TWO RIGOROUS STATISTICAL TOOLS FOR OBJECTIVE ANALYSIS OF QUANTUM RANDOM NUMBER GENERATION

Parthasarathy Srinivasan and Tapas Pramanik

Oracle Corporation USA

ABSTRACT

Quantum Random Number generation(Qrng) provides a superior alternative than classical Random Number Generation (Crng) and the two experiments outlined in this work provide validation of this premise. The first experiment utilizes Random Numbers generated using Qrng and CRng to provide data samples as input to an Evolutionary Algorithm (namely Differential Evolution) , which mutates and thresholds these samples using the known rastrigin and rosenbrock functions and evolves the solution pool towards convergence. Rigorous statistical analysis employing p-values is applied to the convergence data to prove that Qrng is indeed Qualitatively superior to Crng (Qrng surpasses Crng by a factor of 2). These results are complemented with yet another experiment wherein the Qrng and Crng samples are generated and statistically compared with,yet another tool namely bottleneck distance , which leads to a logical conclusion consistent with the one obtained in the first experiment (Qrng again surpasses Crng by the same factor of 2 in the range of statistical distances obtained from employing the two Rng methods).

KEYWORDS

Bottleneck distance, p-value, Evolutionary Algorithm, Takens Embedding

.

1. INTRODUCTION

The statistical evaluation of Random Number Generation is known through the works such as that of Juan**[17]** wherein one of the methods utilizes the P-value test statistic as an indicator of the degree of randomness of a given sample of generated random numbers. In this work (as part of experiment#1) this notion (based upon P-values) is extended to compare and contrast random number samples (in terms of similarity of samples derived from Qrng with those derived from Crng) and utilized in a convergence experiment as elaborated below. This evaluation of random number samples is extended by methods employing takens embeddings which form part of modern Topological Data Analysis methods, with the aim of verifying the consistency and integrity of the results and conclusions obtained from an application of the random numbers with those obtained from direct evaluation (of the raw/unprocessed) samples themselves. The experimental details of the two complementary approaches are now introduced and described; and are projected for still further extensions in upcoming experiments and evaluations.

2. WORK CONCEPTION AND COVERAGE

The present work is envisaged as a beginning to the objective analysis and interpretation, of applications of Qrng to solution of problems in diverse areas of scientific and business computing such as solution of randomized partial differential equations, general classifications of

David C. Wyld et al. (Eds): DMSE, CSEIT, NeTCoM, SPM, CIoT, NCS, NLPD, ArIT, CMLA – 2024
pp. 211-220, 2024. CS & IT - CSCP 2024 DOI: 10.5121/csit.2024.141417

Evolutionary Algorithms, Fourier & Newtonian method implementations on Quantum systems, implementations of algorithms in Quantitative Finance using Qrng effectively and many more. This work investigates, implements, and leverages, the superior computing capabilities of the ExaLogic machine, to establish the consistency of superior quality in Quantum Random Number Generation, with Quantum Simulators running across variants of high performance Turing computing hardware, as demonstrated by the two different experiments performed.

3. NOVELTY OF APPROACH IN THIS CURRENT WORK

Through best of research, the authors have not found any scientific work that directly applies the particular comparative techniques introduced in this work to the specific problem of evaluating the Quality of Random number generation and the consequent implications on applications which rely on the same. Following inspirational references were found useful for the above purpose along with the unique technique introduced by the authorss: [1] (section 6.) , [2] (Section 13.3), [17] (Section 3.3)

4. DETAILED DESCRIPTION OF THE TWO EXPERIMENTS

The first experiment performed in this work involves the Evolutionary approach **[4] [6][10]** to solving (converging to the root) of a mathematical function and entails emulating biological processes; wherein random solution candidates (mathematical /numerical analogue of the gene pool) are evolved by mutating and thresholding **[11]** (mathematically). The thresholds employed for demonstration are well known such as rosenbrock and rastrigin, represented mathematically by equations :

$$\boldsymbol{f(x) = \sum_{i=1}^{i=dim} (x_i{}^2 - 10 * cos(2 * \pi * x_i)) + 10 * n \ ,} \quad (1)$$

dim represents number of input dimensions. with n as size of input

$$\boldsymbol{f(x) = \sum_{i=1}^{i=n-1} ((100 * (x_{i+1} - x_i^2)^2) + (x_{i-1})^2),\ -30 <= x_i <= 30,} \quad (2)$$

where n ⇔ input size

Mutation, thresholding, and crossover are performed using the above mathematical functions in this work.

Statistical analysis of the convergence results obtained from performing the above experiment are analyzed and discussed in detail in section 7.4 .

Yet another experiment , utilizing methods inspired from Topological Data Analysis (TDA) is performed as described here .

In this Topological Data Analysis based comparative Experiment(TDA), Random Numbers are generated both with Qrng and Crng paradigms and the phase space for each of the generated samples is obtained by taking the standard Takens' Embedding**[14]** of the random samples. Once the phase space is obtained the standard Bottle Neck distance(computed using the GUDHI library**[15]**) between the samples is computed for comparison.

Analysis of the convergence results obtained from performing the above experiment are discussed in detail in section 7.5 .

This work concludes with a discussion of the consistence in the logical result interpretation from both the experiments described above and emphasizes the formal/empirical/experimental proof of the fact that Qrng surpasses Crng qualitatively.

5. APPARATUS

5.1. For Classical Random Number Generation(Crng) Experimental Aspect

Hardware	Operating System	Library/Software Used	Library Version
Intel based x86 PC	Windows 10	Python	3.9.12
Intel based x86 PC	Windows 10	Numpy	1.22.3

5.2. For Quantum Simulation (of Random Number Generation(Qrng)) Experimental Aspect:

Hardware	Operating System	Library/Software Used	Library Version
ExaLogic x86-64, 48 CPUComputing Platform	Oracle Linux 7	Python	3.9.12
ExaLogic x86-64, 48 CPU Computing Platform	Oracle Linux 7	Q-sharp	0.18.2109.162713

5.3. Apparatus for Topological Data Analysis Experiment(#2)

Hardware	Operating System	Library/Software Used	Library Version
Intel based x86 PC	Windows 10	Python	3.9.12
Intel based x86 PC	Windows 10	Gudhi Library	3.10.1

6. EXPERIMENT IMPLEMENTATION DETAILS FOR THE TWO EXPERIMENTS

6.1. Core Code Snippet Implementing Evolutionary Program

```
popul_size = 50
mutF = 0.5   # mutate threshold
Thrcr = 0.7  # crossing threshold
gen_max = 10
…….
for g in range(gen_max):
for i in range(popul_size):
for k in range(3):
mutation[k] = population[a][k] + F * (population[b][k] - population[c][k])
……..
p = randomNumber
if p < cr:
new_soln[k] = mutation[k]
else:
new_soln[k] = population[i][k]
```

```
# If new solution yields better error introduce new solution in population
new_soln_err = rastrigin_error(new_soln, dim)
if new_soln_err < popln_errors[i]:
population[i] = new_soln

    #Find the Current best solution index
    best_ind = np.argmin(popln_errors)

    #Plot and continuetill convergence is attained.
```

6.2. Critical Code Snippet(S) Employed For The Topological Data Analysis Experiment #2

Steps :

1) Random numbers are generated**[5][7][12]**

Q# code snippet for Quantum Random Number Generation (QRng)

```
operation QuantumRN() : rslt {

    use cub = Qubit();

    H(cub).
.
    return MResetZ(cub);
  }
```

2) Phase space is reconstructed using Takens theorem **[14]**(Refer Fig. 6.5, 6.6)**.**

Python code snippet for “Takens Embedding” of the Random Numbers generated above.

```
  def takens_embe(dat, pm=2, pd=1):
tak_emb = np.array([dat[0:len(dat) - pd*pm]])
   for i in range(1, pm):
tak_emb = np.append(emb, [dat[i*pd:len(dat) - pd*(pm - i)]], axis=0)

   return tak_emb.T
```

3) Topological data analysis using “bottleneck distance” metric is performed and interpreted **[15].**

Python code snippet for Comparative Topological Data Analysis of Qrng and Crng.

```
p0=Rips_simplex_tree_quant0.persist_int_d(1)
p1=Rips_simplex_tree_quant1.persist_int_d(1)
  print(gudh.bottle_dist(p0, p1))
```

7. RESULTS FROM BOTH EXPERIMENTS

7.1. Experiment #1 :

Axis **X** plot max DE generation **[16]** for each iteration
Axis **Y** plot : minimal error population individual index
Following has been established to be true: QRNG convergence **[8]** is more optimal/favorable, for the qsharp simulator implementation on the Exalogic machine also.

For rastrigin (on Exalogic)
QRNG convergence point : (11,7.5) (Refer Fig. 6.1)

Turing convergence point: (12,15) (Refer Fig. 6.2)

For rosenbrock(on Exalogic)

QRNG convergence point : (11, 7) (Refer Fig. 6.3)

Turing convergence point: (12,14) (Refer Fig. 6.4)

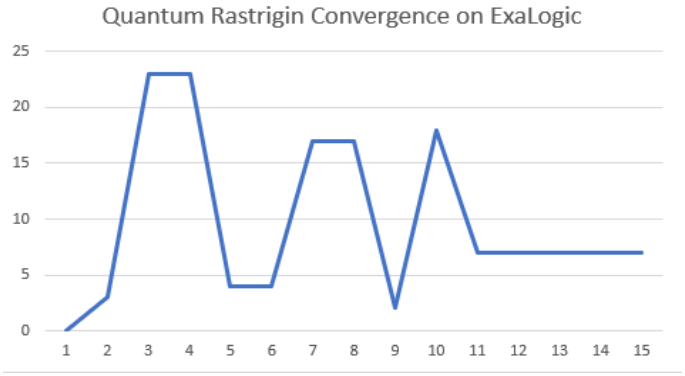

Fig.7.1. Quantum Convergence for rastrigin on Exalogic.

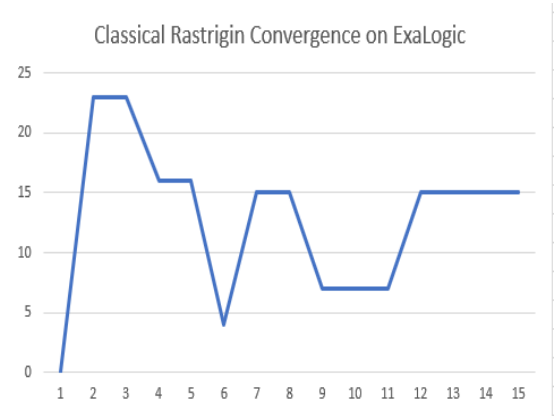

Fig.7.2. Turing Convergence for rastrigin on Exalogic.

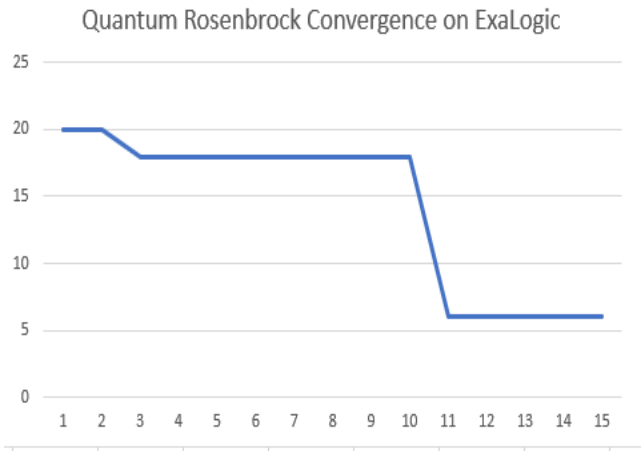

Fig.7.3. Quantum Convergence for rosenbrock on Exalogic.

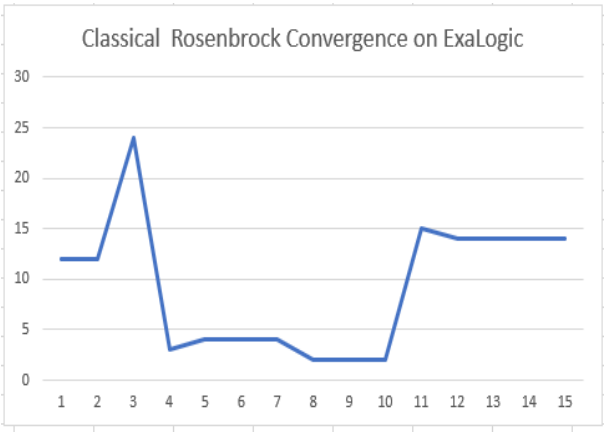

Fig.7.4. Turing Convergence for rosenbrock on Exalogic.

7.2. Experiment #2 :

"Bottleneck Distance" metric from 4 runs each of the TDA program for the Quantum Rng and Classical (Turing) Rng cases :

#QRng Bottleneck Distances : (17.5,25.7,30.46,27.075)(Refer Fig. 6.5, 6.6)

#CRng Bottleneck Distances : (56.38,40.2988,50.1217,38.55)

Sample Plot of QRNG and associated Takens Embedding (Refer Experiment 2 described in Section 4)

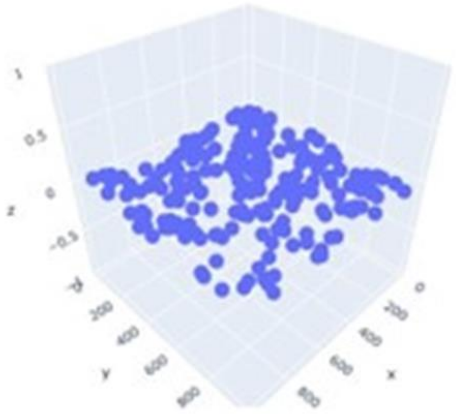

Fig.7.5. Sample Takens Embedding of QRNG data.

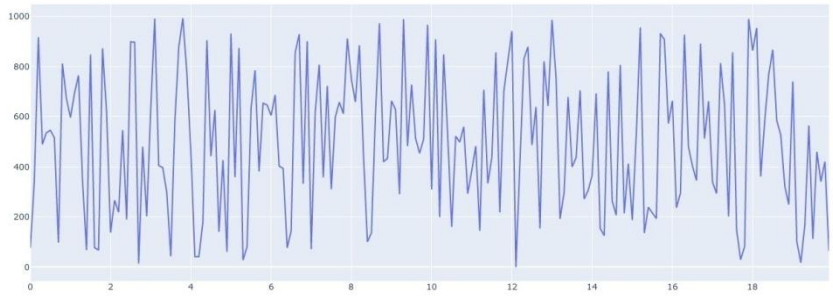

Fig.7.6. Sample plot of Qrng data

8. STATISTICAL RESULT ANALYSIS

8.1. Reason for Choice of p-values (Experiment 1) and Bottleneck Distance (Experiment 2) .

p-values applied to convergence data of the Differential Evolution Algorithm show the effect of employing the higher quality Random numbers obtained from Qrng to a practical application and the resultant impact (quicker convergence).

Bottleneck distance measure applied to Topological Summaries (phase space) of the Random numbers generated from Qrng , serve to complement, and bolster the result obtained from p-value analysis by demonstrating the consistence in the (halving of the range) of the distance measure values for Qrng.

8.2. Statistical Comparison Using P-Values for Experiment #1 :

Comparison : Quantum Rastrigin (on Exalogic) with Quantum Rosen (on Exalogic)

Mann Whitney Test

1 Result : Random value selection from the 2 groups yield equality per test.

2. p-value = 0.1083, (p(x≤Z) = 0.05417). type I error possibility: 0.1083 (10.83%).

Comparison : Quantum Rastrigin (on Exalogic) with Classical Rastrigin (on Exalogic)

Mann Whitney Test

1 Result : Random value selection from the 2 groups yield equality per test.

2. p-value = 0.3541, (p(x≤Z) = 0.1771). type I error possibility: 0.3541 (35.41%).

Experiment #2 (reproduced for ready reference and consistency):

"Bottleneck Distance" metric from 4 runs each of the TDA program for the Quantum Rng and Classical (Turing) Rng cases :

#QRng Bottleneck Distances : (17.5,25.7,30.46,27.075)(Refer Fig. 6.5, 6.6)

#CRng Bottleneck Distances : (56.38,40.2988,50.1217,38.55)

9. INSTRUCTION TO PREPARE THE QSHARP ENVIRONMENT ON EXALOGIC

Install Anaconda from bash script
Update conda
pip install qsharp **[3][9]**
Create Conda Environment at $ANACONDA_SITE_PACAKAGES/qsharp
conda activate $ANACONDA_SITE_PACAKAGES/qsharp
conda install quantum-engineering:qsharp

pip install azure-quantum

cp $ANACONDA_SITE_PACAKAGES/qsharp/lib/libstdc++.so.6.0.26 /usr/lib64 (This step is to fix the libstdc++.so dependency version problem we faced)

10. CONCLUSION

A). Interpretation of Results from Experiment #1

Comparing : p-value 0.3541 (between the 2 quantum groups) ; p-value 0.1083 (between 1 quantum and 1 classical group), shows consistently that the quality of generation of Random numbers by Quantum sources is higher than that by Classical sources. Comparing Type I error possibility <1/2 across the two groups, thereby QRng surpassing Classical Rng.

This establishes the conclusion that the results obtained with the use of Exalogic Hardware are in complete agreement with the results and conclusion of the original study where the quantum simulation was performed on a Windows 10 x86 64 bit PC.

B). Interpretation of Results from Experiment #2 and comparative discussion with Experiment#1

The observations of the bottleneck distance (cost of computing point correspondence between the data sets) as yet another statistical tool for interpretation, show that the Quantum Rng(s) are

closer to each other and resemble each other which is logically consistent with the results obtained from the other experiment(#1) (where the statistical tool employed was the p-value)).

11. RELATED WORK AND ANTICIPATION OF FUTURE DIRECTIONS FOR WORK

1) How do real quantum backends **[13]** surpass the third law of thermodynamics ? and rethink if this is possible to some extent on simulators also (e.g., by possible hybridization etc.) also → how will this help → cost of experimentation vs. tradeoff ? Needs more consideration and exploration.

2) Applying the methods presented to rigorously evaluate potential QRNG based techniques in business and scientific computation. As discussed, the P-value conception in **[17]** is independently extended herein.

3) The Authors are contemplating , based upon the general orientation to the theme of this work w.r.t. takens embedding possibilities of finding alternate validation techniques for the idempotence of the QRngs and CRngs. This is still Work In Progress and will be concluded in a future work. In particular a similarity of this objective is found in **[17]** though the conception and formulation is entirely independent.

REFERENCES

[1] Computational Topology for Data Analysis Tamal Krishna Dey, Yusu Wang , 2016-2021, Cambridge University Press. https://www.cs.purdue.edu/homes/tamaldey/book/CTDAbook/CTDAbook.pdf

[2] A Riemannian Framework for Statistical Analysis of Topological Persistence Diagrams Rushil Anirudh, Vinay Venkataraman, Karthikeyan Natesan Ramamurthy, Pavan Turaga https://doi.org/10.48550/arXiv.1605.08912

[3] Quantum Computation : Microsoft Q# (Q-sharp) local Quantum Simulator https://cloudblogs.microsoft.com/quantum/2023/09/18/azure-quantum-learning-resources-enable-getting-ready-for-a-quantum-supercomputer/

[4] McCaffrey, James. Differential Evolution Optimization. 2021 Sept. https://visualstudiomagazine.com/articles/2021/09/07/differential-evolution-optimization.aspx

[5] Microsoft. Tutorial: Implement a quantum random number generator in Q# . 2023 Jun. https://learn.microsoft.com/en-us/azure/quantum/tutorial-qdk-quantum-random- number-generator?view=qsharp-preview&tabs=tabid-copilot

[6] Charilogis, Vasileios. Modifications for the Differential Evolution Algorithm. 2022 Feb. https://www.mdpi.com/2073-8994/14/3/447

[7] Xiangfan-Ma. Quantum random number generation.2016 Jun. https://www.nature.com/articles/npjqi201621

[8] Medium. Data Convergence — Quantum Computing — Algorithms. 2020 Jan. https://medium.com/data-convergence/data-convergence-quantum-computing-algorithms-4545e1f10f33

[9] Jacak, Marcin M.Quantum generators of random numbers. 2021 Aug. https://www.nature.com/articles/s41598-021-95388-7

[10] Storn. Differential Evolution - A simple and efficient adaptive scheme. 1995 Mar. https://cse.engineering.nyu.edu/~mleung/CS909/s04/Storn95-012.pdf

[11] Xie, Huayang. An Analysis of Selection in Genetic Programming. 2008 Aug. https://homepages.ecs.vuw.ac.nz/~mengjie/students/jasonPhd_thesis.pdf

[12] Anand, Rishabh. Building Your Own Quantum Circuits in Python (With Colorful Diagrams). 2019 May. https://towardsdatascience.com/building-your-own-quantum-circuits-in-python-e9031b548fa7

[13] Quantum theory, the Church-Turing principle, and the universal quantum computer

DAVID DEUTSCH
https://royalsocietypublishing.org/doi/10.1098/rspa.1985.0070
[14] Tiago Toledo Jr.
https://github.com/TNanukem/paper_implementations/blob/main/Takens%20Embedding%20Theorem.ipynb
[15] Vincent Rouvreau
https://github.com/GUDHI/TDA-tutorial/blob/master/Tuto-GUDHI-persistence-diagrams.ipynb
[16] Benchmarking Differential Evolution on a Quantum Simulator, Parthasarathy Srinivasan DOI : 10.54364/AAIML.2023.1196
[17] Statistical Testing of Random Number Generators
Juan Soto https://csrc.nist.rip/nissc/1999/proceeding/papers/p24.pdf