ được thiết kế kỹ càng để đạt được mục đích của định nghĩa. Nhiều phương pháp được trình bày trong

tiêu chuẩn này để thực thi mục tiêu này. Cụ thể là các phương pháp dùng thiết bị cơ khí giả định rằng đồng xu và xúc sắc không có độ chệch, được thiết kế sao cho mỗi mặt có cùng một xác suất xuất hiện trong quá trình ném hay tung và cách thức tung hay ném được thực hiện để không gây ra độ chệch. Hơn nữa, do những khác biệt lớn trong việc áp dụng nội tại các phương pháp lấy mẫu ngẫu nhiên trong máy tính cũng như hệ điều hành máy tính, ngôn ngữ lập trình và phần mềm (xem tài liệu tham khảo [9], [10], [12] và [13] về thông tin thêm), nên tiêu chuẩn này lựa chọn phương pháp tạo mẫu ngẫu nhiên bằng máy tính cầm tay. Ngoài ra, cần lưu ý là tất cả các phương pháp dưới đây đòi hỏi rằng mỗi đơn vị riêng rẽ trong lô được kết hợp trước với một số nhất định từ 1 đến N , sao cho các đơn vị lấy mẫu xác định là kết quả của phương pháp lấy mẫu ngẫu nhiên có thể thu được một cách rõ ràng từ lô đó.

4.4 Cuối cùng, để giảm khó khăn trong việc trình bày, tính từ "giả độc lập" sẽ thường được bỏ bớt khi đề cập đến quy trình hay phương pháp lấy mẫu ngẫu nhiên như vậy (xem tài liệu tham khảo [8]). Hơn nữa, tính từ "ngẫu nhiên" sẽ được sử dụng thường xuyên với nghĩa là danh từ mà nó bỏ nghĩa (thường là số hoặc phép hoán vị) là đầu ra của quá trình tạo ra ngẫu nhiên số hay phép hoán vị đó. Ngoài ra, khi có các ví dụ, cỡ mẫu liên quan được giữ ở con số nhỏ với mục tiêu minh họa một cách đơn giản những khái niệm liên quan.

5 Lấy mẫu ngẫu nhiên – Phương pháp dùng thiết bị cơ khí

5.1 Phương pháp bình

5.1.1 Đặt N vật giống hệt nhau nhưng được đánh số khác nhau (ví dụ: vé, đồng tiền hay quả bóng) vào một chiếc bình để thể hiện rõ ràng mỗi trong số N đơn vị trong lô đó rồi trộn kỹ các đồ vật đó.

5.1.2 Đối với lấy mẫu không hoàn lại, nhắm mắt chọn các đồ vật trong bình, từng vật một mà không trả lại vào bình rồi trộn lại các vật giữa các lần lấy liên tiếp, cho đến khi có được n đơn vị lấy mẫu mong muốn.

CHÚ THÍCH: Phương pháp này thường được sử dụng bởi các công ty xổ số.

5.1.3 Đối với lấy mẫu có hoàn lại, nhắm mắt chọn các đồ vật trong bình, từng vật một rồi trả lại vào bình sau mỗi lần lấy và trộn kỹ lại các vật giữa các lần lấy liên tiếp, cho đến khi có được n đơn vị lấy mẫu mong muốn. Sử dụng phương pháp này, cùng một đơn vị có thể xuất hiện trong mẫu nhiều hơn một lần.

5.2 Phương pháp đồng xu hay xúc sắc

5.2.1 Xác định số m đồng xu hay xúc sắc (hoặc lượt thả đồng xu hay tung xúc sắc) yêu cầu, trong đó N là cỡ lô và k là số mặt của vật được sử dụng, theo công thức sau:

$$m = \lceil \log_2 N / \log_2 k \rceil$$

5.2.2 Khi sử dụng nhiều đồng xu hoặc xúc sắc, gắn kết rõ từng đồng xu hay xúc sắc với một vị trí cụ thể trong thứ tự giải thích các số d_i . Trường hợp sử dụng một đồng xu hay một xúc sắc, ấn định kết quả của lần tung hay thả đầu tiên với số có nghĩa nhất d_m , lần tung hay thả thứ hai với số có nghĩa tiếp theo d_{m-1} , và tiếp tục như vậy.

5.2.3 Thả đồng xu hay tung xúc sắc rồi ghi lại m kết quả theo thứ tự d_i . Chuyển kết quả sang số nguyên thập phân theo công thức sau:

$$y = 1 + \sum_{i=1}^m (d_i - 1)k^{m-i}$$

5.2.4 Lặp lại bước 5.2.3, loại tất cả các giá trị vượt quá N và, trong trường hợp lấy mẫu không hoàn lại, tất cả các giá trị đã được chọn, cho đến khi đạt được n đơn vị mẫu mong muốn.

VÍ DỤ 1: Kiểm tra viên mong muốn có được mẫu ngẫu nhiên gồm 4 đơn vị từ lô có 20 đơn vị và có sẵn một đồng xu. Từ bước 5.2.1, xác định được rằng cần $m = 5$ lần tung đồng xu để có được mỗi số ngẫu nhiên. Ấn định trước rằng giá trị mặt sấp là 1 và giá trị mặt ngửa là 2. Thứ tự những lần tung đầu tiên có được đa tập hợp {1, 2, 1, 2, 2}, thông qua bước 5.2.3 bằng $1 + (0)2^4 + (1)2^3 + (0)2^2 + (1)2^1 + (1)2^0 = 12$. Ba lượt tung tiếp theo cho các tập hợp {1, 2, 2, 2, 1}, {1, 1, 2, 2, 1} và {2, 2, 1, 2, 2} tương ứng bằng 15, 7 và 28. Vì giá trị 28 vượt quá cỡ lô nên cần loại bỏ và các lượt tung tiếp theo cần được thực hiện cho đến khi thu được một số hợp lệ nữa để hoàn thành việc lấy mẫu ngẫu nhiên.

VÍ DỤ 2: Cần mẫu ngẫu nhiên gồm 4 đơn vị từ một lô gồm 50 đơn vị và kiểm tra viên có nhiều xúc sắc sáu mặt có màu sắc khác nhau. Từ bước 5.2.1, xác định được cần $m = 3$ con xúc sắc để thu được từng số ngẫu nhiên. Kiểm tra viên chọn xúc sắc màu xanh da trời, xanh lá cây và đỏ rồi xếp chúng theo cùng thứ tự từ ý nghĩa nhất đến ít ý nghĩa nhất. Tuy nhiên, kiểm tra công thức ở 5.2.3 cho thấy bằng chứng là các số nằm trong phạm vi hợp lệ từ 1 đến 50 sẽ chỉ có được khi mặt xúc sắc đầu tiên là 1 hoặc 2. Kết quả là, hiệu quả nhất định có thể đạt được bằng cách đặt các giá trị mặt cao hơn của xúc sắc xanh da trời là 1 hoặc 2 mà không làm méo xác suất kết quả. Kiểm tra viên quyết định trước đối với giá trị mặt lẻ trên xúc sắc xanh da trời coi là 1 và giá trị mặt chẵn coi là 2. Vòng đầu tiên cho tập hợp {3, 3, 4}, thông qua bước 5.2.3 bằng $1 + (2)6^2 + (2)6^1 + (3)6^0 = 88$, giá trị này quá lớn nhưng khi chuyển thành {1, 3, 4} bằng 16. Ba vòng tiếp theo cho {6, 1, 3} (chuyển thành {2, 1, 3}), {5, 6, 6} (chuyển thành {1, 6, 6}) và {2, 5, 5}, tương ứng bằng 39, 36 và 65. Vì giá trị 65 vượt quá cỡ lô nên cần loại bỏ và cần thực hiện những lần tung bổ sung cho đến khi thu được một giá trị hợp lệ nữa để hoàn thành lấy mẫu ngẫu nhiên.

VÍ DỤ 3: Tinh huống tương tự như trong Ví dụ 2 nhưng lần này kiểm tra viên thấy rằng ba xúc sắc sẽ tạo ra các số trong phạm vi từ 1 đến $6^3 = 216$ mà cỡ lô chỉ là 50. Kiểm tra viên quyết định trước sẽ ánh xạ tất cả các kết quả từ 1 đến 200 lên dải từ 1 đến 50 và loại bỏ mọi kết quả lớn hơn 200 nhằm tránh bóp méo xác suất đầu ra. Bốn lượt tung tương tự như ở ví dụ trước được đánh giá theo phương án ánh xạ này. Các đa tập hợp {3, 3, 4}, {6, 1, 3}, {5, 6, 6} và {2, 5, 5} bằng 88, 183, 180 và 65. Các số này được trừ đi cho bội số của 50 cho đến khi chúng nằm trong phạm vi dải từ 1 đến 50 (nếu thu được giá trị 0 thì giải thích như N), có được các giá trị mẫu tương ứng là 38, 33, 30 và 15. Cỡ mẫu là 4 đơn vị đã thu được nên không cần tiếp tục tung xúc sắc nữa. Chú ý là, về mặt toán học, quá trình ánh xạ này tương đương với việc áp dụng công thức $v_2 = 1 + (v_1 - 1)$ mô đun N , trong đó v_1 là giá trị ban đầu còn v_2 là giá trị được ánh xạ vào dải mong muốn.

6 Lấy mẫu ngẫu nhiên giả độc lập – Phương pháp bảng số

6.1 Bảng số ngẫu nhiên

Hai bảng số ngẫu nhiên được cung cấp trong Phụ lục A. Mỗi bảng số gồm 3 600 số ngẫu nhiên từ 0 đến 9, sắp xếp thành 60 hàng và 60 cột. Việc sử dụng được mô tả tóm tắt dưới đây và nêu chi tiết hơn trong Phụ lục A.

CHÚ THÍCH: Các con số trong bảng tương tự trực tiếp với các giá trị mặt của xúc sắc 10 mặt tung lặp lại và ghi được. Số chữ số m cần thiết cho ứng dụng lấy mẫu tương ứng với số lần tung xúc sắc.

6.2 Phương pháp cơ bản

6.2.1 Xác định số chữ số m cần thiết để thể hiện cỡ lô N . Trường hợp cỡ lô có lũy thừa 10, bỏ qua con số đầu trong cỡ lô đó và giải thích rằng các con số "không" còn lại bằng giá trị cỡ lô (ví dụ nếu $N = 1\ 000$, giải thích giá trị 000 là 1 000).

6.2.2 Chọn ngẫu nhiên một điểm bắt đầu (nghĩa là giá trị hàng và cột) trong bảng sử dụng phương pháp mô tả ở A.2.2.

6.2.3 Đọc con số kết quả cùng với các số $m - 1$ ở bên phải như một số và ghi lại giá trị đó. Trường hợp các con số bên phải vượt quá cột thứ 60 thì coi các cột 1, 2, ... tương ứng là cột 61, 62, ...

6.2.4 Tăng giá trị của hàng thêm một, lặp lại bước 6.2.3 rồi ghi lại giá trị đó. Trường hợp giá trị của hàng vượt quá hàng thứ 60 thì coi hàng 1 là hàng thứ 61 và tăng giá trị cột lên m chữ số.

6.2.5 Lặp lại bước 6.2.4, bỏ qua tất cả các giá trị vượt quá N và, trong trường hợp lấy mẫu không hoàn lại, tất cả các giá trị đã được chọn, cho đến khi thu được số đơn vị lấy mẫu n mong muốn.

VÍ DỤ: Người đánh giá mong muốn chọn một mẫu ngẫu nhiên gồm 5 đơn vị từ một lô 200 đơn vị. Điểm bắt đầu ngẫu nhiên được xác định bằng các lần tung đồng xu là hàng 57 và cột 59, quyết định là sử dụng Bảng A.1. Vì N nhỏ so với giá trị lớn nhất có thể thể hiện bằng 3 chữ số (nghĩa là 1 000) nên người đánh giá quyết định ánh xạ kết quả của dải từ 1 đến 1 000 lên dải từ 1 đến 200. Kết quả là năm số sau: 848, 670, 902, 034 và 518. Giá trị mẫu chuyển đổi trở thành 48, 70, 102, 34 và 118.

7 Lấy mẫu ngẫu nhiên giả độc lập – Phương pháp máy tính

7.1 Tổng quan

7.1.1 Tiêu chuẩn này sử dụng hệ thống thuật toán cụ thể đề cập trong các tài liệu tham khảo [1], [7] và [13]. Các thuật toán này được thiết kế có các tính chất toán học và thống kê cần thiết cho lấy mẫu ngẫu nhiên cũng như có thể linh hoạt áp dụng trong các ngôn ngữ lập trình khác nhau trên các nền tảng máy tính khác nhau đồng thời tạo thuận lợi cho việc kiểm tra và đánh giá các giá trị mẫu được chọn cần thiết cho mục đích quy định. Ví dụ thực thi các phần chương trình chính được cho trong Phụ lục B sử dụng ngôn ngữ lập trình C.

TCVN 9600:2013

7.1.2 Hệ thống các thuật toán liên quan gồm hai hệ con chính:

- a) thuật toán khởi tạo tùy chọn tự động tạo ra số nguyên ban đầu giả ngẫu nhiên dựa trên thời gian trôi qua kể từ ngày tháng chuẩn; và
- b) bộ tạo số ngẫu nhiên.

7.1.3 Đối với mục đích kiểm tra hay đánh giá, thuật toán khởi tạo tùy chọn đề cập ở điểm a) của 7.1.2 và mô tả ở 7.2 có thể bỏ qua với một giá trị xuất phát nhập bằng tay. Giá trị này cần nằm trong dãy số nguyên từ 1 đến và bao gồm cả 2 147 483 398. Bản sao giá trị đầu vào này được giữ lại để lưu hồ sơ khi cần. Tuy nhiên, trong sử dụng chung cho các ứng dụng kiểm soát chất lượng và thiết kế thực nghiệm không nên thường xuyên bỏ qua tùy chọn tạo giá trị xuất phát ngẫu nhiên tự động mà nên là tùy chọn mặc định trong thực tế.

CHÚ THÍCH: Trình bày về các bước của các thuật toán trong điều này được để ở định dạng toán học hơn nhằm hỗ trợ cho việc lập trình. Mã lập trình có các tham chiếu điều được nêu trong Phụ lục B để hỗ trợ cho việc áp dụng điều này.

7.2 Thuật toán khởi tạo

7.2.1 Thuật toán khởi tạo bao gồm:

- a) thuật toán tính thời gian trôi qua, tham chiếu đến ngày tháng và thời gian quá khứ cố định; và
- b) thuật toán tạo số ngẫu nhiên dựa trên phân bố đều, gọi là số ngẫu nhiên của thời gian dựa trên đầu ra của mục a) ở trên, để thu được giá trị xuất phát ngẫu nhiên dựa trên đầu vào theo thời gian.

7.2.2 Thuật toán dưới đây xác định số giây trôi qua tính từ 00:00:00 01-01-2000 đến ngày tháng và thời gian hiện tại.

- a) Tìm ngày tháng và thời gian của hệ thống máy tính theo biến dạng chuỗi, lưu bản sao biến này vào hồ sơ và sau đó phân tích chuỗi thành các thành phần thời gian (nghĩa là giây, phút, giờ, ngày, tháng và năm).
- b) Tính số ngày tròn trôi qua d_0 từ điểm thời gian chuẩn, sử dụng trị số năm đầy đủ bốn con số y , tháng m_1 và ngày d của ngày tháng hiện tại xử lý như sau:

nếu $m_1 < 3$ thì lấy $m_1 = m_1 + 12$ và lấy $y = y - 1$

$$d_0 = d + \lfloor (153 m_1 - 457) / 5 \rfloor + 365 y + \lfloor y / 4 \rfloor - \lfloor y / 100 \rfloor + \lfloor y / 400 \rfloor - 730 426$$

CHÚ THÍCH: Công thức cho d_0 có thể đơn giản hóa một chút đối với năm theo lịch đến và kể cả 2099 bằng cách thay số hạng $\lfloor y / 4 \rfloor$ bằng " $- 730 441$ ".

- c) Tính tổng số giây s_0 trôi qua từ ngày tháng quy chiếu sử dụng con số thu được ở bước b) và thời gian của ngày (ở dạng 24 giờ "giờ:phút:giây") tìm trước trong biến chuỗi ở bước a) theo công thức sau:

$$s_e = 86\,400 d_e + 3\,600 h + 60 m_2 + s$$

trong đó h , m_2 và s tương ứng là giờ, phút và giây.

CHÚ THÍCH 1: Một số ngôn ngữ lập trình có hàm dựng sẵn để thực hiện tính toán trực tiếp s_e . Các hàm nội tại như vậy cần được xác nhận hiệu lực trước khi sử dụng để đảm bảo ảnh hưởng của năm nhuận và thời gian ghi ban ngày được xử lý thích hợp.

CHÚ THÍCH 2: Trong ứng dụng 32 bit của thuật toán này, giá trị s_e sẽ tăng theo thời gian đến điểm gây tràn tính toán. Cần chú ý khi lập trình để đảm bảo giá trị đầu vào luôn được ánh xạ lên dãy từ 1 đến và bao gồm cả 2 147 483 398.

d) Giá trị tính ở bước c) là giá trị khởi tạo cho bộ tạo giá trị ngẫu nhiên và được dùng để thu được giá trị cuối cùng. Bản sao giá trị này được ghi vào biến riêng để lưu hồ sơ khi cần.

e) Số thời gian j mà bộ tạo số ngẫu nhiên sau đó được yêu cầu là một số nguyên ngẫu nhiên từ 1 đến và bao gồm cả 100, dựa trên hai con số ít có nghĩa nhất của giá trị thu được ở bước c) cộng thêm 1, được thể hiện như sau:

$$j = s_e - 100 \lfloor s_e / 100 \rfloor + 1$$

7.2.3 Bộ tạo số ngẫu nhiên dùng cho thuật toán tạo giá trị xuất phát tự động (hàm khởi tạo) có dạng quan hệ truy hồi đồng dạng tuyến tính:

$$a) x_{j+1} = 40\,692 x_j \bmod 2\,147\,483\,399,$$

có thể thực hiện trên máy tính có khả năng xử lý số nguyên 32 bit qua các bước sau đây:

$$b) k = \lfloor x_j / 52\,774 \rfloor;$$

$$c) x_{j+1} = 40\,692 (x_j - 52\,774 k) - 3\,791 k;$$

$$d) \text{ Nếu } x_{j+1} < 0 \text{ thì lấy } x_{j+1} = x_{j+1} + 2\,147\,483\,399.$$

7.2.4 Tạo giá trị xuất phát cho thuật toán lấy mẫu ngẫu nhiên bằng cách ấn định kết quả ở điểm c) 7.2.2 cho x_j rồi gọi công thức ở 7.2.3 j lần cho mỗi bước 7.2.2 e), mỗi lần thay x_j bằng x_{j+1} cho đến khi thực hiện hết số lần gọi yêu cầu.

7.2.5 Giá trị cuối cùng của x_{j+1} thu được từ bước 7.2.4 là số nguyên ngẫu nhiên từ 1 đến và bao gồm cả 2 147 483 398 và dùng làm giá trị xuất phát ban đầu cho thuật toán lấy mẫu ngẫu nhiên mô tả ở 7.3 [cụ thể là giá trị y_i ở bước b) của 7.3.6]. Bản sao giá trị này được ghi vào biến riêng để lưu hồ sơ khi cần.

7.3 Thuật toán tạo số ngẫu nhiên

7.3.1 Thuật toán tạo số ngẫu nhiên bao gồm

a) một dãy xáo trộn được tạo bởi thuật toán tạo số ngẫu nhiên phân bố đều và,

TCVN 9600:2013

b) tổ hợp, thuật toán tạo số ngẫu nhiên phân bố đều.

7.3.2 Tạo dãy A gồm 32 phần tử dùng như phương tiện xáo trộn đầu ra của thuật toán lấy mẫu ngẫu nhiên.

7.3.3 Bộ tạo số ngẫu nhiên dưới đây được dùng để tạo dãy xáo trộn:

a) $x_{i+1} = 40\,014 x_i \bmod 2\,147\,483\,563$,

có thể thực hiện trên máy tính 32 bit qua các bước sau đây:

b) $k = \lfloor x_i / 53\,668 \rfloor$;

c) $x_{i+1} = 40\,014 (x_i - 53\,668 k) - 12\,211 k$;

d) Nếu $x_{i+1} < 0$ thì lấy $x_{i+1} = x_{i+1} + 2\,147\,483\,563$.

7.3.4 Khởi tạo dãy A bằng cách ấn định kết quả ở 7.1.3 hoặc 7.2.5 cho x_i rồi gọi bộ tạo giá trị nêu trong 7.3.3 a) 40 lần, mỗi lần gọi thay x_i bằng x_{i+1} , loại bỏ 8 giá trị đầu tiên và sau đó ấn định mỗi trong số 32 giá trị đầu ra còn lại của x_{i+1} cho dãy đó theo thứ tự đảo (nghĩa là từ phần tử 32 xuống đến phần tử 1).

7.3.5 Đặt phần tử 1 của dãy A (nghĩa là $A[1]$) là giá trị khởi tạo k cho tổ hợp thuật toán tạo số ngẫu nhiên.

7.3.6 Tổ hợp bộ tạo số ngẫu nhiên dùng để tạo mẫu ngẫu nhiên có dạng tổ hợp quan hệ tái diễn đồng dạng tuyến tính và các bước xác định chỉ số dãy sau đây:

a) $x_{i+1} = 40\,014 x_i \bmod 2\,147\,483\,563$,

b) $y_{i+1} = 40\,692 y_i \bmod 2\,147\,483\,399$,

c) $J = \lfloor 32 k / 2\,147\,483\,563 \rfloor + 1$;

d) $k = A[J] - y_{i+1}$;

e) $A[J] = x_{i+1}$;

f) Nếu $k < 1$ thì lấy $k = k + 2\,147\,483\,562$.

CHÚ THÍCH: Hai bộ tạo số ngẫu nhiên ở trên được mô tả trong 7.2.3 và 7.3.3 (tham khảo các điều con nếu cần các ứng dụng tương đương 32 bit).

7.3.7 Thuật toán ở 7.3.6 được bắt đầu bằng việc đặt x_i theo giá trị cuối cùng của x_{i+1} từ 7.3.4 và đặt y_i theo giá trị tham chiếu ở 7.2.5. Các giá trị x_{i+1} và y_{i+1} dùng làm giá trị tiếp theo của x_i và y_i cho tất cả các lần gọi sau của thuật toán. Chỉ số ngẫu nhiên J cho dãy xáo trộn A được tính bằng cách sử dụng giá trị của k (ban đầu từ 7.3.5) và hiệu của $A[J]$ và y_{i+1} được ấn định cho k , trong khi $A[J]$ được cập nhật với x_{i+1} . Cuối cùng giá trị của k được thay đổi nếu cần để tạo ra giá trị dương.

7.3.8 Đầu ra của thuật toán lấy mẫu ngẫu nhiên là giá trị k , là số ngẫu nhiên từ 1 đến và bao gồm cả 2 147 483 562, được chỉnh tỷ lệ như biến thực phân bố đều chuẩn hóa U trong dải từ 0 đến 1, không bao gồm giá trị đầu mút của dải này, như sau: $U = k / 2\,147\,483\,563$.

7.3.9 Đầu ra của 7.3.8 có thể được chỉnh tỷ lệ là biến nguyên phân bố đều L trong dải từ 1 đến N , bao gồm cả N , như sau: $L = \lfloor NU \rfloor + 1$.

7.3.10 Để tạo ra mẫu ngẫu nhiên, lặp lại các bước từ 7.3.6 đến 7.3.9 cho đến khi thu được số giá trị ngẫu nhiên cần thiết.

7.4 Hồ sơ đánh giá

Khi có yêu cầu duy trì hồ sơ cho việc đánh giá của cơ quan chịu trách nhiệm hay cơ quan có thẩm quyền, thì ghi lại cỡ lô và cỡ mẫu.

Ngoài ra, đối với các thuật toán, ghi lại giá trị xuất phát nhập bằng tay theo 7.1.3 hoặc nếu sử dụng bộ tạo giá trị xuất phát ngẫu nhiên thì ghi lại

- ngày và thời gian của hệ thống máy tính sử dụng để tính giá trị xuất phát ban đầu,
- giá trị của số khởi tạo theo 7.2.2 d) và
- giá trị của số khởi tạo cuối cùng theo 7.2.5.

8 Áp dụng cho tình huống lấy mẫu phổ biến

8.1 Khái quát

8.1.1 Điều này đưa ra thuật toán dùng cho nhiều phương án lấy mẫu ngẫu nhiên phù hợp với nhiều tình huống thực tiễn khác nhau.

8.1.2 Xuyên suốt điều này, U được xác định là biến ngẫu nhiên thực, phân bố đều trong dải từ 0 đến 1, không bao gồm các giá trị đầu mút của dải, như cung cấp bởi thuật toán ở 7.3. Nếu sử dụng nguồn khác cho U và đầu ra được biết bao gồm 1 nhưng không bao gồm 0 là giá trị đầu mút của dải thì đặt U bằng $1 - U$. Nếu nguồn khác của U bao gồm 0 và 1 là giá trị đầu mút của dải thì giá trị 1 cần được chặn và loại ra.

8.2 Số nguyên ngẫu nhiên trong dải

Số nguyên ngẫu nhiên K trong dải từ M đến và gồm cả N có thể được tạo ra theo thuật toán dưới đây.

- Tạo giá trị thực ngẫu nhiên U .
- Đặt K bằng $M + \lfloor U(N - M + 1) \rfloor$.

8.3 Hoán vị ngẫu nhiên

Đối với dãy A với N phần tử riêng biệt, phép hoán vị ngẫu nhiên N đơn vị lấy n mỗi lần có thể được tạo ra theo thuật toán xáo trộn dưới đây.

- Án định N giá trị chỉ số phần tử riêng rẽ theo thứ tự ban đầu đến $A[1: N]$.
- Đặt J bằng 1.
- Tạo số nguyên ngẫu nhiên K trong dải từ J đến và bao gồm cả N .
- Đổi $A[J]$ và $A[K]$.
- Tăng J thêm 1.
- Nếu J nhỏ hơn hoặc bằng n thì đi tiếp bước c).
- Thu được hoán vị ngẫu nhiên từ các giá trị n đầu tiên của dãy A .

8.4 Trộn ngẫu nhiên

Đối với dãy A với N phần tử riêng biệt, trộn ngẫu nhiên N đơn vị có thể được tạo ra theo thuật toán dưới đây.

- Án định N giá trị chỉ số phần tử riêng rẽ theo thứ tự ban đầu đến $A[1: N]$ và sao chép vào dãy $B[1: N]$.
- Sử dụng dãy B tạo xáo trộn ngẫu nhiên N đơn vị lấy N (nghĩa là tất cả) đồng thời, sử dụng phương pháp nêu ở 8.3.
- So sánh sự ngang bằng các phần tử từ 1 đến N của dãy A và dãy B
- Nếu phần tử bất kỳ của dãy B bằng với phần tử đối ứng ở dãy A thì dừng việc so sánh và đi tiếp bước b).
- Thu được phép trộn ngẫu nhiên từ dãy B .

CHÚ THÍCH: Thuật toán này có thể được thực hiện hiệu quả hơn ở bước b) và c) bằng cách so sánh phần tử $A[J]$ với phần tử $B[J]$ ngay khi xác định được $B[J]$ chứ không đợi đến khi hoán vị hết toàn bộ dãy B .

8.5 Lấy mẫu ngẫu nhiên có hoàn lại

Mẫu ngẫu nhiên đơn gồm n đơn vị từ lô N đơn vị có thể được tạo ra có hoàn lại theo thuật toán dưới đây.

- Tạo số nguyên ngẫu nhiên K trong dải từ 1 đến và bao gồm cả N .
- Lặp lại bước a) cho đến khi thu được n giá trị của K .

CHÚ THÍCH: Có thể áp dụng lặp lại phương pháp này để thu được số lượng mẫu bất kỳ với cỡ mẫu bất kỳ. Nếu giá trị thu được của mẫu đơn không được lựa chọn thì có thể sử dụng mẫu đó cho kiểm tra lấy mẫu liên tiếp.

8.6 Lấy mẫu ngẫu nhiên không hoàn lại

Mẫu ngẫu nhiên đơn gồm n đơn vị riêng rẽ từ lô N đơn vị có thể được tạo ra không hoàn lại bằng một trong hai phương pháp dưới đây.

a) Phương pháp 1

- 1) Tạo số nguyên ngẫu nhiên K trong dải từ 1 đến và bao gồm cả N .
- 2) Xác nhận rằng giá trị K chưa được tạo ra trước đó; nếu nó khác biệt thì lưu lại giá trị đó còn nếu không thì loại ra.
- 3) Lặp lại bước 1) và 2) cho đến khi thu được n giá trị của K .

b) Phương pháp 2

- 1) Tạo hoán vị ngẫu nhiên N đơn vị, mỗi lần lấy n theo 8.3.
- 2) Sử dụng các giá trị n đầu tiên trong đầu ra dãy A làm mẫu ngẫu nhiên.

CHÚ THÍCH: Có thể sử dụng một trong hai phương pháp này để thu được số mẫu với cỡ mẫu bất kỳ (cho các mục đích như lấy mẫu hai lần hay lấy mẫu nhiều lần) bằng cách sử dụng tổng n_1 của cỡ mẫu riêng n_1 như giá trị đầu vào của n cho thuật toán này, để nguyên các giá trị ở thứ tự đầu ra ban đầu, sau đó lấy các giá trị kết quả n_1 đầu tiên làm mẫu đầu tiên, các giá trị kết quả n_2 tiếp theo làm mẫu thứ hai và cứ tiếp tục như vậy. Hơn nữa, nếu giá trị kết quả của một mẫu đơn không được chọn thì mẫu đó có thể được sử dụng cho kiểm tra lấy mẫu liên tiếp bằng cách kiểm tra từng đơn vị theo thứ tự được chọn.

8.7 Lấy mẫu ngẫu nhiên cho phương án lấy mẫu liên tục (CSP)

Phương án lấy mẫu liên tục CSP-1 được thiết kế để áp dụng cho việc kiểm tra chất lượng dây chuyền sản xuất và luân phiên giữa các giai đoạn kiểm tra 100 % yêu cầu i đơn vị được chấp nhận liên tiếp trước khi tiến hành các giai đoạn tiếp theo của kiểm tra lấy mẫu ở xác suất f , với sự trở lại kiểm tra 100 % khi tìm thấy đơn vị không chấp nhận được. Trong các giai đoạn kiểm tra lấy mẫu, các đơn vị từ dây chuyền sản xuất có thể được chọn cho kiểm tra theo một trong hai phương pháp dưới đây.

a) Phương pháp 1

- 1) Đối với mỗi đơn vị sản xuất, tạo giá trị thực ngẫu nhiên U .
- 2) Nếu U nhỏ hơn hoặc bằng f thì chọn đơn vị đó cho kiểm tra lấy mẫu.
- 3) Lặp lại bước 1) và 2) cho đến khi thu được đơn vị không chấp nhận được.

b) Phương pháp 2

- 1) Đối với mỗi phân đoạn sản xuất n đơn vị, trong đó n bằng $1/f$, tạo số nguyên ngẫu nhiên K trong dải từ 1 đến và bao gồm cả n .
- 2) Chọn đơn vị tương ứng với K làm đơn vị lấy mẫu để kiểm tra.
- 3) Lặp lại bước 1) và 2) cho đến khi thu được đơn vị không chấp nhận được.

TCVN 9600:2013

CHÚ THÍCH: Đối với phương án CSP-1, giá trị f được quy định là nghịch đảo của số nguyên.

8.8 Lấy mẫu ngẫu nhiên phân tầng

Đối với lô gồm hai hay nhiều tầng cỡ N_i , chọn mẫu ngẫu nhiên đơn cỡ n_i từ mỗi tầng i sử dụng các phương pháp nêu trong 8.3 hoặc 8.6 khi lấy mẫu không hoàn lại được yêu cầu hoặc phương pháp nêu trong 8.5 khi yêu cầu lấy mẫu có hoàn lại.

8.9 Lấy mẫu ngẫu nhiên một lần từ cỡ lô ban đầu chưa biết

Mẫu ngẫu nhiên đơn gồm n đơn vị khác nhau từ một lô ban đầu chưa biết cỡ lô, nhưng ít nhất là bằng cỡ n , có thể thu được theo phương pháp dưới đây (lấy từ tài liệu tham khảo [11]).

- Án định các đơn vị n đầu tiên từ lô cho dãy mẫu $A[1:n]$.
- Nếu có đơn vị khác trong lô liệt kê thì đặt N bằng số đếm đơn vị tiếp theo; nếu không thì thực hiện bước f).
- Tạo số nguyên ngẫu nhiên K trong dải từ 1 đến và bao gồm cả N .
- Nếu K nhỏ hơn hoặc bằng n thì đặt $A[K]$ bằng N .
- Thực hiện bước b).
- Thu được mẫu ngẫu nhiên từ dãy A và cỡ lô từ giá trị N .

CHÚ THÍCH: Phương pháp này cũng có thể sử dụng khi đã biết cỡ lô.

8.10 Lấy mẫu ngẫu nhiên một lần theo thứ tự, không hoàn lại

Mẫu ngẫu nhiên đơn gồm n đơn vị riêng biệt từ lô N đơn vị có thể được tạo ra trực tiếp theo thứ tự tăng dần theo các phương pháp dưới đây.

- Phương pháp 1 (lấy từ tài liệu tham khảo [2])
 - Khởi tạo các biến sau:
 - tạo dãy $A[1:n]$;
 - đặt L bằng N , K bằng $N - n$ và J bằng 0.
 - Tăng J thêm 1.
 - Nếu J lớn hơn n thì đi tiếp bước 8).
 - Tạo giá trị thực ngẫu nhiên U và đặt P bằng 1.
 - Đặt P bằng $P \cdot K / N$.
 - Nếu P nhỏ hơn hoặc bằng U :
 - đặt $A[J]$ bằng $N - L + 1$ sau đó tăng L thêm 1;

ii) thực hiện bước 2).

7) Nếu P lớn hơn U :

i) tăng L thêm 1 và K thêm 1;

ii) thực hiện bước 5).

8) Thu được mẫu ngẫu nhiên theo thứ tự tăng dần từ dãy A .

b) Phương pháp 2 (lấy từ tài liệu tham khảo [3])

1) Lấy $C(a, b)$ là hàm số thu được số tổ hợp các đơn vị a lấy b mỗi lần (còn gọi là hệ số nhị thức và bằng $a! / [(a - b)! b!]$).

2) Tạo số nguyên ngẫu nhiên L trong dải từ 1 đến và bao gồm cả $C(N, n)$.

3) Tạo dãy $A[1:n]$.

4) Đặt K bằng 0, J bằng 1 và m bằng $n - 1$.

5) Đặt $A[J]$ bằng 0.

6) Nếu J không bằng 1 thì đặt $A[J]$ bằng $A[J - 1]$.

7) Đặt $A[J]$ bằng $A[J] + 1$.

8) Đặt R bằng $C(N - A[J], n - J)$.

9) Tăng K thêm R .

10) Nếu K nhỏ hơn L thì thực hiện bước 7).

11) Giảm K đi R .

12) Tăng J thêm 1.

13) Nếu J nhỏ hơn hoặc bằng N thì thực hiện bước 5).

14) Đặt $A[n]$ bằng $A[m] + L - K$.

15) Thu được mẫu ngẫu nhiên theo thứ tự tăng dần từ dãy A

CHÚ THÍCH: Do những hạn chế trong trình bày lượng số nguyên lớn trên máy tính cũng như những giới hạn trong độ phân giải của bộ tạo số ngẫu nhiên, cần chú ý để đảm bảo rằng Phương pháp 2 có thể thực hiện được bằng máy tính và không có độ chệch quá mức do bộ tạo số ngẫu nhiên được sử dụng.

VÍ DỤ: Mẫu ngẫu nhiên gồm 5 đơn vị sắp theo thứ tự được yêu cầu lấy từ lô gồm 25 đơn vị. Có thể tạo $25! / (20! 5!) = 53\ 130$ tổ hợp gồm 5 đơn vị từ 25 đơn vị, thực hiện được trên máy tính hiện đại. Ngoài ra, bộ tạo số ngẫu nhiên mô tả ở Điều 7 được chọn cho mục đích lấy mẫu này; giá trị đầu ra tối đa của nó là 2 147 483 562 lớn hơn trên 40 419 lần so với dải yêu cầu nên độ chệch tạo ra trong phương pháp này là không đáng kể đối với mục đích thực tiễn. Số nguyên ngẫu nhiên đơn từ 1 đến và bao gồm cả 53 130 được tạo ra là 7 319. Tập hợp mẫu thu được là {1, 7, 13, 18, 19}.

8.11 Lấy mẫu chùm

Đối với tổng thể hoặc lô gồm chùm các đơn vị liên quan, ghi các chùm vào một danh mục và chọn mẫu ngẫu nhiên từ danh mục này sử dụng các phương pháp nêu trong 8.3 hoặc 8.6 khi lấy mẫu không hoàn lại được yêu cầu hoặc phương pháp nêu trong 8.5 khi yêu cầu lấy mẫu có hoàn lại. Mẫu thu được bao gồm tổng số đơn vị trong các chùm được chọn.

8.12 Lấy mẫu ngẫu nhiên với xác suất tỷ lệ thuận với cỡ mẫu

Đối với tổng thể gồm các đơn vị có cỡ tích hợp khác nhau, có thể thu được mẫu ngẫu nhiên các đơn vị được chọn tỷ lệ với cỡ mẫu bằng cách sử dụng một trong hai phương pháp dưới đây.

a) Phương pháp 1

- 1) Trong danh mục N đơn vị có cỡ khác nhau, ghi lại cỡ mẫu cộng dồn S_i của các đơn vị bên cạnh từng đơn vị liên tiếp.
- 2) Tạo số nguyên ngẫu nhiên K trong dải từ 1 đến và kể cả S_N , trong đó S_N là cỡ cộng dồn của toàn bộ tổng thể.
- 3) Từ danh mục, chọn đơn vị kèm theo cỡ cộng dồn lớn nhất không vượt quá K làm thành phần mẫu.
- 4) Lặp lại bước 2) và 3) cho đến khi thu được số n đơn vị lấy mẫu (có hoàn lại hoặc không hoàn lại) mong muốn.

b) Phương pháp 2

- 1) Từ danh mục N đơn vị có cỡ khác nhau, xác định cỡ đơn vị lớn nhất M .
- 2) Tạo cặp số nguyên ngẫu nhiên (K, L) với K trong dải từ 1 đến và kể cả N và L trong dải từ 1 đến và kể cả M .
- 3) Nếu cỡ đơn vị K không vượt quá M thì chọn đơn vị K làm thành phần mẫu.
- 4) Lặp lại bước 2) và 3) cho đến khi thu được số n đơn vị lấy mẫu (có hoàn lại hoặc không hoàn lại) mong muốn.

VÍ DỤ: Một công ty marketing muốn tiến hành một cuộc khảo sát các hộ gia đình với lựa chọn tỷ lệ với cỡ hộ gia đình (nghĩa là số nhân khẩu). Thu được một danh sách 10 hộ gia đình xếp theo thứ tự số lượng nhân khẩu (cỡ mẫu) và các cỡ là {2, 2, 3, 3, 3, 4, 4, 5, 6, 7}. Cỡ cộng dồn của danh sách này là: {2, 4, 7, 10, 13, 17, 21, 26, 32, 39}. Mẫu ngẫu nhiên không hoàn lại gồm 4 hộ gia đình được yêu cầu. Số nguyên ngẫu nhiên từ 1 đến và kể cả 39 được tạo ra, thu được {7, 33, 2, 11}. Đơn vị lấy mẫu tương ứng là các hộ gia đình với thứ hạng liệt kê: {3, 10, 1, 5}.

8.13 Lấy mẫu nhiều tầng

Đối với tổng thể hoặc lô có cấu trúc lồng nhóm thứ bậc nhỏ dần, chọn mẫu ngẫu nhiên từ các nhóm lớn nhất, sau đó các nhóm con nhỏ hơn từ mỗi nhóm được chọn trước đó, tiếp tục quy trình này cho

đến khi đạt đến từng cấp đơn vị trong sơ đồ. Ở mỗi tầng, sử dụng phương pháp lấy mẫu ngẫu nhiên nêu trong 8.3 hoặc 8.6 khi lấy mẫu không hoàn lại được yêu cầu hoặc phương pháp nêu trong 8.5 khi yêu cầu lấy mẫu có hoàn lại. Số đơn vị trong mẫu là tích của số mẫu được lấy ở mỗi tầng.

VÍ DỤ: Một lô gồm 20 palet, mỗi palet có 20 thùng. Mỗi thùng có 10 đơn vị. Người mua muốn kiểm tra sản phẩm bằng phương án lấy mẫu nhiều tầng. Một mẫu ngẫu nhiên gồm 4 palet được chọn. Sau đó, từ mỗi palet đã chọn, chọn ra mẫu ngẫu nhiên gồm 4 thùng. Cuối cùng, từ mỗi thùng được chọn, chọn ra mẫu ngẫu nhiên gồm 3 đơn vị. Quy trình này thu được một mẫu gồm 48 đơn vị từ 4 000 đơn vị trong lô.

8.14 Ngẫu nhiên hóa trong thiết kế thực nghiệm

Trong các ứng dụng thiết kế thực nghiệm, ngẫu nhiên hóa được sử dụng để thực hiện những hoạt động như phân bổ các xử lý thực nghiệm cho các đơn vị hoặc đối tượng và thiết lập thứ tự đánh giá các đơn vị, bao gồm cả thứ tự đánh giá đối với các thiết kế lặp. Có thể sử dụng một trong hai phương pháp ngẫu nhiên hóa dưới đây.

a) Phương pháp 1

- 1) Ấn định số nguyên từ 1 đến N cho từng phần tử trong danh mục N xử lý hoặc đơn vị, tùy trường hợp.
- 2) Tạo hoán vị ngẫu nhiên N số nguyên được lấy (nghĩa là tất cả) tại một thời điểm.
- 3) Tiến hành hoạt động thực nghiệm theo trình tự thu được từ bước 2).

b) Phương pháp 2

- 1) Tạo N biến thực ngẫu nhiên U_i và ấn định từng giá trị theo thứ tự tạo ra cho từng phần tử liên tiếp trong danh mục các xử lý hoặc đơn vị, tùy trường hợp.
- 2) Sắp xếp danh mục xử lý hoặc đơn vị theo thứ tự tăng dần theo giá trị tương ứng U của chúng.
- 3) Tiến hành hoạt động thực nghiệm theo trình tự thu được từ bước 2).

VÍ DỤ 1: Nhà nghiên cứu y học muốn thử nghiệm tác dụng của một loại thuốc mới so với điều trị sử dụng thông thường trong điều kiện y tế đặc biệt. Mười hai đối tượng tình nguyện và được đánh số từ 1 đến 12 khi tham gia vào thử nghiệm lâm sàng này. Nhà nghiên cứu hoạch định phân bổ điều trị A (thuốc mới) cho 6 đối tượng và điều trị B (thuốc hiện thời) cho sáu đối tượng còn lại. Như một công đoạn làm giảm độ chệch, trước hết thử nghiệm viên quyết định ngẫu nhiên hóa thứ tự 12 điều trị trước khi phân bổ chúng vào danh mục ngẫu nhiên hóa 12 đối tượng. Từng số chỉ số các điều trị và đối tượng được ngẫu nhiên hóa riêng theo phương pháp 1. Danh mục ngẫu nhiên hóa thu được của các điều trị là {B, B, A, B, A, A, B, A, A, B, B, A} và danh mục ngẫu nhiên hóa đối tượng là {3, 7, 12, 5, 1, 9, 11, 4, 10, 2, 8, 6}. Lúc này các điều trị có thể phân bổ trực tiếp cho các đối tượng dựa trên thứ tự trong các danh mục, thu được {B3, B7, A12, B5, A1, A9, B11, A4, A10, B2, B8, A6}. Kết quả là điều trị A được phân bổ cho các đối tượng {1, 4, 6, 9, 10, 12} còn điều trị B được phân bổ cho các đối tượng {2, 3, 5, 7, 8, 11}.

VÍ DỤ 2: Người thử nghiệm muốn tiến hành thực nghiệm lặp lại, thử nghiệm từng đơn vị trong số 5 đơn vị theo thứ tự ngẫu nhiên tại ba thời điểm khác nhau. Tập hợp {1, 2, 3, 4, 5} được hoán vị ngẫu nhiên theo phương pháp 1, thu được ba danh mục theo thứ tự như sau: {2, 1, 5, 4, 3}, {1, 5, 2, 3, 4} và {4, 3, 5, 2, 1}. Ba danh mục này

TCVN 9600:2013

được ấn định theo thứ tự lần thử tương ứng là 1, 2 và 3 và thử nghiệm viên thử nghiệm các đơn vị theo thứ tự quy định trong danh mục kèm theo từng lần thử.

8.15 Hình vuông La Tinh ngẫu nhiên

Hình vuông La Tinh ngẫu nhiên bậc n là dãy $n \times n$ chứa các ký hiệu từ bảng chữ cái cỡ n , sắp xếp sao cho mỗi ký hiệu xuất hiện đúng một lần trong mỗi hàng và đúng một lần trong mỗi cột. Sẽ hữu ích trong việc hoạch định một số loại thiết kế thực nghiệm. Hình vuông La Tinh ngẫu nhiên bậc n có thể tạo ra bằng phương pháp dưới đây (từ tài liệu tham khảo [4]).

- a) Tạo dãy $A[1:n, 1:n]$ và $C[1:n]$.
- b) Đặt R bằng 1.
- c) Ấn định các số nguyên từ 1 đến n cho dãy $C[1:n]$
- d) Đặt J bằng N .
- e) Đặt C bằng 1.
- f) Đặt I bằng 0.
- g) Tạo số nguyên ngẫu nhiên X trong dải từ 1 đến n và bao gồm cả J .
- h) Đặt H bằng 1.
- i) Nếu I lớn hơn 50, thực hiện bước c).
- j) Nếu $A[H, C]$ bằng $C[X]$, tăng I thêm 1 và thực hiện bước g).
- k) Tăng H thêm 1.
- l) Nếu H nhỏ hơn hoặc bằng $R - 1$, thực hiện bước i).
- m) Đặt $A[R, C]$ bằng $C[X]$, giảm J đi 1
- n) Nếu X lớn hơn J , thực hiện bước r).
- o) Đặt K bằng X .
- p) Đặt $C[K]$ bằng $C[K + 1]$.
- q) Tăng K thêm 1; nếu K nhỏ hơn hoặc bằng J , thực hiện bước p).
- r) Tăng C thêm 1; nếu C nhỏ hơn hoặc bằng n , thực hiện bước f).
- s) Tăng R thêm 1; nếu R nhỏ hơn hoặc bằng n , thực hiện bước c).
- t) Thu được hình vuông La Tinh ngẫu nhiên từ dãy A .

CHÚ THÍCH: Thuật toán tạo các hình vuông La Tinh ngẫu nhiên phân bố đều có thể thấy trong tài liệu tham khảo [5]. Ngoài ra, cần chú ý rằng có sự kết nối giữa hình vuông La Tinh ngẫu nhiên và việc tạo ra xáo trộn ngẫu nhiên một hoán vị vì mỗi hàng và cột hình vuông La Tinh là một xáo trộn của tất cả các hàng và các cột trước đó.

Phụ lục A

(quy định)

Bảng số ngẫu nhiên

A.1 Mô tả

Phụ lục này đưa ra hai bảng số ngẫu nhiên áp dụng khi không có sẵn máy tính để thực hiện các thuật toán tạo số ngẫu nhiên của tiêu chuẩn này. Mỗi bảng gồm 3 600 số ngẫu nhiên từ 0 đến 9, mỗi số xuất hiện với tần suất bằng nhau. Các bảng đều được sắp xếp dưới dạng 60 hàng và 60 cột để thuận tiện cho việc sử dụng thời gian trong ngày làm việc khởi đầu. Các bảng được tạo ra bằng cách sử dụng các thuật toán mô tả trong Điều 7.

A.2 Sử dụng

A.2.1 Số chữ số và giải thích

A.2.1.1 Xác định số chữ số m cần thiết để thể hiện cỡ lô N . Số chữ số bằng số chữ số trong cỡ lô trừ trường hợp cỡ lô là lũy thừa của 10; trong trường hợp này, bỏ qua số đầu trong cỡ lô và giải thích các số không còn lại là số chữ số yêu cầu cũng như bằng với giá trị cỡ lô (ví dụ nếu $N = 100$, giải thích giá trị 00 là 100).

A.2.1.2 Khi cỡ lô nhỏ hơn hoặc bằng một nửa của 10^m thì các số nhập vào bảng có thể được giải thích dựa trên ánh xạ của giá trị quan trắc lên dải từ 1 đến N , với điều kiện là độ lệch không được đưa vào quá trình này. Điều này có thể được thực hiện bằng cách bỏ qua tất cả các giá trị vượt quá kN , trong đó $k = \lfloor 10^m / N \rfloor$, trước khi ánh xạ giá trị theo công thức $v_2 = 1 + (v_1 - 1) \cdot m / N$, trong đó v_1 là giá trị ban đầu và v_2 là giá trị ánh xạ vào dải mong muốn.

A.2.2 Điểm bắt đầu

A.2.2.1 Trước khi có thể thu được các số trong bảng, cần quyết định về kế hoạch chọn điểm bắt đầu. Các bảng được thiết kế để dễ dàng cho phép thời gian trong ngày từ đồng hồ có thể hiển thị thời gian tính đến giây sử dụng cho mục đích này. Dưới đây là phương pháp có thể sử dụng.

- Ghi thời gian hiện tại dưới dạng "giờ:phút:giây".
- Sử dụng giá trị giây để xác định giá trị hàng, giải thích 00 là 60.
- Sử dụng giá trị phút để xác định giá trị cột, giải thích 00 là 60.
- Sử dụng giá trị giờ để xác định việc sử dụng Bảng A.1 hay Bảng A.2, dựa trên giá trị là chẵn hay lẻ.

TCVN 9600:2013

VÍ DỤ: Thử nghiệm viên mong muốn chọn điểm bắt đầu trong bảng để chọn mẫu ngẫu nhiên từ lô gồm 100 đơn vị. Thời gian hiện thời được ghi là 10:35:13. Do đó, tọa độ nhập vào bảng là hàng 13 và cột 35 của Bảng A.2 (vì 10 là số chẵn). Chữ số tại vị trí đó là 6 nhưng vì cần hai chữ số để chọn mẫu từ 100 đơn vị nên giá trị từ cột 36 cũng được tính để nhận biết đơn vị mẫu đầu tiên là 66.

A.2.2.2 Cũng có thể sử dụng phương pháp bất kỳ khác đưa ra nguồn ngẫu nhiên các số nguyên phân bố đều trong phạm vi dải từ 1 đến và bao gồm cả 60 như phương pháp đồng xu hay xúc sắc trong 5.2 hoặc thuật toán trên máy tính trong Điều 7 (có thể sử dụng để tạo một danh mục dài các tọa độ nhập ngẫu nhiên để sử dụng vào những lần liên tiếp). Ngoài ra, khi kết thúc việc chọn mẫu từ bảng trong một lần cụ thể, có thể ghi lại các tọa độ nhập ngay tiếp sau và sau đó sử dụng làm điểm bắt đầu cho lần lấy mẫu tiếp theo.

VÍ DỤ: Tiếp tục với ví dụ ở A.2.2.1, giả định rằng mẫu ngẫu nhiên có hoàn lại gồm 10 đơn vị được yêu cầu và hướng được sử dụng để chọn những số này là hướng đi xuống. Mẫu thu được là {66, 13, 10, 45, 32, 22, 41, 49, 22, 99}. Tọa độ của giá trị ngay sau đó là hàng 23 và cột 35. Các giá trị này có thể được ghi lại và sử dụng làm điểm nhập khi mẫu được yêu cầu cho lần tiếp theo.

A.2.2.3 Tọa độ nhập cũng có thể được xác định bằng cách thiết lập chữ số ban đầu dựa trên các giá trị hàng và cột từ A.2.2.1 hoặc A.2.2.2, và các chữ số bổ sung dựa trên giá trị của hàng hiện thời và cột bổ sung dựa trên các số được tạo ngẫu nhiên không hoàn lại trong dải từ 1 đến và kể cả 60 đối với các chữ số $m - 1$ còn lại yêu cầu. Số nhiều chữ số thu được được hình thành theo thứ tự các số nhập trong cột được tạo ra.

VÍ DỤ: Một mẫu ngẫu nhiên từ lô 1 000 được yêu cầu. Hàng và cột đầu tiên được xác định tương ứng là 5 và 11, trong Bảng A.1. Cần thêm hai chữ số nữa và chúng được tạo ra bên ngoài là 1 và 30, thu được tọa độ sau đây tương ứng cho chữ số đầu tiên, thứ hai và thứ ba: (5, 11), (5, 1) và (5, 30). Hướng lấy trong bảng được quyết định là hướng đi xuống. Do đó, số đầu tiên là 511, tiếp theo là 943, 419, 413, 899, 209, ...

A.2.3 Xử lý với các ranh giới của bảng

A.2.3.1 Khi đọc số gồm m chữ số và các chữ số phía bên phải của chữ số đầu tiên vượt quá cột thứ 60 thì coi các cột 1, 2, ... tương ứng là cột 61, 62, Có thể áp dụng nguyên tắc này trong phạm vi bảng hiện hành hoặc bằng cách coi bảng số ngẫu nhiên khác là mở rộng của bảng đầu tiên.

A.2.3.2 Nguyên tắc thông thường là thu được các số ngẫu nhiên tiếp theo bằng cách tăng giá trị hàng thêm một và đọc m chữ số của số đó dựa trên các giá trị cột xác định trước và thứ tự của chúng. Khi giá trị của hàng này vượt quá hàng thứ 60 thì coi hàng 1 là hàng 61 và tăng giá trị cột thêm m trong trường hợp các cột được sử dụng liên tiếp hoặc tăng thêm một trong trường hợp A.2.2.3 được sử dụng để thiết lập các cột, và tiếp tục thu được các số. Có thể áp dụng nguyên tắc này trong phạm vi bảng hiện hành hoặc bằng cách coi bảng số ngẫu nhiên khác là mở rộng của bảng đầu tiên, với điều kiện là nó chưa được sử dụng như theo A.2.3.1.

A.2.4 Hồ sơ đánh giá

Khi hồ sơ được yêu cầu lưu giữ cho mục đích đánh giá bởi cơ quan chịu trách nhiệm hoặc cơ quan chức năng, thì ghi lại cỡ lô và cỡ mẫu.

Ngoài ra, đối với các bảng và sử dụng bảng, ghi lại

- a) hàng ban đầu,
- b) (các) cột ban đầu và thứ tự của chúng,
- c) hướng lấy trong bảng,
- d) bảng ban đầu sử dụng và cách thức mở rộng bởi bảng khác, nếu áp dụng, và
- e) ánh xạ sử dụng.

Bảng A.1 – Số ngẫu nhiên (đối với giá trị lẻ)

Hàng <i>i</i>	Cột <i>j</i>											
	5	10	15	20	25	30	35	40	45	50	55	60
5	95183	14683	96585	84761	65044	65183	55567	28734	19802	56410	79127	02879
	08509	97009	47525	88791	93751	70490	17749	32927	65085	94970	55541	89466
	45448	66819	86936	95349	08657	75106	97487	85268	59208	43206	14898	29083
	02230	00022	46390	76658	91934	64676	42429	96812	30560	99913	72809	66736
	13275	96798	51425	67147	15216	71831	16229	25862	22090	91420	24352	03550
10	44439	33385	95151	92374	14683	00323	57667	78341	09004	80139	81182	87552
	17629	80967	42144	58190	24550	62189	94525	44967	15860	85739	93323	87043
	14328	77127	40397	78105	75031	99553	84296	01482	25738	32761	85035	68873
	96896	02466	86706	09507	66840	68509	38033	90785	75831	98886	00905	48343
	09725	80938	27971	01243	29232	28799	88456	99618	20071	79865	63584	69087
15	55021	37184	69480	56317	19944	56756	37514	86439	69831	15172	81398	69574
	06492	95014	54908	21591	13771	35967	78637	29918	47923	61404	63378	72394
	20604	54145	27781	35157	50127	61025	57344	36615	07766	83959	34546	67011
	20202	58870	67569	71756	76284	30909	87763	21951	67756	82597	15210	04291
	27160	01595	64831	07126	25821	81524	12585	76273	36256	41879	33287	84361
20	95089	78572	87167	65888	93358	23879	84496	16147	31130	96978	80361	85195
	74825	21529	24660	33314	64512	80550	51712	23057	53841	32470	36790	60455
	80338	94074	65731	39470	03907	72355	40407	86049	81583	06786	16673	06017
	16596	43179	42026	94264	28301	29514	60657	21732	21548	28693	15241	68944
	34134	42056	40153	00994	14179	44447	99399	86963	71862	01306	15489	00515
25	01118	98623	33695	49221	97197	21424	91691	09365	62483	98893	22106	45399
	67371	71659	30505	71239	56944	35898	02207	93274	40142	98319	41218	43739
	03485	55173	68477	12348	76971	64800	86498	42059	08942	32931	73896	27772
	33328	74045	25331	37635	39081	28786	20843	32565	24316	17888	47626	69199
	84302	10060	25334	84920	30270	09722	61706	52863	03417	95658	74490	00143
30	94775	52191	94552	99265	55079	64517	16803	13037	50984	14886	04385	67907
	51700	63604	96771	34444	30002	67975	93167	16746	97842	25589	12568	81785
	75920	13260	44283	27735	31134	97100	36706	24404	56970	44575	68832	42374
	32385	28423	46784	59222	17776	57726	56449	32109	11825	57995	91217	12802
	13424	00587	12231	44543	62984	58391	22054	16134	73790	59050	24893	62342
35	90896	00608	31377	53338	84813	76825	92192	24937	81481	01866	22641	21817
	60682	38700	34039	93512	38596	40004	71447	97193	52407	44146	77116	99965
	38746	34667	84499	70915	91391	25660	12328	35273	08135	04799	14489	19984
	51658	18422	05732	91001	98070	13591	44468	88460	66964	24038	93987	66335
	49174	12449	97583	85835	82313	96349	92721	64617	06030	22312	94263	80291
40	44215	21953	21844	44114	93162	51028	29551	66121	63959	97789	44259	90865
	33877	94654	10025	84935	94630	49660	23473	89644	67212	75851	83767	45647
	83411	43288	47832	40488	89085	69731	00790	60182	71358	22571	94204	64211
	27135	82404	52031	44648	97600	72166	70830	27701	01755	00523	01837	31304
	29475	31431	46863	70098	98659	72035	21538	12923	76963	78288	59083	18839
45	56886	38711	66126	16504	87900	74055	46028	84821	83323	35962	27522	87875
	55061	35916	15955	28228	36994	73167	17137	36572	48592	60721	97714	61215
	01646	62126	37253	24997	53016	96515	40536	39311	64151	93960	24053	87645
	41789	28167	90577	84499	25059	90583	09422	87357	55416	81135	41286	92320
	26066	80119	44259	94514	21211	44302	29023	28138	03693	50650	38450	61118
50	63559	20927	12881	25582	07872	28073	59006	55666	68690	59772	25162	87924
	35054	84077	02504	10800	75293	86466	92406	56289	79807	55271	73177	70568
	32826	26937	75563	14290	30078	70820	58639	64900	61699	34974	11738	64065
	07860	90064	91220	46786	45994	47375	70140	35592	05990	58470	82014	05265
	79276	72512	19525	27397	88975	77137	40032	06205	06997	53504	07760	62546
55	41093	04332	68677	27073	94104	58532	53616	32156	66153	00264	36374	15230
	27382	27938	91695	64013	46719	61629	33668	32391	35411	68209	33885	64050
	38984	47230	59448	97802	37987	22733	52199	12325	18625	01271	84870	10911
	51411	44221	93363	48654	42656	99464	08481	98128	66677	89441	66019	75095
	80310	18848	23722	30788	27435	03780	85737	05561	57203	07316	98597	73621
60	40082	39571	89790	65382	01447	15984	60854	72833	87320	95245	40678	89785
	86922	84354	81939	32180	32891	52704	84659	95442	86204	44040	51613	15984
	07932	69932	18796	87070	82202	05372	93506	60697	48535	89027	45719	51567
	20383	07288	50265	48321	63056	35861	80864	86357	51567	68151	11723	06990
	45471	31340	30187	23899	36361	96780	55823	37743	06957	77884	78061	36603

Bảng A.2 – Số ngẫu nhiên (đối với giá trị chẵn)

Hàng <i>i</i>	Cột <i>j</i>											
	5	10	15	20	25	30	35	40	45	50	55	60
5	51326	91644	88971	00664	10776	90888	97107	00930	87438	23714	33246	72109
	96821	43647	41707	91062	20037	47660	34652	82087	29652	51614	11015	66187
	67104	67934	95662	85259	08546	77032	28958	26815	66683	50539	65189	22993
	93315	93110	01022	16079	74364	97582	34309	18699	24437	12839	48005	46478
	45823	44862	86472	77354	95916	36599	52350	89866	98532	69704	33735	13696
10	32573	58513	70797	53560	23115	38325	74380	99917	21721	00323	29402	27080
	74553	41700	25357	17428	66708	83630	42360	16842	40782	21345	88668	95845
	06794	48960	75160	18552	00424	25976	69852	35837	69810	64014	03045	53378
	75606	34848	74458	71100	01512	89662	01391	49140	18048	59282	77344	72419
	97535	55169	47044	81940	16507	12220	16375	65306	38691	75019	55186	47269
15	89809	48811	09494	06576	58258	73155	46712	11106	38382	70893	39503	42937
	88095	48475	07399	02165	98038	97247	48441	93066	20276	72169	49706	56553
	48854	75803	34089	45335	13056	44509	77886	65036	42619	95564	66315	49084
	97951	73015	36585	89007	12069	96858	17241	37189	44193	28270	81624	75256
	48953	47945	81153	10121	69935	60894	23151	00113	40597	71164	61331	10930
20	83334	44992	30210	99370	60425	87391	63244	54828	12155	64082	43773	66172
	31105	74546	74875	51747	41021	22338	49403	20173	75454	62122	37227	35790
	94245	19396	03648	28548	42987	78668	09292	23073	21116	07199	21398	18770
	19221	70541	58195	21383	57877	27812	06504	18289	18606	17680	09218	58984
	33503	45776	88514	10094	28589	47548	64714	96174	10026	87111	29333	77885
25	03602	99374	41918	43875	11258	83646	46042	26986	11003	94756	89972	00805
	43277	65791	12217	23767	09833	66504	82359	95754	40249	71472	61588	04428
	31966	42142	71410	28139	60147	68496	89021	01615	36565	85598	18048	32584
	19193	31952	87947	89521	65225	97987	14794	90695	69314	10359	27881	38183
	98935	48606	27475	73462	66692	11151	21709	32836	92997	33682	12722	38906
30	21880	10506	78478	85067	30375	82944	06660	40750	84252	20463	37184	27248
	81695	06183	08147	01241	16278	54886	36468	21315	49106	10291	16837	40481
	68883	60266	66180	05680	38799	40481	73524	73255	79950	42007	56334	54332
	59547	60741	56065	71467	07193	38784	08169	07389	64049	21355	86589	06583
	91709	73154	86898	20234	05773	47157	72305	20819	57301	89018	74851	50560
35	57195	97283	25156	59277	33608	73937	19341	17262	63955	41678	36229	54204
	03657	71909	82018	83110	21722	03455	30654	57890	18530	60458	57145	08764
	93373	88795	11353	44726	66989	24389	93445	53752	98703	55276	18391	53513
	35158	50868	45055	12180	29993	69555	69613	69358	96861	01667	47738	11964
	38056	14298	10431	53147	76843	32128	46844	23407	62423	01712	46033	64425
40	49152	05010	84942	25483	52825	17485	67614	12493	88626	39589	56044	21968
	80000	90734	70131	19986	34949	76990	48325	39323	66921	89134	21853	18973
	53597	22379	94302	15425	62185	27894	37281	38876	97902	34008	45051	05607
	09151	34061	64751	96631	50373	61603	84917	56084	57647	80898	55489	24602
	99734	68144	63963	73011	22832	98145	31523	60195	34172	40637	60940	51237
45	84547	89655	53120	95599	04602	07968	85748	74914	76227	07158	24432	22963
	18815	26665	25301	67754	88457	19913	96787	71084	14867	03077	89575	66834
	14169	38336	41192	56208	29069	87045	32135	25975	71643	74200	52556	30213
	90528	60501	73201	72999	30355	86428	39401	72077	48056	17853	24894	19838
	99055	42696	14376	24907	06082	61789	03963	64664	09132	87218	64755	46107
50	62530	10183	38149	70004	74983	02092	40704	01062	17000	61170	99026	24025
	74196	77214	89483	43933	80953	81268	46485	23647	98173	55947	96727	86378
	27293	56047	73998	19996	94427	09157	62999	88803	81272	22315	92708	07343
	94220	93209	32369	82003	82433	85790	47632	36285	68771	06006	37556	51601
	68430	23169	58879	97812	39399	71469	40835	04924	30336	59222	06350	45656
55	61949	23031	50698	85772	85990	36942	11098	06636	57547	73247	46229	52551
	57248	90383	23502	22642	80722	38164	12160	51707	22075	20624	91644	08780
	10777	53979	65288	39116	80635	49653	36903	33854	79873	67823	23256	31643
	06717	92287	42775	79274	90874	44006	27312	15909	25276	59863	75607	22277
	09519	67689	13829	30992	44921	67375	94754	95322	25501	78486	99059	62524
60	57335	48704	79426	49770	32989	22640	88230	66598	27685	29719	99930	26181
	14911	08271	21662	40886	53783	76430	41233	44057	28385	21751	51476	64387
	04837	08929	81607	33210	61894	17240	37617	56753	61251	49433	65644	63758
	14430	20139	15027	52208	16440	59911	57566	22227	60109	95260	21388	96686
	68896	64599	91227	55882	60220	70202	73354	34776	55530	20599	45720	75145

Phụ lục B

(tham khảo)

Mã máy tính thuật toán tạo số ngẫu nhiên

B.1 Tổng quan

Bộ mã dưới đây được viết trên ngôn ngữ lập trình C (xem tài liệu tham khảo [6] về chi tiết ngôn ngữ) và đưa ra việc áp dụng các thuật toán mô tả ở Điều 7. Tài liệu tham khảo cho các phần liên quan của Điều 7 được nêu trực tiếp trong phần ghi chú trong các phân đoạn mã lập trình để hỗ trợ việc chuyển dịch sang các ngôn ngữ lập trình khác.

B.2 Chương trình minh họa

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <math.h>

/* Function prototypes */
int SeedGen(void);
double U(void);

/* Global variables */
long ij, Seed, Seed2, S2k;
char str[20];
const long m1=2147483563, m2=2147483399, mm1=2147483562,
        a1=40014, a2=40692, q1=53668, q2=52774, r1=12211, r2=3791;
const double ufac=4.6566130573917691e-10;

/* Demonstration program */
int main(void)
{
    long i, nn, n, Seed1, *A, *ptr;
    char yn[4];
    time_t tnow;

    system("CLS");
    printf("Demonstration program for ISO 24153-1: Random sampling procedures -\n");
    printf("Part 1: Quality control and designed experiment applications (Clause 7)\n");
    printf("\nRandom number and seed algorithms used in a single sampling application\n");

    printf("Lot size: ");
    scanf("%d", &nn);
    printf("Sample size: ");
    scanf("%d", &n);
    printf("\nManual seed (Y/N): ");
    scanf("%s", yn);
    if (yn[0] == 'Y' || yn[0] == 'y') { /* manual seed option */
        printf("\nEnter an integer between 1 and 2147483398 inclusive: ");
        scanf("%d", &Seed);
        Seed1 = Seed; /* 7.1.3 save copy of seed */
        S2k = 0;
        tnow = time(NULL);
        strftime(str, 20, "%Y-%m-%d %H:%M:%S", localtime(&tnow));
    }
    else {
        Seed = SeedGen(); /* call automatic seed function of 7.2 */
    }
}
```

```

Seed1 = Seed;          /* 7.2.5 save copy of seed */
}
Seed2 = Seed;         /* RNG function seed parameters */
ij = -1;              /* RNG function initialization parameter */

/* Create holding array for sample values */
A = (long *) calloc(n, sizeof(long));
if (A == NULL) {
    printf("Array allocation failed\n");
    exit(1);
}

/* Select a random sample (with replacement) */
ptr = A;
for (i = 0; i < n; i++)
    *(ptr++) = 1 + (long)floor(U() * nn); /* 7.3.9 scaled output over (1;nn) */

/* 7.4 Program output with audit details */
printf("\nLot size:      %d\n", nn);
printf("Sample size:    %d\n", n);
printf("Date and time:   %s\n", str);
printf("Elapsed seconds: %d\n", S2K);
printf("Seed:           %d\n", Seed1);
printf("Selected sample:\n");
for (i = 0; i < n; i++) printf("%8d", A[i]);
printf("\n");
system("PAUSE");
return 0;
}

```

B.3 Hàm tạo giá trị xuất phát ngẫu nhiên tự động

Khi giá trị xuất phát nhập bằng tay được yêu cầu cho mục đích xác nhận hay đánh giá như đề cập ở 7.1.3 thì không gọi hàm này.

```

int SeedGen(void)
{
/* B.1.3 Automatic random seed generator function */
/* 7.2.2 a) computer system date and time capture */
long i, j, k;
struct tm t, *ptr;
time_t tnow, tref;

/* Reference time: 2000-01-01 00:00:00 */
t.tm_year = 2000 - 1900;
t.tm_mon = 0; t.tm_mday = 1; t.tm_hour = 0;
t.tm_min = 0; t.tm_sec = 0; t.tm_isdst = 0;

tref = mktime(&t);
tnow = time(NULL);
ptr = localtime(&tnow);
strftime(str, 20, "%Y-%m-%d %H:%M:%S", ptr);

/* 7.2.2 b) number of complete days since 2000-01-01 00:00:00 */
/* 7.2.2 c) number of seconds since 2000-01-01 00:00:00 */
S2k = (long)difftime(tnow, tref);

/* quasi-random seed generator */
Seed = S2k; /* 7.2.2 d) initial seed */
j = S2k - (S2k / 100) * 100 + 1; /* 7.2.2 e) warm-up value */
for (i = 1; i <= j; i++) { /* 7.2.4 */
    k = Seed / q2; /* 7.2.3 b) RNG #2 */
    Seed = a2 * (Seed - k * q2) - k * r2; /* 7.2.3 c) RNG #2 */
    if (Seed < 0) Seed += m2; /* 7.2.3 d) RNG #2 */
}
}

```

```

}
return Seed; /* 7.2.5 output seed */
/* function additionally modifies global variables str[] and S2k */
}

```

B.4 Hàm tạo số ngẫu nhiên

Khi chọn tùy chọn giá trị xuất phát nhập bằng tay theo 7.1.3 thì tham số "giá trị xuất phát" (và do đó "Seed2") được đặt bằng giá trị nhập bằng tay trước khi gọi hàm này.

```

double U(void)
{
/* B.1.4 Random number generation function */
int j, k, i1;
static long k1, Shuffle[32];

if (ij < 0) { /* 7.3.1 a) initialize shuffling array */
for (j = 39; j >= 0; j--) { /* 7.3.4 fill shuffling array */
k = Seed / q1; /* 7.3.3 b) RNG #1 */
Seed = a1 * (Seed - k * q1) - k * r1; /* 7.3.3 c) RNG #1 */
if (Seed < 0) Seed += m1; /* 7.3.3 d) RNG #1 */
if (j <= 31) Shuffle[j] = Seed; /* 7.3.4 */
} /* final Seed value; input to 7.3.6 a) */
ij = 0; /* disable further initialization */
k1 = Shuffle[0]; /* input to 7.3.6 c) */
}
/* 7.3.6 combined random number generator (CRNG) */
k = Seed / q1; /* 7.3.6 a) RNG #1; input from 7.3.4 */
Seed = a1 * (Seed - k * q1) - k * r1; /* 7.3.6 a) RNG #1 */
if (Seed < 0) Seed += m1; /* 7.3.6 a) RNG #1 */
k = Seed2 / q2; /* 7.3.6 b) RNG #2; input from 7.2.5 */
Seed2 = a2 * (Seed2 - k * q2) - k * r2; /* 7.3.6 b) RNG #1 */
if (Seed2 < 0) Seed2 += m2; /* 7.3.6 b) RNG #1 */
i1 = floor(32.0 * k1 / m1); /* 7.3.6 c) array index calculation */
k1 = Shuffle[i1] - Seed2; /* 7.3.6 d) unscaled temporary output */
Shuffle[i1] = Seed; /* 7.3.6 e) update shuffling array */
if (k1 < 1) k1 += m1; /* 7.3.6 f) unscaled integer output */
return (k1*ufac); /* 7.3.8 scaled real output over (0;1) */
}

```

Phụ lục C

(tham khảo)

Lấy mẫu ngẫu nhiên và mã máy tính ngẫu nhiên hóa

C.1 Giới thiệu

Phụ lục này cung cấp ứng dụng ngôn ngữ lập trình C (xem tài liệu tham khảo [6] về chi tiết ngôn ngữ) cho các thuật toán lựa chọn mô tả ở Điều 8. Mã chương trình của Phụ lục này được viết để minh họa các mô tả trong Điều 8 và không phải là tùy chọn nhất thiết trong cấu trúc của nó.

C.2 Chương trình minh họa

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

/* Function prototypes */
void RandPermNM(long m, long n, long A[]);
void RandPermN(long n, long A[]);
void RandDerang(long n, long A[], long B[]);
void RSWOR(long n, long m, long A[]);
void SRSWORIUls(long n, long m, long B[], long A[]);
void OSRSWORA(long n, long m, long A[]);
void OSRSWORB(long n, long m, long l, long A[]);
void RLS(long n, long A[][]);
long RandIntMN(long m, long n);
long RandInt1N(long n);
long C(long m, long n);
double U(void);

/* Global variables */
long Seed;
const long ka = 16807, kq = 127773, kr = 2836, km = 2147483647;

/* Demonstration program */
int main(void)
{
    long i, j, m, n;
    system("CLS");
    printf("Clause 8 demonstration program\n\n");

    Seed = 543210; m = 1; n = 100;
    printf("8.2 Random integer between m and n\n");
    printf("m, n, i: %d %d %d\n", m, n, RandIntMN(m, n));
    system("PAUSE"); printf("\n");

    Seed = 543210; n = 100;
    printf("8.2 Random integer between 1 and n\n");
    printf("n, i: %d %d\n", n, RandInt1N(n));
    system("PAUSE"); printf("\n");

    Seed = 543210; m = 5; n = 10;
    long A[n];
    for (i = 0; i < n; i++) A[i] = i+1;
    printf("8.3 Random permutation of n items taken m at a time\n");
    RandPermNM(m, n, A);
    printf("m, n: %d %d\n", m, n);
    for (i = 0; i < m; i++) printf("%d ", A[i]);
```

```

printf("\n");
system("PAUSE"); printf("\n");

Seed = 543210; n = 10;
long A1[n];
for (i = 0; i < n; i++) A1[i] = i+1;
printf("8.3 Random permutation of n items taken all at a time\n");
RandPermN(n, A1);
printf("n: %d\n", n);
for (i = 0; i < n; i++) printf("%d ", A1[i]);
printf("\n");
system("PAUSE"); printf("\n");

n = 10;
long A2[n];
printf("8.4 Random derangement\n");
RandDerang(n, A1, A2);
for (i = 0; i < n; i++) printf("%d ", A2[i]);
printf("\n");
system("PAUSE"); printf("\n");

Seed = 543210; n = 5; m = 20;
long A3[n];
printf("8.6 Random sampling without replacement\n");
RSWOR(n, m, A3);
printf("m, n: %d %d\n", m, n);
for (i = 0; i < n; i++) printf("%d ", A3[i]);
printf("\n");
system("PAUSE"); printf("\n");

Seed = 543210; n = 5; m = 20;
long A4[n], A5[m + 2];
for (i = 0; i < m+2; i++) A5[i] = 0;
for (i = 0; i < m; i++) A5[i] = i+1;
printf("8.9 Single random sampling from an initially unknown lot size\n");
SRSWOR1ULS(n, m, A5, A4);
printf("m, n: %d %d\n", m, n);
for (i = 0; i < n; i++) printf("%d ", A4[i]);
printf("\n");
system("PAUSE"); printf("\n");

Seed = 543210; n = 5; m = 20;
long A6[n];
printf("8.10 Method A: ordered single random sampling without replacement\n");
OSRSWORA(n, m, A6);
printf("m, n: %d %d\n", m, n);
for (i = 0; i < n; i++) printf("%d ", A6[i]);
printf("\n");
system("PAUSE"); printf("\n");

Seed = 543210; n = 5; m = 20;
long A7[n+1], b, d;
b = C(m, n); d = RandInt1N(b);
printf("8.10 Method B: ordered single random sampling without replacement\n");
OSRSWORB(n, m, d, A7);
printf("m, n, d, C(m,n): %d %d %d %d\n", m, n, d, b);
for (i = 1; i <= n; i++) printf("%d ", A7[i]);
printf("\n");
system("PAUSE"); printf("\n");

Seed = 543210; n = 8;
long A8[n+1][n+1];
printf("8.15 Random Latin square\n");
RLS(n, A8);
printf("Order: %d\n", n);
for (i = 1; i <= n; i++) {
    for (j = 1; j <= n; j++) {
        printf("%3d ", A8[i][j]);
    }
}

```

```

    }
    printf("\n");
}
printf("\n");
system("PAUSE");

return 0;
}

```

C.3 Các hàm

```

/*****
long RandInt1N(n)
long n;
{
/* 8.2 Random integer in a range (1 to n inclusive) */
return (1 + (long)floor(U() * n));
}
*****/

/*****
long RandIntMN(m, n)
long m, n;
{
/* 8.2 Random integer in a range (m to n inclusive; m < n) */
return (m + (long)floor(U() * (n - m + 1)));
}
*****/

/*****
void RandPermN (n, A)
long n, A[];
{
/* 8.3 Random permutation of N items taken all at a time */
/* A[] is both the input and output array; output in A[] */
long j, k, temp;
for (j = 0; j < n-1; j++) {
k = RandIntMN(j, n-1);
temp = A[j]; A[j] = A[k]; A[k] = temp;
}
return;
}
*****/

/*****
void RandPermNM (m, n, A)
long m, n, A[];
{
/* 8.3 Random permutation of n items taken m at a time */
/* A[] is both the input and output array; output in A[0:m-1] */
long j, k, temp;
if (m == n) {m = n - 1;}
for (j = 0; j < m; j++) {
k = RandIntMN(j, n-1);
temp = A[j]; A[j] = A[k]; A[k] = temp;
}
return;
}
*****/

/*****
void RandDerang (n, A1, A2)
long n, A1[], A2[];
{
/* 8.4 Random derangement */
/* A1[] is the input array; A2[] is the output array */
long i, iFlag, n1;
for (i = 0; i < n; i++)

```



```

    A2[i] = A1[i];      /* make a copy */
    iFlag = 1; n1 = n;
    for (;;) {
        RandPermN(n1, A2);
        for (i = 0; i < n1; i++) {
            if (A2[i] == A1[i]) {iFlag = 0; break;}
        }
        if (iFlag == 0) iFlag = 1;
        else break;
    }
return;
}

/*****/
void RSWOR (n, m, A)
long n, m, A[];
{
/* 8.6 Random sampling without replacement */
/* n = sample size; m = lot size */
/* A[] is the sample output array */
long i, k;
long B[m]; /* array to keep track of units chosen */
for (i = 0; i < m; i++) B[i] = 0;
i = -1;
do {
    k = RandInt1N(m);
    if (B[k] == 0) {
        B[k] = 1; i = i + 1; A[i] = k;
    }
} while (i < n);
return;
}

/*****/
void SRSWORIU LS (n, m, B, A)
long n, m, B[], A[];
{
/* 8.9 Single random sampling from an initially unknown lot size */
/* n = sample size; m = lot size; (n < m) */
/* A[] is the sample output array */
/* B[] simulates a lot of unknown size (0 indicates lot concluded) */
long k, v;
m = 0; /* lot size counter */
for (;;) {
    m = m + 1;
    v = B[m-1];
    if (v == 0) {m = m - 1; break;}
    if (m <= n) A[m-1] = v;
    else {
        k = RandInt1N(m);
        if (k <= n) A[k-1] = v;
    }
}
return;
}

/*****/
void OSRSWORA (n, m, A)
long n, m, A[];
{
/* 8.10 a) Method A: ordered single random sampling without replacement */
/* n = sample size; m = lot size; (n < m) */
/* A[] is the sample output array */
long j, k, k1;
double p, x;
k = m - n; k1 = m; j = 0;
OsrsworA1:
j = j + 1;

```

```

if (j > n) goto OsrsworA3;
x = U(); p = 1.0;
OsrsworA2:
p = p * k / k1;
if (p <= x) {
    A[j-1] = m - k1 + 1; k1 = k1 - 1;
    goto OsrsworA1;
}
else {
    k1 = k1 - 1; k = k - 1;
    goto OsrsworA2;
}
OsrsworA3:
return;
}

/*****/
void OSRSWORB (n, m, lx, A)
long n, m, lx, A[];
{
/* 8.10 b) Method B: ordered single random sampling without replacement */
/* Finds the combination set of m things taken n at a time */
/* for a given lexicographical index. */
/* n = sample size; m = lot size; (n < m) */
/* lx = lexicographical index of combination sought [1 <= lx <= C(m,n)] */
/* A[] is the sample output array */
long i, k, n1, r;
k = 0; n1 = n - 1;
for (i = 1; i < n; i++) {
    A[i] = 0;
    if (i != 1) A[i] = A[i-1];
OsrsworB1:
    A[i] = A[i] + 1;
    r = C(m - A[i], n - i);
    k = k + r;
    if (k < lx) goto OsrsworB1;
    k = k - r;
}
A[n] = A[n1] + lx - k;
return;
}

/*****/
void RLS (n, A)
long n, A[n+1][n+1];
{
/* 8.15 Random Latin square */
/* n = order; A[][] is the output random Latin square */
long B[n+1], h, i, j, k, r, c, x;
for (r = 1; r <= n; r++) {
Rls1:
    for (i = 1; i <= n; i++) B[i] = i;
    j = n;
    for (c = 1; c <= n; c++) {
        i = 0;
Rls2:
        x = floor(U() * j + 1);
        for (h = 1; h <= r-1; h++) {
            if (i > 50) goto Rls1; /* row no good */
            if (A[h][c] == B[x]) {
                i = i + 1; goto Rls2; /* column no good */
            }
        }
        A[r][c] = B[x]; j = j - 1;
        for (k = x; k <= j; k++) B[k] = B[k + 1];
    }
}
return;
}

```

```

}
/*****
long C(m, n)
long m, n;
{
/* 8.10 b) Method B: auxiliary function */
/* Calculates the number of combinations of m things taken n at a time. */
long i, k, x, n1;
n1 = n; k = m - n1;
if (n1 < k) {k = n1; n1 = m - k;}
x = n1 + 1;
if (k == 0) {x = 1;}
if (k >= 2) {
for (i = 2; i <= k; i++)
x = (x * (n1 + i)) / i;
}
return x;
}
*****/
double U(void)
{
/* Source: bibliographic reference [12] */
/* RNG based on: x[i+1] = 16807 * x[i] mod 2147483647 */
/* Included for illustration purposes only */
long k;
k = Seed / kq;
Seed = ka * (Seed - k * kq) - kr * k;
if (Seed < 0) {Seed = Seed + km;}
return (1.0 * Seed / km);
}

```

Thư mục tài liệu tham khảo

- [1] BAYS, C. and DURHAM, S.D. Improving a Poor Random Number Generator. *ACM Transactions on Mathematical Software*, 2 (1), 1976, pp. 59-64 (Cải tiến bộ tạo số ngẫu nhiên kém. *Giao dịch ACM trên phẩm mềm toán học*)
- [2] BISSELL, A.F. Ordered random selection without replacement. *Applied Statistics*, 35, 1986, pp. 73-75 (Chọn ngẫu nhiên theo thứ tự, không hoàn lại. *Thống kê ứng dụng*)
- [3] BUCKLES, B.P. and LYBANON, M. Algorithm 515, Generation of a Vector from the Lexicographical Index. *ACM Transactions on Mathematical Software*, 3 (2), 1977, pp. 180-182 (Tạo vectơ từ chỉ mục từ điển. *Giao dịch ACM trên phẩm mềm toán học*)
- [4] BYERS, J.A. Random selection algorithms for spatial and temporal sampling. *Computers in Biology and Medicine*, 26, 1996, pp. 41-52 (Thuật toán chọn lọc ngẫu nhiên dùng cho lấy mẫu theo không gian và thời gian. *Máy tính trong sinh học và y học*)
- [5] JACOBSON, M.T. and MATTHEWS, P. Generating uniformly distributed random Latin squares. *Journal of Combinatorial Design*, 4, 1996, pp. 405-437 (Tạo hình vuông La Tinh ngẫu nhiên phân bố đều. *Tạp chí thiết kế tổ hợp*)
- [6] ISO/IEC 9899:1999, *Programming languages — C (Ngôn ngữ lập trình C)*
- [7] L'ECUYER, P. An Efficient and Portable Combined Random Number Generator. *Communications of the ACM*, 31 (6), 1988, pp. 742-749, 774 (Bộ tạo số ngẫu nhiên kết hợp hiệu quả và di động. *Truyền thông ACM*)
- [8] MARSAGLIA, G. Random Number Generators. *Journal of Modern Applied Statistical Methods*, 2 (1), 2003, pp. 2-13 (Bộ tạo số ngẫu nhiên. *Tạp chí phương pháp thống kê ứng dụng hiện đại*)
- [9] MCCULLOUGH, B.D. Assessing the Reliability of Statistical Software: Part II. *The American Statistician*, Vol. 53, No. 2 (May), 1999, pp. 149-159 (Đánh giá độ tin cậy của phần mềm thống kê: Phần II. *Nhà thống kê học người Mỹ*)
- [10] MCCULLOUGH, B.D. and WILSON, B. On the Accuracy of Statistical Procedures in Microsoft EXCEL 97. *Computational Statistics and Data Analysis*, 31 (1), 1999, pp. 27-37 (Độ chính xác của quy trình thống kê trong Microsoft EXCEL 97. *Thống kê và phân tích dữ liệu trên máy tính*)
- [11] MCLEOD, A.I. and BELLHOUSE, D.R. A convenient algorithm for drawing a simple random sample. *Applied Statistics*, 32, 1983, pp. 182-184 (Thuật toán thuận lợi cho việc vẽ biến ngẫu nhiên đơn giản. *Thống kê ứng dụng*)
- [12] PARK, S.K. and MILLER, K.W. Random Number Generators: Good Ones are Hard to Find. *Communications of the ACM*, 31 (10), 1988, pp. 1192-1201 (Bộ tạo số ngẫu nhiên: Khó tìm được sản

TCVN 9600:2013

phẩm tốt. *Truyền thông ACM*)

[13] PRESS, W.H., Teukolsky, S.A., Vetterling, W.T., and Flannery, B.P. *Numerical Recipes in Fortran 77: The Art of Scientific Computing*, Second Edition (Volume 1 of Fortran Numerical Recipes), Cambridge University Press, Cambridge, UK, 1992, 2001 (*Bí quyết số trong Fortran 77: Công nghệ Khoa học máy tính*)

[14] SOM, R.K. *A Manual of Sampling Techniques*. Heinemann Educational Books Ltd., London, 1973 (*Sổ tay Kỹ thuật lấy mẫu*)

