Testing randomness: Implementing poker approaches with hands of four numbers

Wael M. F. Abdel-Rehim [1], Ismail A. Ismail [2] and Ehab Morsy [3]

[1] Department of Mathematics and Computer Science, Faculty of Science, Suez Canal University, Suez, Egypt

[2] College of Computers and Informatics, Misr International University, Cairo, Egypt

[3] Department of mathematics, Suez Canal University, Ismailia 22541, Egypt

Abstract

In this paper we discuss the problem of testing randomness motivated by the need to evaluate of the quality of different random number generators which may not generate a true random numbers. Such number generators are used by many practical applications including computer simulations, cryptography, and communications industry, where the quality of the randomness of the generated numbers affects the quality of these applications. In this paper we concentrate with one of the most popular approaches for testing randomness, Poker test. In particular, two versions of Poker test are known: the classical Poker test and the approximated Poker test, where the latter has been motivated by the difficulties involved in implementing the classical approach at the time it is designed. Given a sequence of random numbers to be tested, the basic Poker approach divides this sequence into groups of five numbers, observes which of the possible patterns is matched by each quintuple, computes the occurring probability of each of these patterns, and finally applies Chi-square test to check the randomness of such sequence. In [11], we showed such approach can be implemented with no significant extra running time compared with the approximated approach.

In this paper, motivated by certain practical applications such as cryptography and simulation we implement the classical Poker test (respectively, the corresponding approximated approach) that uses hands of four numbers instead of hands of five numbers. The numerical experiments applied hands of four numbers approach show that the running time is significantly less than those applied hands of five numbers approach.

Keywords: *Poker test, randomness, random numbers tests, cryptography, secret keys.*

1. Introduction

Measuring the quality of randomness of a given sequence is a crucial problem that significantly affects the quality of many practical applications such as distributed algorithms, cryptography, statistical sampling, and computer simulation. In other words, the quality of such applications depends on generating unpredictable (random) sequence of quantities. From the practical point of view, such sequence must be of sufficiently large size in the sense that the probability of any particular value being selected must be sufficiently small in order to prevent an adversary from optimizing a search scheme based on such probability.

There are many techniques described in the literature for generating random and pseudorandom bits and numbers. A random bit generator is a device or an algorithm which outputs a sequence of independent and unbiased binary digits. A random bit generator can be used to generate uniformly distributed random numbers. However, generating of random bits is an inefficient procedure in most practical environments (storing and transmitting a large number of random bits are impractical if these are required in applications). We can overcome this difficulty by substituting a random bit generator with a pseudorandom bit generator (PRBG); given a true random binary sequence of length k, PRBG is a deterministic algorithm that outputs a binary sequence of length l>> k which appears to be random. The main idea behind PRBG is to take a small truly random sequence and expand it to a sequence of much larger length in such a way that an adversary cannot distinguish between output sequences of the PRBG and random sequences of length l.

In order to make sure that such generators are secure enough, they should be subjected to a variety of statistical tests designed to detect the specific characteristics expected of random sequences. We now review a number of empirical tests described in the literatures (see [3, 4, 6]).

Autocorrelation Test tests the correlation between numbers and compares the sample correlation to the expected correlation of zero.

Frequency Test develops frequency distribution of individual samples, uses the chi-square test to compare the distribution of the set of numbers generated to a uniform distribution.

Serial Test develops frequency distribution of pairs of samples. Then we compare the actual distribution against this expected distribution, using the chi-square test.

Gap test is used to examine the length of "gaps" between occurrences of samples in a certain range. It determines the length of consecutive subsequences with samples not in a specific range.

Runs Test tests the runs up and down or the runs above and below the mean by comparing the actual values to expected values. The statistic for comparison is the chi-square.

Poker Test (to be explained in details in the next section) treats numbers grouped together as a poker's hand. Then the hands obtained are compared to what is expected using the chi-square test (see [8, 10, 13]).

Note that, such techniques help detect certain kinds of weaknesses the generator may have by taking a sample output sequence of the generator and subjecting it to various statistical tests; each statistical test determines whether the sequence possesses a certain property that a truly random sequence would be likely to exhibit. That is, the conclusion of each test is not definite, but rather probabilistic. If the sequence is deemed to have failed any one of the statistical tests, the generator may be rejected as being non-random; alternatively, the generator may be subjected to further testing.

2. Poker test

In this section we present in details the two versions of Poker test, the classical Poker test and the approximated Poker test.

2.1 Classical Poker test

The classical Poker test consists of using all possible categories obtained from poker that uses hands of five numbers, i.e., AAAAA (five of a kind), AAAAB (four of a kind), AAABB (full house), AAABC (three of a kind), AABBC (two pairs), AABCD (one pair), and ABCDE (bust). In general, the poker test using hands of five numbers considers n groups of five successive integers denoted by (X5i, X5i+1,…, X5i+4), $0 \leq i \leq n$, and then observes which of the seven possible patterns is matched by each quintuple. The following table summarizes such patterns and their corresponding probabilities.

Name	Pattern	Probability
All different	ABCDE	0.3024
One Pair	AABCD	0.5040
Two pairs	AABBC	0.1080
Three of a kind	AAABC	0.0720
Full house	AAABB	0.0090
Four of a kind	AAAAB	0.0045
Five of a kind	AAAAA	0.0001

It is well known that the poker test can apply with is not restricted to hands of five numbers [7]. In particular, Poker test that uses hands of four numbers is more convenient to be applied to certain applications such as simulation [14], and cryptography [1, 9] in which we need to generate random integers or a random sequence of bits. For example, in cryptography, secret keys (used for encryption of messages or other purposes) are generated using random number generators (RNGs)[9]. Thus we want to apply Poker test to bit streams (typically represented by a 32-bit or 64-bit unsigned integer) rather than floating point numbers, and since 64 bits is not evenly divisible by five we use the closest number that divides 64: four. That is, the generated sequence of random numbers is divided into segments of four bits. With four distinct elements, the following classes of poker hands will be represented: four of a kind (AAAA), three of a kind (AAAB), two pairs (AABB), one pair (AABC), and a bust (ABCD).

A Chi-square test is based on the number of quintuple in each category. We count the number of occurrences in each k-tuples, and then use a chi- square analysis against the theoretical probabilities to determine whether the stack represents a fair poker deck. The theoretical probabilities of such five categories can be computed in a similar way of that applied to the case of seven categories. For the sake of completeness, we compute such probabilities in details as follows. Clearly, the probability of choosing any number equals 1/4.

1) The probability of choosing four of a kind = $(\frac{10}{10} \times \frac{1}{10} \times \frac{1}{10} \times \frac{1}{10} \times \frac{4!}{4!\,0!}) = 0.001$

2) The probability of choosing three of a kind = $(\frac{10}{10} \times \frac{1}{10} \times \frac{1}{10} \times \frac{9}{10} \times \frac{4!}{3!\,1!}) = 0.036$

3) The probability of choosing two pairs = $(\frac{10}{10} \times \frac{1}{10} \times \frac{9}{10} \times \frac{1}{10} \times \frac{1}{2} \frac{4!}{2!\,2!}) = 0.027$

4) The probability of choosing one pair = $(\frac{10}{10}\times\frac{1}{10}\times\frac{9}{10}\times\frac{8}{10}\times\frac{4!}{2!\,2!})$ = 0.432

5) The probability of choosing a bust = $(\frac{10}{10}\times\frac{9}{10}\times\frac{8}{10}\times\frac{7}{10}\times\frac{4!}{4!})$ = 0.504

2.2 Approximated Poker test

At the time the classical Poker test is designed, checking the occurrences of these subsequences of length five using a computer program creates difficulties for the programmers as they have no one systematic similarity. In other words, the running time of such computations would be needed years using primitive computing machines. This motivates constructing a simpler version of the classical test to overcome the programming difficulties involved.

A good compromise would simply be to count the number of distinct values in the set of five [3], [14]. Namely, corresponding to the classical Poker test that uses hands of five numbers we get five categories, 1different, 2different, 3 different, 4 different and 5 different. Thus, a finite time algorithms have been designed to implement such modified Poker test [2], [6].

This breakdown is easier to determine systematically, and the test is nearly as good. In general, we consider n groups of k successive numbers, and then count the number of k-tuples with r different values. A chi-square test is then made using the following probability of the existence of r different.

$$\Pr = \frac{d(d-1)\ldots(d-r+1)}{d^{k}}\begin{Bmatrix}k\\r\end{Bmatrix} \qquad (1)$$

where $\begin{Bmatrix}k\\r\end{Bmatrix}$ denote the Stirling number of the second kind [12] (the number of ways to partition a set of k elements into exactly r parts). The Stirling number can be computed using a well known formula. The following table summarizes the values for the Stirling numbers for k=4 and r=1, 2, 3, 4.

r	1	2	3	4
$\begin{Bmatrix}4\\r\end{Bmatrix}$	1	7	6	1

The classical Poker test with hands of four numbers attains a corresponding approximated version based on Stirling number by considering only four categories, 1 different, 2 different, 3 different, and 4 different.

It is now possible to make a table with number of special quintuples and the measured number. To calculate the expected values we use equation (1) with d=10. Now, we determine theoretical probabilities of such categories.

$$\Pr(1\,\text{different}) = \frac{10}{10^{4}}\begin{Bmatrix}4\\1\end{Bmatrix} = 0.001$$

$$\Pr(2\,\text{different}) = \frac{10(10-1)}{10^{4}}\begin{Bmatrix}4\\2\end{Bmatrix} = 0.063$$

$$\Pr(3\,\text{different}) = \frac{10(10-1)(10-2)}{10^{4}}\begin{Bmatrix}4\\3\end{Bmatrix} = 0.432$$

$$\Pr(4\,\text{different}) = \frac{10(10-1)(10-2)(10-3)}{10^{4}}\begin{Bmatrix}4\\4\end{Bmatrix} = 0.504$$

Then different hands obtained can be compared to what is expected using the chi-square test to see how far the data has strayed from the theoretical distribution.

3. Experimental results

In this section we implement and compare the running time of the classical Poker test that uses hands of four numbers and that uses hands of five numbers and the corresponding approximated versions. We evaluate both versions of the test by implementing programs using C++ code that create random numbers and count the occurrence of these differences or count number of occurrences, then classified each to possible type of poker hand. Finally, it determines the chi-square. The experimental results are reported on PC 2.4 GHz, 1024 MB of RAM, 256 KB of cache.

Example 1: We implement the classical Poker test (with hands of four numbers) on the five million digits (1,250,000 Poker hands). The degrees of freedom (df) for chi-square table equals 4 (one less than the number of cells/categories). If the computed value of chi-square is equal to or greater than the tabled critical value at the prespecified level of significance, then the null hypothesis is rejected and hence the distribution is not truly random. Otherwise (the computed value of chi-square is less than the tabled critical value), the null hypothesis is retained. That is, the data is consistent with the series being random.

Table 1: Chi-Square analysis of Poker test with hands of 4 numbers

Cell/Poker Hand	Observed number of hands (O)	Expected number of hands (E)	$(O-E)^2$ / E
Busts (All different)	629854	630000	0.0338
One pair	539919	540000	0.0122
Two pair	33650	33750	0.2963
Three of a kind	45333	45000	2.4642
Four of a kind	1244	1250	0.0288
Sums	1250000	1250000	X^2 = 2.84

For df =4, we get $X^2_{.05} = 9.49$ and $X^2_{.01} = 13.3$. Since the obtained value $X^2 = 4.77$ is less than $X^2_{.05} = 9.49$, the null hypothesis is retained. This implies that the underlying data is truly random.

Example 2: Now we implement the modified version of the Poker test (with hands of four numbers) on the five million digits (1,250,000 Poker hands)

Table 2: Chi-Square analysis of modified approach (Stirling) with hands of 4 numbers

Cell/Poker Hand	Observed number of hands (O)	Expected number of hands (E)	$(O – E)^2 / E$
1 different	1305	1250	2.4200
2 different	78819	78750	0.0605
3 different	539884	540000	0.0249
4 different	629992	630000	0.0001
Sums	1250000	1250000	$X^2 = 2.51$

For df =3, we get $X^2_{.05} = 7.81$ and $X^2_{.01} = 11.3$. Since the obtained value $X^2 = 2.87$ is less than $X^2_{.05} = 7.81$, the null hypothesis is retained. Thus, the data is consistent with the series being random.

Now, we analyze Chi-Square for both the classical and the modified Poker test approaches described in Figures 3 and 4. We apply the two methods to ensemble of different size and checked the results to see if they are within a specified confidence level. The results are shown in the following table.

Table 3:Chi-Square analysis for Poker test with hands of 5and 4 numbers

Random No.	Poker test with hands of 5 numbers Chi-Square value	Poker test with hands of 4 numbers Chi-Square value
1000	7.93	4.29
5000	7.1	1.91
10000	3.42	6.43
50000	3.63	2.09
100000	3.51	1.71
500000	7.64	5.34
1000000	2.63	5.49
5000000	1.65	2.84
10000000	3.37	6.2

We observe that df= 6 (respectively, df=4) for the classical poker test with hands of 5 (respectively, 4) numbers. Our chi-squared values is less than the critical value for the 0.05 significance level (12.9 to be precise in the case hands of 5 numbers and 9.49 in the case hands of 4 numbers), we accept the null hypothesis as true and conclude that the two methods seemed to produce acceptable chi-square statistics. The chi-squares were within the 95% confidence interval.

Finally, we analyze the running time of the classical Poker test with hands of 5 and 4 numbers and the corresponding modified Poker test described in Figures 3 and 4. We determine the running time of all these approaches in milliseconds. The resulting running time for the classical approaches is shown in the following table.

Table 4: Running time in milliseconds for Poker test with hands of 5 and 4 numbers

No. of random numbers	Poker test with hands of 5 numbers	Poker test with hands of 4 numbers
1000	*47*	32
5000	*78*	47
10000	*93*	79
50000	*172*	140
100000	*219*	204
500000	*359*	234
1000000	*1375*	256
5000000	*3219*	297
10000000	*6047*	437

The results of table 4 (shown in Figure 1) imply there is a significant improvement in term of the running time in the case of applying the classical Poker test with hands of 4 numbers, especially when the number of random numbers is sufficiently large.

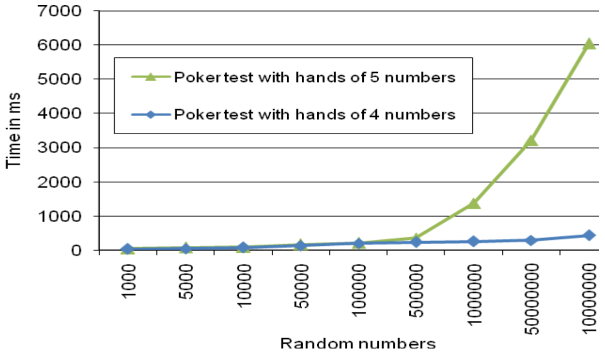

Figure 1: Performance comparison of the classical poker approaches with hands of 5 and 4 numbers in term of their implementing times.

Table 5: Chi-Square analysis for the modified Poker test with hands of 5 and 4 numbers

Random No.	Modified approach with hands of 4 numbers Chi-Square value	Modified approach with hands of 4 numbers Chi-Square value
1000	3.28	1.97
5000	6.85	1.82
10000	2.7	2.34
50000	2.85	1.66
100000	3.28	1.69
500000	2.78	2.29

1000000	1.79	5.34
5000000	1.41	2.51
10000000	1.18	1.41

We observe that df=4 (respectively, df=3) for the modified Poker test with hands of 5 (respectively, 4) numbers. Our chi-squared values is less than the critical value for the 0.05 significance level (9.49 to be precise in the case hands of 5 numbers and 7.81 in the case hands of 4 numbers), we accept the null hypothesis as true and conclude that the two methods seemed to produce acceptable chi-square statistics. The chi-squares were within the 95% confidence interval.

Table 6: Running time in milliseconds for the modified Poker test with hands of 5 and 4 numbers

No. of Random Numbers	***The modified Poker test with hands of 5 numbers***	***The modified Poker test with hands of 4 numbers***
1000	*32*	16
5000	*31*	31
10000	*78*	68
50000	*141*	119
100000	*203*	182
500000	*343*	218
1000000	*1296*	235
5000000	*2984*	278
10000000	*5906*	412

The results of table 6 (shown in Figure 2) imply also that there is a significant improvement in term of the running time in the case of applying the modified Poker test with hands of 4 numbers.

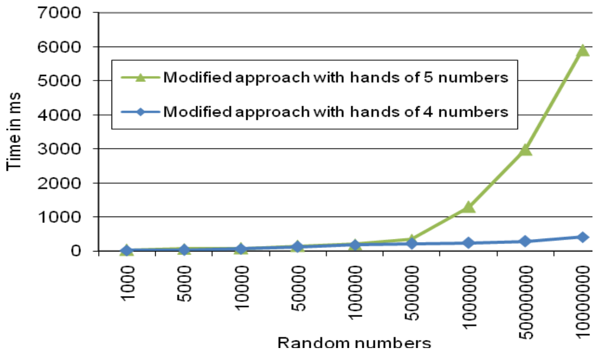

Figure 2: Performance comparison of modified approaches of both hands of 5 and 4 numbers in term of their implementing times.

```
1. Read number of hands to deal
2. Open File to read random numbers
3. do
4. Loads the hands into an array // a sequence of 4 numbers
5. Determine which kind of combination this group of 4 contains
6. Count the number of similar values; breaks at 4
7. Increment the appropriate counter //(5 counters: all different, one pair, two pairs, three of a kind and four of a kind)
8. While (loads the hands < number of hands)
9. Calculate the percentage of the n total repetitions corresponding to each counter
10. Computes the expected theoretical values
11. Compute chi square using the expected probabilities
12. Measure execution time in the program
13. Print "The program execution rime"
14. Print "Chi square"
```

Figure 3: The classical Poker test with hands of 4 numbers pseudo-code algorithm

```
1. Read number of hands to deal
2. Open File to read random numbers
3. do
4. Loads the hands into an array // a sequence of 4 numbers
5. Determine which kind of combination this group of 4 contains
6. Count the number of distinct values; breaks at 4
7. Increment the appropriate counter //(4 counters: 4 different, 3 different and 2 different 1 different)
8. While (loads the hands < number of hands)
9. Calculate the percentage of the n total repetitions corresponding to each counter
10. Computes the expected values using Stirling numbers
11. Compute chi square using the expected probabilities
12. Measure execution time in the program
13. Print "The program execution rime"
14. Print "Chi square"
```

Figure 4: The modified Poker test with hands of 4 numbers pseudo-code algorithm

4. Conclusions

We have been studied Poker test, one of the most popular approaches for testing randomness. We have been compared the performance of implementing the classical Poker approach (respectively, the corresponding approximated approach) that uses hands of four numbers and that uses hands of five numbers. In particular, we have been shown that the running time of implementing the Poker test with hands of four numbers is significantly less than that of hands of five numbers. This encourages us to apply Poker test with four hands instead of five hands, especially in applications involving testing the randomness of a sequences of bit such as cryptography.

References

[1] A. J. Menezes, Paul C van Oorschot, Scott A Vanstone, “Handbook of applied cryptography”, CRC Press, 1997

[2] Andrew karl, “Pseudorandom Numbers: Generation, Statistical Measures, Monte Carlo Methods, and Implementation in C++”, Senior Thesis, Department of Mathematics, University of Notre Dame, 2008.

[3] D. E. Knuth, “The Art of Computer Programming: Seminumerical Algorithms”, Volume 2 (3rd Ed.). Addison-Wesley Longman Publishing Co., Inc., 1997.

[4] David J. Sheskin, “ Handbook of: Parametric and Nonparametric Statistical Procedures”, Chapman& Hall LCRC, 2004.

[5] Deborah J. Bennett, “Randomness”, Harvard University Press, 1998.

[6] John A. Hamilton and David A. Nash, “Distributed Simulation”, CRC Press LLC, 1997.

[7] M G Kendall, and B B Smith, “Randomness and random sampling numbers”, Journal of the Royal Statistical Society 101, 1938, pp.147–166.

[8] Mario Rutti, “A Random Number Generator Test Suite for the C++ Standard”, Diploma Thesis, Institute for Theoretical Physics, ETH Zurich, 2004.

[9] Stefan Brands and Richard Gill, “Cryptography, statistics and pseudo randomness I”, Probability and mathematical statistics, Vol. 15, 1995.

[10] Sorin Talamba, “A Theoretical and Empirical Study of Uniform Pseudo-Random Number Generators” , Senior Thesis, Department of Computer Science, Middlebury College, 2001.

[11] Wael M. F. Abdel-Rehim, Ismail A. Ismail and Ehab Morsy, “Implementing the classical poker approach for Testing Randomness”, (Submitted), 2012.

[12] Eric W. Weisstein, “Stirling Number of the Second Kind”, From Math World-A Wolfram Web Resource-http://mathworld.wolfram.com/StirlingNumberoftheSecondKind.html

[13] William J. Stewart, “Probability, Markov chains, queues, and simulation: the mathematical basis”, Princeton University Press, 2009.

[14] Zaven A. Karian, Edward J. Dudewicz, ”Modern statistical systems and GPSS simulation”, Second Edition, CRC Press LLC, 1998.

Wael Mohamed Fawaz Abdel-Rehim received the B.Sc. degree in Mathematics and Scientific Computations from the Faculty of Science, Suez Canal University, Ismailia, Egypt in 2007. After that he worked as a demonstrator at the Faculty of Science in Suez, Suez Canal University, Egypt. His current research interests are in wireless networks, network simulation, Random number testing, and performance evaluation of computer networks.

Ismail A. Ismail Professor, Dean, College of Computers and informatics, Misr international University, Egypt, received the M.S. and Ph Degrees from the University of Cairo, Egypt, in 1971 and 1976, respectively. His research interests include pattern analysis and machine intelligence, data structures and analysis, genetic algorithms, neural network, and database.

Ehab Morsy received the PhD degree in Combinatorial Optimization from Graduate School of Informatics, Kyoto University, Japan, 2009. After that and till now he worked as an assistant professor at Department of Mathematics, Faculty of Science, Suez Canal University, Egypt. He has 12 publications in various international journals and conferences. His current research interests are Design and Analysis of Graph Algorithms, Distributed Computing, and Combinatorial Optimization.