Published online on the journal's webpage: **http://journal.ittelkom-pwt.ac.id/index.php/dinda**

Journal of Dinda

Data Science, Information Technology, and Data Analytics

Vol. 4 No. 2 (2024) 135 - 140 E-ISSN: 2809-8064

Comparison of Fisher-Yates Shuffle and Linear Congruent Algorithms for Question Randomization

Nugroho Dwi Saputro[1*]
[1*]Informatics, Faculty of Engineering and Informatics, Universitas PGRI Semarang
[1*]nungputra@upgris.ac.id

Abstract

Randomization of questions becomes important to prevent cheating and ensure that each student works on different questions. Randomization algorithms, such as Fisher-Yates Shuffle and Linear Congruent Method, have been developed for this purpose. This research aims to compare Fisher-Yates algorithm and Linear Congruent algorithm in generating random numbers or permutations. The test is conducted using the Chi-Square method to evaluate the quality of randomness generated by both algorithms. The Chi-Square value of the shuffling results is calculated and compared with the critical value of Chi-Square at a significance level of 0.05 with a degree of freedom (df) of 4, which is 9.488. The results show that the Chi-Square value for the Fisher-Yates algorithm is 3.8 and for the Linear Congruent algorithm is 4.3, both of which are below the critical value. This indicates that there is not enough evidence to reject the Null Hypothesis (H_0), implying that the difference in randomness quality between the two algorithms is not statistically significant. Therefore, both algorithms are considered to have equivalent performance. The decision to choose one of the algorithms can be based on other considerations such as complexity and efficiency. Further research is recommended to explore the performance of the algorithms under different conditions.

Keywords: *Chi-Square, Fisher-Yates, comparison, Linear Congruent*

1. Introduction

The rapid development of technology has an impact on various aspects of life, including education, which has undergone a significant transformation[1] . In the context of education, technology is a vital element that supports school administration, the development of teaching materials, and assists the implementation of teaching and learning in the classroom[2]. The use of technology in education is not only limited to administration but also includes the use of computer science and increasingly diverse learning media. Information and communication technology (ICT) has introduced various innovations in teaching and learning methods, making the education process more effective, efficient and affordable.

One of the significant impacts of technological developments in education is the use of computer-based learning media and learning management systems (LMS). During the COVID-19 pandemic in 2020-2022, the use of LMS has increased dramatically[3]. LMS is a distance learning system that allows interaction between teachers and students through a digital platform. The use of LMS not only helps in delivering teaching materials but also in managing classes, structuring evaluations, and facilitating communication between teachers and students. Over time, the development of distance learning systems has diversified, providing various options for educational institutions to choose the solution that best suits their needs.

Distance learning systems usually include classroom management, teaching materials, and evaluation of student learning outcomes. This evaluation is often done through assignments and exams that contain questions or projects that students have to do independently. In this context, randomization of questions becomes important to prevent cheating and ensure that each student works on different questions.

Received: 31-07-2024 | Accepted: 08-08-2024 | Published: 13-08-2024

Randomization algorithms, such as Fisher-Yates Shuffle and Linear Congruent Method, have been developed for this purpose[4].

This research focuses on comparing two randomization algorithms, namely Fisher-Yates Shuffle and Linear Congruent Method, to determine which one is more effective in item randomization. To compare the accuracy of these two randomization algorithms, the Chi-Square Goodness of Fit Test method was used. This Chi-Square test helps determine whether the distribution of randomization results is close to the uniform distribution expected from an ideal randomization process. Thus, this study aims to evaluate the extent to which the results of each randomization algorithm match the theoretical uniform distribution.

Previous research has extensively discussed the effectiveness of various randomization algorithms in different contexts. However, direct comparisons between Fisher-Yates Shuffle and Linear Congruent Method in the context of question randomization for education are limited. Most studies focus on the use of these algorithms in other contexts, such as data randomization in computing or computer games, without specifically examining applications in education.

Furthermore, previous studies often neglect in-depth statistical analysis to evaluate the quality of the resulting randomization. This study seeks to fill this gap by using the Chi-Square Goodness of Fit test to provide a more comprehensive analysis of the distribution of the randomization results. This is important because a near uniform distribution indicates that the randomization algorithm works well, ensuring fairness in student evaluation.

2. Research Methods

This research uses Python program code to calculate the frequency of occurrence of each permutation generated by two randomization algorithms, namely Fisher-Yates Shuffle and Randomized Quicksort. Fisher-Yates Shuffle is a classic algorithm known for its simplicity and ability to generate almost perfectly random permutations with O(n) time complexity. It works by iterating through the array from back to front, swapping each element with a random element from the previous part of the array. On the other hand, Randomized Quicksort, although better known as a sorting algorithm, can also generate random permutations by using random elements as pivots during the partitioning process, thus creating a random order in the process.

In this study, both algorithms were run to generate a large number of permutations from the same input array. Each generated permutation is recorded and its frequency is calculated. To evaluate the effectiveness of randomization, these frequency results were analyzed using the Chi-Square Goodness of Fit test. The Chi-Square test is used to determine whether the resulting permutation frequency distribution is close to the uniform distribution expected in ideal randomization. A uniform distribution indicates that each permutation has an equal chance of occurring, which is an indicator of good randomization [10].

This analysis is important to ensure that the randomization algorithm works well in the context of applications that require fair and unbiased randomization results, such as in educational evaluation and distance learning systems. By ensuring that the distribution of randomization results is close to uniform, we can be more confident that the algorithm produces truly random and reliable randomization.

2.1 *Algoritma Fisher-Yates Shuffle*

The Fisher-Yates Shuffle algorithm, named after its creators Ronald Fisher and Frank Yates, is used to randomize the order of a given input sequence. This algorithm ensures that each permutation generated has an equal probability of appearing. The method is useful for producing random permutations of numbers from 1 to N in a fair and random manner. The shuffling process involves swapping elements within an array so that each element has an equal chance of occupying any position. Consequently, the Fisher-Yates Shuffle is widely used in various applications, such as card games and computer simulations, to guarantee true randomness [5].

The steps used to generate a random permutation of numbers from 1 to N are as follows[6] :

1. Write down the numbers from 1 to N.
2. Randomly select a number with the variable k from 1 to N (the number of unprocessed numbers).
3. Swap the value of k with one of the unprocessed numbers.
4. Repeat steps 2 and 3, decreasing the value of N by 1 each time, until all numbers have been processed.

5. The sequence of numbers obtained in step 3 represents a random permutation of the initial order.

The basic method of the Fisher-Yates algorithm is evident when all elements in the array have been shuffled, which can be illustrated through a flowchart. as in Figure 1.

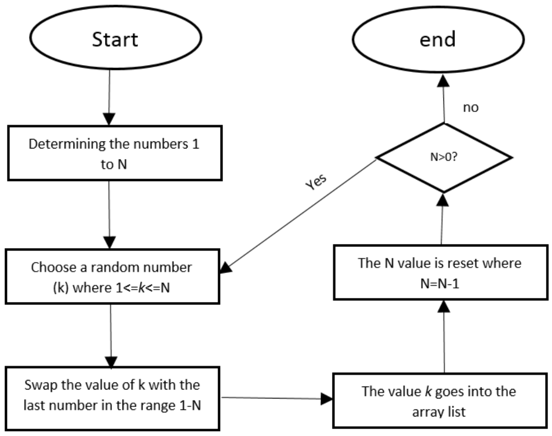

Figure 1. Flowchart of Fisher-Yetes shuflle algorithm

2.2 *Linear Congruent Method*

Linear Congruent Method is one of the algorithms used to generate random numbers. The LCM method is the best known and most widely used algorithm in computer programs to generate random values. The advantages of the LCM method lie in its speed, ease of implementation, as well as the availability of portable code, clear parameters, and reliable test results. The goal of the LCM method is to generate random numbers using a linear model defined by the formula [7] :

$$x_i = (a.x_0 + c) \ mod \ m \qquad (1)$$

x_i = resulting sequence of pseudo-random numbers.
a = multiplier.
x_0 = initial value or seed.
c = an addition
m = modulus

This formula shows that each random number x_i is calculated based on the previous random number x_0, multiplied by the constant aaa, plus the constant c, and then the modulus *m* is taken.

2.2 *Chi-Square Method*

The Chi-Square Goodness of Fit Test is a statistical method used to determine if there is a significant difference between the observed frequency distribution and the expected frequency distribution in one or more categories [8]. This test is often used to test the hypothesis of whether data follows a particular distribution, such as a uniform distribution, normal distribution, or other theoretical distribution.

The Chi-Square Goodness of Fit Test method has 6 steps used.

1.Determine the Expected Distribution.
2.Calculate the Observed Frequency (Oi) and the Expected Frequency (Ei).
3.Calculate the Chi-Square Statistic.
The formula for calculating the Chi-Square value:

$$x^2 = \sum \frac{(Oi-Ei)^2}{Ei} \qquad (2)$$

x^2 = chi-square value
Oi = Observed Frequency
Ei = Expected Frequency

4.Determine the Degrees of Freedom

$$df{=}k{-}1 \qquad (3)$$

df = Degrees of freedom
k = number of categories

5. Compare with Critical Values.

Use the Chi-Square table to determine the critical value (x^2_{crit}) based on the significance level (α) and degrees of freedom.

3. Results and Discussion

The dataset used in each algorithm is only 5 sampling to find the frequency of similar values and with 100 trials. Setting the initial seed or x_0 is an important practice so that the observation frequency value (Oi) remains consistent when tested, the initial seed value set is 42. This consistency allows for a controlled comparison between the Fisher-Yates and Linear Congruent algorithms, as it ensures that any observed differences in the output can be attributed to the algorithms themselves rather than variations in the initial conditions. By standardizing the initial seed, the randomness generated by each algorithm can be reliably assessed, making the results more robust and reproducible. This approach is particularly important in applications where reproducibility and consistency are critical, such as in scientific research, cryptographic processes, and simulations. It allows researchers to have confidence in the stability and reliability of the algorithms under scrutiny, ultimately leading to more dependable conclusions and informed decisions regarding their use in practical applications.

3.1 *Determine H_0 and H_1*

Based on the calculation of this test if the Chi-Square value calculated from the data exceeds either for Fisher-Yates Algorithm or Linear Congruent Algorithm means, the Null Hypothesis (H_0) is rejected, hence the Alternative Hypothesis (H_1). If based on the results of the Chi-Square calculation it is less than or equal to, then there is not enough evidence to reject the Null Hypothesis. To determine the limit of rejecting the null hypothesis (H0), the researcher took a value of 5% or 0.05 significance level value.

This level of significance is a common standard in hypothesis testing, indicating that there is a 5% risk of concluding that a difference exists when there is no actual difference. This rigorous criterion helps ensure that the results are not due to chance and that any detected differences are likely to be meaningful and reliable. Thus, by adhering to this significance level, the researcher can confidently determine the validity of the algorithms' performance and their applicability in real-world scenarios where high-quality randomness is essential.

3.2 *Calculate Oi and Ei*

At this stage, researchers used the python programming language to calculate the frequency of observations generated by each algorithm with 100 trials.

```
Program count Oi Fisher Yet
    def fisher_yates_shuffle(data):
        array = data[:]
        n = len(array)
        for i in range(n-1, 0, -1):
            j = random.randint(0, i)
            array[i], array[j] = array[j],
  array[i]
      return array
    data = [1, 2, 3, 4, 5]
    random.seed(42)
    frequency = {1: 0, 2: 0, 3: 0, 4: 0, 5:
  0}

    for _ in range(100):
        shuffled_data                       =
  fisher_yates_shuffle(data)
        first_element = shuffled_data[0]
      frequency[first_element] += 1
```

After the program is run, the Oi value is as follows:

Table 1. Table *Oi* algoritma fisher yet

Elemen	**Observation Frequency (Oi)**
1	19
2	17
3	15
4	25
5	24

This table 1 explains that Results of chi-square testing on randomization results using the Fisher-Yates algorithm for five categories. The observed frequencies are as follows: value 1 appears 19 times, value 2 appears 17 times, value 3 appears 15 times, value 4 appears 25 times, and value 5 appears 24 times, with a total frequency of 100.

```
Program count Oi Linier Congruent
def __init__(self, seed, a, c, m):
        self.seed = seed
        self.a = a
        self.c = c
        self.m = m
        self.current = seed

    def next(self):
       self.current = (self.a * self.current
+ self.c) % self.m
        return self.current

    def random(self):
        return self.next() / self.m
seed = 42
a = 1664525
c = 1013904223
m = 2**32
lc = LinearCongruential (seed, a, c, m)

def lc_shuffle(data, lc):
    array = data[:]
    n = len(array)
    for i in range(n-1, 0, -1):
        j = int(lc.random() * (i + 1))
           array[i],  array[j]  =  array[j],
array[i]
return array
   data = [1, 2, 3, 4, 5]

# Simulasi 100 kali shuffle dan catat elemen
pertama
frequency = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0}

for _ in range(100):
    shuffled_data = lcg_shuffle(data, lcg)
    first_element = shuffled_data[0]
    frequency[first_element] += 1
```

After the program is run, the Oi value is as follows:

Table 2. Table *Oi* algoritma linier congruent

Elemen	**Observation Frequency (Oi)**
1	23
2	26
3	16
4	15

5	20

This table 2 explains that chi-square testing of randomization results using the Linear Congruent algorithm for five categories. The observed frequencies are as follows: value 1 appears 23 times, value 2 appears 26 times, value 3 appears 16 times, value 4 appears 15 times, and value 5 appears 20 times, with a total frequency of 100. To test whether this distribution corresponds to the expected uniform distribution, we calculate the expected frequency, that is each category is expected to appear 20 times.

The value of Ei is obtained from the number of trials divided by the number of elements or categories, hence:

$$E_i = \frac{\text{Total Number of Trials}}{Number\ of\ Elements\ or\ Catergories} = \frac{100}{5} = 20$$

3.3 *Calculate Chi-square*

This stage takes the previous calculation results and enters them into the following formula:

$$x^2 = \sum \frac{(Oi - Ei)^2}{Ei}$$

The resulting chi square for each element is shown in the following table:

Table 3. Table of fisher's chi square values

	Chi-square value
x_1^2	0,05
x_2^2	0,45
x_3^2	1,25
x_4^2	1,25
x_5^2	0,8
Total x^2	**3,8**

Table 4. Table of congruent linear chi square values

	Chi-square value
x_1^2	0,45
x_2^2	1,8
x_3^2	0,8
x_4^2	1,25
x_5^2	0
Total x^2	**4,3**

The chi-square test results for the Fisher-Yates algorithm and the Linear Congruent algorithm show that the Fisher-Yates algorithm obtained a chi-square value of 3.8, while the Linear Congruent algorithm obtained a chi-square value of 4.3. In the test results table, the Fisher-Yates algorithm is placed in table 3, and the Linear Congruent algorithm is placed in table 4. These values indicate how far the observed frequency distribution deviates from the expected distribution for each algorithm. Based on the chi-square values obtained, the two algorithms do not show significant differences with the expected distribution, because these values are still smaller than the critical value determined for 4 degrees of freedom at a significance level of 0.05.

3.4 *Determining Degrees of Freedom*

The next step in hypothesis testing involves comparing the statistical test results with the Chi-Square critical value. For this test, the Chi-Square critical value is calculated based on the degrees of freedom (df) and the predetermined significance level. With a degree of freedom of df = 5 - 1 being 4 and a significance level of 0.05, the critical value according to the relevant Chi-Square table is 9.488. This means that, in the context of this test, if the Chi-Square value calculated from the data exceeds 9.488, the Null Hypothesis will be rejected. Conversely, if the calculated Chi-Square value is less than or equal to 9.488, then there is not enough evidence to reject the Null Hypothesis. This critical value provides the limit or threshold used to determine whether the results obtained from the data are statistically significant or not at the 95% confidence level.

In practical terms, this threshold serves as a benchmark for researchers to assess the randomness quality of different algorithms. For instance, in our study, the Chi-Square values for both the Fisher-Yates and Linear Congruent algorithms were found to be below this critical value, indicating no significant deviation from expected randomness. Such evaluations are crucial in fields where the integrity of random sequence generation directly impacts outcomes, such as in cryptographic key generation, random sampling in surveys, and Monte Carlo simulations. Ensuring that an algorithm meets the necessary randomness criteria supports its reliability and effectiveness in various applications, reinforcing the importance of statistical testing in algorithm development and validation processes.

3.5 *Comparing Critical Values*

In this research, the critical value of Chi-Square for degree of freedom (df) = 4 and significance level 0.05 is 9.488 [9]. The Fisher-Yates algorithm comparison results produced a Chi-Square value of 3.8, while the Linear Congruent algorithm produced a Chi-Square value of 4.3. Both of these test results were compared with the predetermined critical values. The Chi-Square value of the Fisher-Yates algorithm, 3.8, and the Chi-Square value of the Linear Congruent algorithm, 4.3, are both below the critical value of 9.488. This means that there is not enough evidence to reject the Null Hypothesis for both algorithms. Thus, we conclude that the observed difference in randomness quality or performance between the Fisher-Yates algorithm and the Linear Congruent algorithm is not statistically significant at the 0.05 significance level.

Furthermore, this finding suggests that both the Fisher-Yates and Linear Congruent algorithms perform similarly in terms of generating random sequences when evaluated against the Chi-Square test. The lack of significant difference implies that either algorithm can be reliably used in applications where randomness is crucial, without a substantial impact on the outcome. This is particularly important in fields such as cryptography, simulations, and statistical sampling, where the quality of randomness directly affects the validity and reliability of the results. Future research could explore other statistical tests or different significance levels to further validate these findings or examine the performance of these algorithms under varying conditions and constraints.

4. Conclusion

The conclusion of this research is that there is no significant difference between Fisher-Yates algorithm and Linear Congruent algorithm in the context of this test. Both algorithms show comparable performance in generating random numbers or random permutations based on the Chi-Square values obtained. Therefore, there is no statistical reason to claim that one algorithm is overall better than the other in terms of randomness quality. The choice of the best algorithm will largely depend on the context of use and the specific needs of the task at hand.

Reference

[1] P. R. Foina, “The impact of technology on education,” *Inclusão Soc.*, vol. 17, no. 2, Apr. 2024.

[2] Ariana, S. Andini, B. R. Dwi Putra, and T. Darmansah, “Implementasi Teknologi Informasi dalam Manajemen Administrasi Perkantoran di Sekolah MAS Al-Wasliyah Tembung,” *Transform. J. Econ. Bus. Manag.*, vol. 3, no. 2, pp. 92–99, Jun. 2024,.

[3] L. Christopher, M., “Impact of Learning Management Systems (LMS) toward Students during the Covid-19 Pandemic.,” 2024.

[4] H. Priya and S. Chakraborty, “Randomized Algorithms,” in *Science and Technology - Recent Updates and Future Prospects Vol. 6*, B P International, 2024, pp. 143–165.

[5] and K. S. . B. E. Ekojono, R. . Cahyaningrum, “Implementasi Metode Fisher-Yates Shuffle Dan Fuzzy Tsukamoto Pada Game 2D Gopoh Berbasis Android”,” *JIP*, vol. 4, no. 3, pp. 174–180, 2018.

[6] A. Maulana, F. Fauziah, and R. T. Komalasari, “PENERAPAN ALGORITMA FISHER-YATES UNTUK MENGACAK SOAL PENERIMAAN FORUM STUDI MAHASISWA INFORMATIKA UNIVERSITAS NASIONAL,” *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 5, no. 2, p. 104, Nov. 2020,.

[7] R. Pebrian, F. Fauziah, and I. D. Sholihati, “Algoritma Linear Congruent Method Dan Algoritma Fisher-Yates Shuffle Pada Kuis Ketangkasan Berbasis Android,” *JIPI (Jurnal Ilm. Penelit. dan Pembelajaran Inform.*, vol. 6, no. 2, pp. 167–177, Nov. 2021.

[8] M. L. Hagel, F. Trutzenberg, and M. Eid, “Applying the Robust Chi-Square Goodness-of-Fit Test to Multilevel Multitrait-Multimethod Models: A Monte Carlo Simulation Study on Statistical Performance,” *Psychol. Int.*, vol. 6, no. 2, pp. 462–491, Mar. 2024.

[9] Singhal, Richa & Rana, Rakesh. (2015). Chi-square test and its application in hypothesis testing. Journal of the Practice of Cardiovascular Sciences. 1. 10.4103/2395-5414.157577.

[10] Chi-square Goodness of Fit Test. In: The Concise Encyclopedia of Statistics, Springer, New York, NY, 2008.