International Journal of Applied Sciences and Smart Technologies

Volume 5, Issue 2, pages 213-228

p-ISSN 2655-8564, e-ISSN 2685-9432

Cloud Quantum Coin-Tossing Gambling

Jose C. Moreno[1*]

[1]*New American Quantum Education Society Operations (NAQESO)*
1340 Reynolds Ave. #116-1070, Irvine, CA 92614, USA
*Morenj13@alumni.uci.edu

(Received 18-05-2023; Revised 30-05-2023; Accepted 08-06-2023)

Abstract

Quantum computers are an alternative way to create multipartite probabilities for a game as a function of participant's inputs. In some situations, quantum gambling could be an improvement over the predictability of certain types of random number generators. However, NISQ computers require a protocol whose expected statistical gains (losses) can be confirmed empirically given the participants' inputs. A zero-sum coin-tossing protocol with Nash equilibrium [1] is tested with a quantum computer where hypothetical players enter parameters, in their respective qubits, and are compensated 1 or R coin(s) after each outcome. In theory, independently of R, the protocol implies that there is no gain improvement for a player when the other maintains the equilibrium parameter; gain is zero or better for the player maintaining it. However, outcomes obtained with several setting combinations imply Nash equilibrium only when R is a small fraction. For $R \gg 1$, given thousands of outcomes, there is Nash-like equilibrium such that a player may not improve gain significantly by changing the parameter if the other maintains it, that is, losses (gains) are considerably minimized with the parameter. The data suggests that gains (losses) would be expected statistical functions of the participants' choices if two played in this manner.

Keywords: NISQ computer, Nash equilibrium, coin-tossing game

1 Introduction

Given the availability of quantum computers through the cloud and their current development, there are tasks that are realizable with a few qubits, such as generating multipartite probabilities as a function of remote inputs. Such a task is the case in quantum gambling protocols [2],[3],[4]. A gambling protocol with a quantum computer provides essentially probabilistic outcomes as a function of the parameters entered by participants. Certainly, quantum gambling can be an alternative to other types of RNGs [5],[6],[7] needed to create multipartite probabilities, and perhaps be an improvement over the predictability of those other types in some situations [8],[9],[10],[11]. On the other hand,

This work is licensed under a Creative Commons Attribution 4.0 International License

Volume 5, Issue 2, pages 213-228
p-ISSN 2655-8564, e-ISSN 2685-9432

games with NISQ computers require evaluation from the participants. “Errors” in the output are expected [12]. External factors can influence outcomes significantly [13]. Theoretical probabilities do not inform the number of repetitions required to verify them. In this way, players must be able to confirm that the gains (losses) result significantly from expected probabilities defined by the player’s choices.

The protocol presented is a variant of two-player coin tossing quantum gambling [14],[15] with Nash equilibrium [1] adapted to a cloud IBM superconducting quantum computer [16] where each participant could operate on one qubit of a two-qubit entanglement. Such a protocol could be realized with actual remote players operating on two qubits. In the present version of the game, the input of both players is required, measurements of the qubits are not performed at the same time, as shown in Fig. 1, and there are Nash equilibrium parameters, selected independently by each player, for which there is zero average gain per game (which will be referred simply as “gain”), or it may be improved, for the one that maintains the corresponding parameter regardless of what the other does, that is, there is no gain improvement for a player if the other is maintaining it. As shown in Fig. 1, The protocol is as follows: player-q[0] “splits” $|0\rangle_{q[0]}$ into a superposition $|\psi\rangle_{q[0]} = \cos\frac{\alpha}{2}|0\rangle_{q[0]} + \sin\frac{\alpha}{2}|1\rangle_{q[0]}$, concealing parameter α. Then, Player-q[1] also “splits” qubit $|0\rangle_{q[1]}$ into two parts, also maintaining the parameter unknown to the other, creating $|\psi\rangle_{q[1]} = \cos\frac{\beta}{2}|0\rangle_{q[1]} + \sin\frac{\beta}{2}|1\rangle_{q[1]}$, but only with $|1\rangle_{q[0]}$, which means that both form

$$\cos\frac{\alpha}{2}|0\rangle_{q[0]} \otimes |0\rangle_{q[1]} + (\sin\frac{\alpha}{2}|1\rangle_{q[0]}) \otimes \left(\cos\frac{\beta}{2}\left|0\rangle_{q[1]} + \sin\frac{\beta}{2}\right| 1\rangle_{q[1]}\right), \qquad (1)$$

and the first measurement is on $q[1]$. The rules for the game are as follow:

1) If the outcome is $|1\rangle_{q[1]}$, then player-q[1] receives one coin,

2) if not, the state of $q[0]$ is projected on a verification state $|\phi^+(\gamma)\rangle$ where γ is always decided by the two players before starting the game. If the state is verified, then player-q[0] receives R coin(s) ($R > 0$); otherwise, player-q[1] receives them.

International Journal of Applied Sciences and Smart Technologies

Volume 5, Issue 2, pages 213-228

p-ISSN 2655-8564, e-ISSN 2685-9432

Quantum Computer game protocol

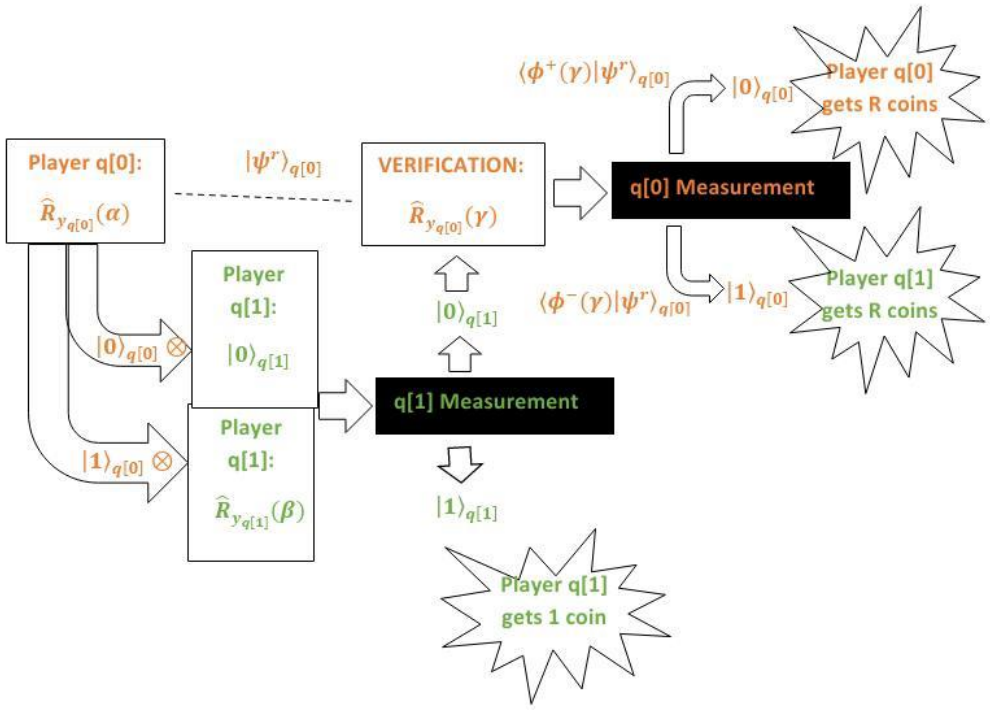


Figure 1. First, the parameter for α is entered on q[0] by one player; then the other enters β on q[1]. Both parameters lead to y-rotations. An entanglement is formed in such a way that the tensor product of $|0\rangle_{q[0]}$ and $|0\rangle_{q[1]}$ form a state, or $|1\rangle_{q[0]}$ and the state of q[1] after its rotation. The player of q[1] gets one coin if it is $|1\rangle_{q[1]}$; otherwise, another y-rotation is applied on q[0] which is now $|\psi^r\rangle_{q[0]}$. The operation is equivalent to projecting $|\psi^r\rangle_{q[0]}$ on $\langle\phi^+(\gamma)|$ or $\langle\phi^-(\gamma)|$. The former gives $|0\rangle_{q[0]}$, resulting in R coin(s) for the player of q[0]; the latter gives $|1\rangle_{q[0]}$ which means that q[1] receives the R coin(s).

International Journal of Applied Sciences and Smart Technologies

Volume 5, Issue 2, pages 213-228

p-ISSN 2655-8564, e-ISSN 2685-9432

Table 1. All possible ways to earn coins for $R > 0$ within the range shown. Notice that if the player of q[0] selects $\alpha = 0$, the average loss per game is minimized (to zero) for player-q[0] (and player-q[1]) no matter what the other player selects; the same is true for the player of q[1] when $\beta = \pi$ with the additional possibility of earning coins if $\alpha \neq 0$. There is no gain improvement for one player when the other sets the corresponding equilibrium parameter. In this way, $\alpha = 0$, $\beta = \pi, \gamma = \pi/2$, is a Nash equilibrium point in the given range. Because it is a zero-sum game, the equivalent table for the player of q[0] is the negative of each of the gains (losses) for the player of q[1].

Table for the average gains per game for the player of q[1].
$R > 0, \gamma = \pi/2$

		q[1]		
		$\frac{\pi}{2} \leq \beta < \pi$	$\beta = \pi$	$\frac{3\pi}{2} \geq \beta > \pi$
	$-\frac{\pi}{2} \leq \alpha < 0$	Depends on $R, \alpha,$	q[1] earns	Depends on R, α, β
q[0]	$\alpha = 0$	zero	zero	zero
	$\frac{\pi}{2} \geq \alpha > 0$	Depends on $R, \alpha,$	q[1] earns	Depends on R, α, β

In general, both players could follow different strategies to increase the likelihood of earning as many coins as possible, not knowing each other's specific settings. The strategy for player-q[0] is not only to diminish the likelihood of $|1\rangle_{q[0]}$, (to make sure the other does not get one coin) but also not to create a state that cannot be verified. For player-q[1], the goal is to "split" the state $q[1]$ enough to increase the likelihood of $|1\rangle_{q[1]}$, but not so much that it allows the other player to verify the remaining state of $q[0]$ if $|1\rangle_{q[1]}$ does not take place. On the other hand, there is Nash equilibrium when $\gamma = \frac{\pi}{2}, \alpha = 0, \beta = \pi$, within the range shown in Table 1. If player-q[0] changes α', either positively or negatively, there is gain for player-q[1] if $\beta = \pi$. If player-q[0] does not change the parameter, but the other does, the game remains zero-gains for both players. Thus, there is no gain improvement for the player that changes the parameter if the other does not.

International Journal of Applied Sciences and Smart Technologies

Volume 5, Issue 2, pages 213-228

p-ISSN 2655-8564, e-ISSN 2685-9432

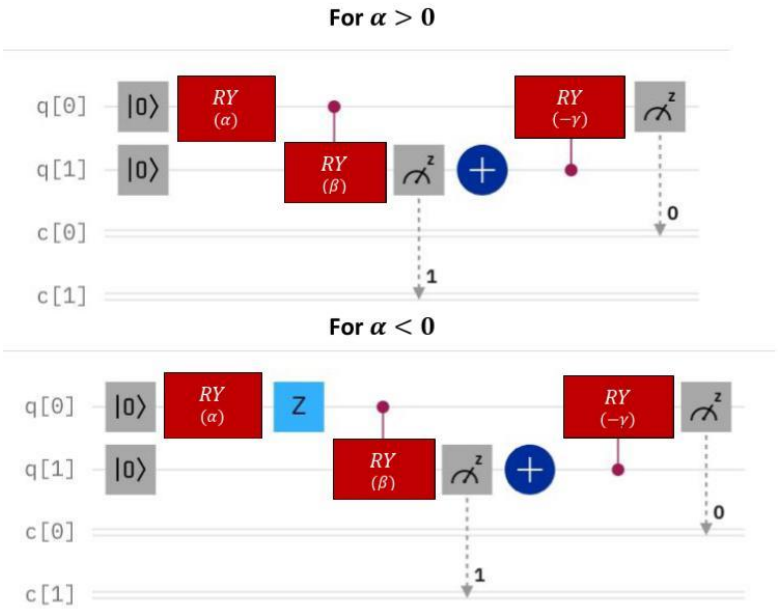


Figure 2. The upper circuit was used for $\alpha \geq 0$; the one below for $\alpha < 0$; in this way, only $|\alpha|$ was used given that $\hat{Z}\hat{R}y(|\alpha|)|0\rangle = \hat{R}y(-|\alpha|)|0\rangle$. The circuits were implemented in the IBM quantum computer "Oslo". For all the data, $\gamma = \pi/2$.

It is important to mention that no actual remote players were used to gather data; however, all data was acquired with an IBM superconducting quantum computer in the cloud. The circuit for the protocol is shown in fig. 2. Each program run determines the hypothetical player that earns coins in one "shot"; however, also thousands of continuous "shots" were obtained in one program run, repeated three times, to calculate the average and standard deviation.

The results coincide with the theoretical Nash equilibrium for $0 < R \ll 1$, and with the theoretical maximum gains (losses) when $R \gg 1$, after thousands of outcomes with a specific set of discrete parameters within the range of Table 1. Nevertheless, based on additional results, such a Nash equilibrium probably could also be confirmed with at least 20 repetitions of each setting, also letting R be a small fraction of a coin. For $R \gg 1$, given thousands of outcomes, the data suggests that if a player maintains $\alpha = 0$ or $\beta =$

Volume 5, Issue 2, pages 213-228
p-ISSN 2655-8564, e-ISSN 2685-9432

π, when the other does not, that player considerably minimizes loss, that is, a player cannot guarantee significant gain improvement by changing the parameter when the other does not, implying Nash-like equilibrium. In this manner, all the data implies that the gains (losses) that would result from the implementation of the protocol in the NISQ device with two remote players, with the specific set of discrete parameters, would be considerably expected functions of those parameters decided by the players.

The protocol presented differs from the cryptographic goal of common quantum coin-flipping protocols presented in the literature [17]. Originally, quantum coin-tossing was conceived as a solution to a "telephone" coin-toss with distrustful parties [18]. Sharing quantum information back and forth between the parties is a solution to the quandary and there have been demonstrations of such [19]. In contrast, the protocol presented in this paper requires a trustful connection to a quantum computer if two played in the cloud. Our protocol is a different paradigm that suggests to use the quantum computer as a true source of entropy as an alternate to other forms of generating multipartite probabilities rather than a secure cryptographic exchange between two parties, although a known cryptographic protocol is being tested.

2 Research Methodology

To initiate a game, two players decide R, such that $R > 0$, and impose a rotation parameter to define a verification state; then, they make concealed y-rotations on their respective qubits and perform measurements to determine the one that earns coins. The matrix representation for the y-rotation is,

$$\hat{R}_{y_{q[0]}}(\gamma) = \begin{pmatrix} \cos\left(\frac{\gamma}{2}\right) & -\sin\left(\frac{\gamma}{2}\right) \\ \sin\left(\frac{\gamma}{2}\right) & \cos\left(\frac{\gamma}{2}\right) \end{pmatrix}, \tag{2}$$

which can be written,

$$\hat{R}_{y_{q[0]}}(-\gamma) = |0\rangle_{q[0]}\langle\phi^{+}(\gamma)|_{q[0]} + |1\rangle_{q[0]}\langle\phi^{-}(\gamma)|_{q[0]} \tag{3}$$

such that

$$\langle\phi^{+}(\gamma)| = \cos\left(\frac{\gamma}{2}\right)\langle 0|_{q[0]} + \sin\left(\frac{\gamma}{2}\right)\langle 1|_{q[0]} \tag{4}$$

in the z-basis $\{|0\rangle, |1\rangle\}$, and

International Journal of Applied Sciences and Smart Technologies

Volume 5, Issue 2, pages 213-228

p-ISSN 2655-8564, e-ISSN 2685-9432

$$\langle \phi^{-}(\gamma)| = -\sin\left(\frac{\gamma}{2}\right)\langle 0|_{q[0]} + \cos\left(\frac{\gamma}{2}\right)\langle 1|_{q[0]}; \quad (5)$$

both eq. (4) and (5) will be projected on the state of $q[0]$ so that its measurement reveals whether it ends up in $\langle \phi^{+}(\gamma)|$ (verification state) or $\langle \phi^{-}(\gamma)|$ (non-verification state). However, before such a projection, Player-q[0] performs a y-rotation of $|0\rangle_{q[0]}$ with angle α, resulting in $|\psi(\alpha)\rangle_{q[0]} = \cos(\frac{\alpha}{2})|0\rangle_{q[0]} + \sin(\frac{\alpha}{2})|1\rangle_{q[0]}$ which is allowed to interact with $|\psi(\beta)\rangle_{q[1]} = \cos(\frac{\beta}{2})|0\rangle_{q[1]} + \sin(\frac{\beta}{2})|1\rangle_{q[1]}$ where player-q[1] decides β after player-q[0]. The entanglement that results from their interaction is

$$\begin{aligned} |E\rangle = \cos\left(\frac{\alpha}{2}\right)|0\rangle_{q[0]} \otimes |0\rangle_{q[1]} + \sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)|1\rangle_{q[0]} \otimes |0\rangle_{q[1]} \\ + \sin\left(\frac{\alpha}{2}\right)\sin\left(\frac{\beta}{2}\right)|1\rangle_{q[0]} \otimes |1\rangle_{q[1]}. \end{aligned} \quad (6)$$

Now, if the measurement on $q[1]$ indicates $|1\rangle_{q[1]}$ then Player-q[1] earns one coin (Player-q[0] loses one). In case $q[1]$ is $|0\rangle_{q[1]}$, the state of q[0] that remains from eq. (6), that is $|\psi^{r}\rangle_{q[0]}$ such that

$$|\psi^{r}\rangle_{q[0]} = N \cdot [\cos\left(\frac{\alpha}{2}\right)|0\rangle_{q[0]} + \sin\left(\frac{\alpha}{2}\right)\cos\left(\frac{\beta}{2}\right)|1\rangle_{q[0]}] \quad (7)$$

where

$$N = \sqrt{\frac{1}{\left(\cos\left(\frac{\alpha}{2}\right)\right)^2 + \left(\sin\left(\frac{\alpha}{2}\right)\right)^2\left(\cos\left(\frac{\beta}{2}\right)\right)^2}}, \quad 8)$$

is projected on the verification (non-verification) state: if $|\psi^{r}\rangle_{q[0]}$ ends up in $\langle \phi^{+}(\gamma)|$ then Player-q[0] earns R coin(s) (Player-q[1] loses R coin(s)); otherwise, Player-q[1] earns them (Player-q[0] loses R coin(s)). Thus, the goal for Player-q[0] is not only that $|\psi(\alpha)\rangle_{q[0]}$ increases the likelihood that $|\psi^{r}\rangle_{q[0]}$ will be verified, but also that $|1\rangle_{q[0]}$ is unlikely. Player-q[1] must make $|\psi(\beta)\rangle_{q[1]}$ such that the one coin can be earned, but not that $|\psi^{r}\rangle_{q[0]}$ can be verified. Both will try to pick values that will minimize the gain of the opponent.

As stated in the previous section, $\gamma' = \frac{\pi}{2}, \alpha' = 0, \beta' = \pi$ is a Nash equilibrium point. If $G_{q[1]}$ and $G_{q[0]}$ are the average gain (or loss) per round of the game for Player-q[1] and Player-q[0] respectively, then

Volume 5, Issue 2, pages 213-228

p-ISSN 2655-8564, e-ISSN 2685-9432

$$G_{q[1]} = -G_{q[0]}; \tag{9}$$

thus, calculating the optimal gain for one of the players implies necessarily the loss for the other. In particular,

$$G_{q[1]} = P_1 + R(P_2 - P_3) \tag{10}$$

where P_1 is the probability that $q[1]$ is in state $|1\rangle_{q[1]}$, P_2 the probability that $|\psi^r\rangle_{q[0]}$ is not verified, and P_3 that it is verified. Eq. (9) can be used to write the expression for $G_{q[0]}$. The probabilities satisfy the condition

$$P_1 + P_2 + P_3 = 1. \tag{11}$$

Consequently, if $q[1]$ is not in state $|1\rangle_{q[1]}$ there is the possibility that $|\psi^r\rangle_{q[0]}$ will be verified, or not, such that

$$P_3 = (1 - P_1) \cdot \left(\langle\phi^+|\psi^r\rangle_{q[0]}\right)^2 \tag{12}$$

or

$$P_2 = (1 - P_1) \cdot \left(\langle\phi^-|\psi^r\rangle_{q[0]}\right)^2. \tag{13}$$

From eq. (10), (11), (12), and (13) follows that

$$G_{q[1]} = P_1 + R(1 - P_1)\left[1 - 2\left(\langle\phi^+|\psi^r\rangle_{q[0]}\right)^2\right]. \tag{14}$$

Explicitly using eq. (4), (7) and (8),

$$\begin{aligned} G_{q[1]}(\alpha,\beta,\gamma) = {} & \left(\sin\left(\frac{\alpha}{2}\right)\right)^2 \left(\sin\left(\frac{\beta}{2}\right)\right)^2 \\ & + R\left[1 - \left(\sin\left(\frac{\alpha}{2}\right)\right)^2 \left(\sin\left(\frac{\beta}{2}\right)\right)^2\right] \\ & \cdot \left\{1 - \frac{2\left(\cos\frac{\gamma}{2}\cos\frac{\alpha}{2} + \sin\frac{\gamma}{2}\sin\frac{\alpha}{2}\cos\frac{\beta}{2}\right)^2}{\left(\cos\frac{\alpha}{2}\right)^2 + \left(\sin\left(\frac{\alpha}{2}\right)\right)^2 \left(\cos\left(\frac{\beta}{2}\right)\right)^2}\right\}. \end{aligned} \tag{15}$$

for

$$\alpha' = 0,\ \beta' = \pi,\ \gamma' = \frac{\pi}{2}, \tag{16}$$

follows that $G_{q[1]}(\alpha',\beta',\gamma') = 0$; changing either α or β in eq. (15) while the other player maintains either β' or α' does not improve the average gain per game for any player, as illustrated in Fig. 3 (consistent with Table 1), when $-\frac{\pi}{2} \le \alpha \le \frac{\pi}{2}$ or $\frac{\pi}{2} \le \beta \le \frac{3\pi}{2}$. Independently of R, (α',β',γ') is a Nash equilibrium point within the range of Fig. 3.

Now, to assess the quantum computer, data was obtained from "Oslo" with two types of specific choices for each qubit using the circuit shown in Fig. 2. In the type-$\pi/3$

International Journal of Applied Sciences and Smart Technologies

Volume 5, Issue 2, pages 213-228

p-ISSN 2655-8564, e-ISSN 2685-9432

games, the choices for the hypothetical player operating $q[0]$ are $\left\{\frac{\pi}{3}, 0, -\frac{\pi}{3}\right\}_\alpha$, and $\left\{\frac{2\pi}{3}, \pi, \frac{4\pi}{3}\right\}_\beta$ for q[1]; for the type-$\pi/2$ games, $\left\{\frac{\pi}{2}, 0, -\frac{\pi}{2}\right\}_\alpha$ and $\left\{\frac{\pi}{2}, \pi, \frac{3\pi}{2}\right\}_\beta$ respectively. The upper qubit in Fig. 2 is q[0], the one below is q[1]. The parameter for the verification was $\gamma = \frac{\pi}{2}$ (on the right of fig. 2). One "shot" was obtained for each setting combination, but the process was repeated twenty times. The possible output after each repetition was $|1\rangle_{q[0]} \otimes |1\rangle_{q[1]}$, $|1\rangle_{q[0]} \otimes |0\rangle_{q[1]}$, or $|0\rangle_{q[0]} \otimes |0\rangle_{q[1]}$; respectively, each outcome was used to calculate P_1, P_2, and P_3 in eq. (10), that is, their frequencies divided by twenty. The results were compared to eq. (15). The probability for the "erroneous" state $|0\rangle_{q[0]} \otimes |1\rangle_{q[1]}$ was calculated. In addition, 1000 "shots" for each of the setting combinations were performed in one program run, repeated 3 times, to obtain an average and the standard deviation.

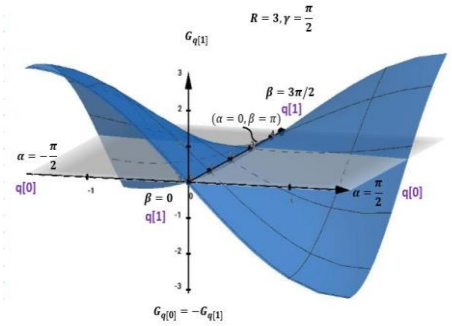


Figure 3. An illustration of Nash equilibrium for $R = 3$: if the player of q[0] moves through the equilibrium point $(\alpha = 0, \beta = \pi)$, that is, on the gray-plane and parallel to the α-axis, then there is positive gain for the player of q[1]. Doing the same along the β-axis, does not change the gain or loss for any of the players. Thus, it is a point where there is no improvement in gain for a player when the other keeps the equilibrium parameter constant. The four quadrants surrounding the equilibrium point show that, without knowing the parameter from the other, gains can be either positive or negative.

3 Results and Discussions

Table 2,3 and 4 below summarize all the results. The first two present the measured gains (losses) for the two types of games. The last one presents theoretical calculations and the "erroneous" state probability.

International Journal of Applied Sciences and Smart Technologies

Volume 5, Issue 2, pages 213-228

p-ISSN 2655-8564, e-ISSN 2685-9432

Table 2. $G_{q[1]}(\alpha,\beta,\gamma) = P_1 + R(P_2 - P_3)$, where P_1, P_2, and P_3 are the average probabilities obtained from the circuit in fig. 2 with three repetitions of 1000 "shots" each one. The standard deviations of each average were used to estimate the measurement errors. In this way, those thousands of "shots" were obtained continuously in one program run rather than one in each program run as it would be in a game.

The average of three program runs of 1000 shots in each one for $\gamma' = \pi/2$ (Type-$\frac{\pi}{3}$ game)

	$\beta = 2\pi/3$	$\beta = \pi$	$\beta = 4\pi/3$
$\alpha = -\frac{\pi}{3}$	$G_{q[1]}(-\frac{\pi}{3},\frac{2\pi}{3},\gamma') = (0.362 \pm 0.027)(R) + (0.178 \pm 0.016)$	$G_{q[1]}(-\frac{\pi}{3},\pi,\gamma') = (-0.011 \pm 0.027)(R) + (0.215 \pm 0.007)$	$G_{q[1]}(-\frac{\pi}{3},\frac{4\pi}{3},\gamma') = (-0.379 \pm 0.036)(R) + (0.182 \pm 0.009)$
$\alpha = 0$	$G_{q[1]}(0,\frac{2\pi}{3},\gamma') = (0.011 \pm 0.020)(R) + (0.015 \pm 0.004)$	$G_{q[1]}(0,\pi,\gamma') = (-0.021 \pm 0.020)(R) + (0.023 \pm 0.001)$	$G_{q[1]}(0,\frac{4\pi}{3},\gamma') = (-0.049 \pm 0.039)(R) + (0.014 \pm 0.004)$
$\alpha = \frac{\pi}{3}$	$G_{q[1]}(\frac{\pi}{3},\frac{2\pi}{3},\gamma') = (-0.392 \pm 0.031)(R) + (0.165 \pm 0.008)$	$G_{q[1]}(\frac{\pi}{3},\pi,\gamma') = (-0.026 \pm 0.021)(R) + (0.241 \pm 0.021)$	$G_{q[1]}(\frac{\pi}{3},\frac{4\pi}{3},\gamma') = (0.307 \pm 0.016)(R) + (0.191 \pm 0.002)$

The average of three program runs of 1000 "shots" in each one for $\gamma' = \frac{\pi}{2}$ (Type-$\frac{\pi}{2}$ game)

	$\beta = \pi/2$	$\beta = \pi$	$\beta = 3\pi/2$
$\alpha = -\frac{\pi}{2}$	$G_{q[1]}(-\frac{\pi}{2},\frac{\pi}{2},\gamma') = (0.588 \pm 0.017)(R) + (0.238 \pm 0.013)$	$G_{q[1]}(-\frac{\pi}{2},\pi,\gamma') = (-0.012 \pm 0.014)(R) + (0.469 \pm 0.014)$	$G_{q[1]}(-\frac{\pi}{2},\frac{3\pi}{2},\gamma') = (-0.627 \pm 0.015)(R) + (0.239 \pm 0.008)$
$\alpha = 0$	$G_{q[1]}(0,\frac{\pi}{2},\gamma') = (-0.035 \pm 0.019)(R) + (0.015 \pm 0.002)$	$G_{q[1]}(0,\pi,\gamma') = (-0.021 \pm 0.020)(R) + (0.023 \pm 0.001)$	$G_{q[1]}(0,\frac{3\pi}{2},\gamma') = (-0.063 \pm 0.025)(R) + (0.012 \pm 0.003)$
$\alpha = \frac{\pi}{2}$	$G_{q[1]}(\frac{\pi}{2},\frac{\pi}{2},\gamma') = (-0.663 \pm 0.020)(R) + (0.220 \pm 0.010)$	$G_{q[1]}(\frac{\pi}{2},\pi,\gamma') = (-0.032 \pm 0.026)(R) + (0.460 \pm 0.006)$	$G_{q[1]}(\frac{\pi}{2},\frac{3\pi}{2},\gamma') = (0.589 \pm 0.011)(R) + (0.245 \pm 0.008)$

International Journal of Applied Sciences and Smart Technologies

Volume 5, Issue 2, pages 213-228

p-ISSN 2655-8564, e-ISSN 2685-9432

Table 3. $G_{q[1]}(\alpha,\beta,\gamma) = P_1 + R(P_2 - P_3)$, where P_1, P_2, and P_3 are respectively the frequencies of $|1\rangle_{q[0]} \otimes |1\rangle_{q[1]}$, $|1\rangle_{q[0]} \otimes |0\rangle_{q[1]}$, and $|0\rangle_{q[0]} \otimes |0\rangle_{q[1]}$ divided by 20. Data was obtained from the circuit in fig. 2 with one "shot" for each run of the program but repeated 20 times. This is how the outcomes for an actual game are obtained.

$G_{q[1]}$ for 20 program runs of one "shot" in each one for $\gamma' = \frac{\pi}{2}$(Type-$\frac{\pi}{3}$ game)

	$\beta = 2\pi/3$	$\beta = \pi$	$\beta = 4\pi/3$
$\alpha = -\frac{\pi}{3}$	$G_{q[1]}(-\frac{\pi}{3},\frac{2\pi}{3},\gamma') = 0.4(R) + 0.15$	$G_{q[1]}(-\frac{\pi}{3},\pi,\gamma') = -0.1(R) + 0.4$	$G_{q[1]}(-\frac{\pi}{3},\frac{4\pi}{3},\gamma') = -0.45(R) + 0.5$
$\alpha = 0$	$G_{q[1]}(0,\frac{2\pi}{3},\gamma') = -0.1(R)$	$G_{q[1]}(0,\pi,\gamma') = 0.05(R) + 0.05$	$G_{q[1]}(0,\frac{4\pi}{3},\gamma') = 0$
$\alpha = \frac{\pi}{3}$	$G_{q[1]}(\frac{\pi}{3},\frac{2\pi}{3},\gamma') = -0.45(R) + 0.25$	$G_{q[1]}(\frac{\pi}{3},\pi,\gamma') = 0.25$	$G_{q[1]}(\frac{\pi}{3},\frac{4\pi}{3},\gamma') = -0.10(R) + 0.25$

$G_{q[1]}$ for 20 program runs of one "shot" in each one for $\gamma' = \pi/2$

(Type-$\frac{\pi}{2}$ game)

	$\beta = \pi/2$	$\beta = \pi$	$\beta = 3\pi/2$
$\alpha = -\frac{\pi}{2}$	$G_{q[1]}(-\frac{\pi}{2},\frac{\pi}{2},\gamma') = 0.75(R) + 0.25$	$G_{q[1]}(-\frac{\pi}{2},\pi,\gamma') = 0.30$	$G_{q[1]}(-\frac{\pi}{2},\frac{3\pi}{2},\gamma') = -0.65(R) + 0.3$
$\alpha = 0$	$G_{q[1]}(0,\frac{\pi}{2},\gamma') = -0.1(R)$	$G_{q[1]}(0,\pi,\gamma') = 0.05(R) + 0.05$	$G_{q[1]}(0,\frac{3\pi}{2},\gamma') = -0.25(R)$
$\alpha = \frac{\pi}{2}$	$G_{q[1]}(\frac{\pi}{2},\frac{\pi}{2},\gamma') = -0.35(R) + 0.35$	$G_{q[1]}(\frac{\pi}{2},\pi,\gamma') = -0.15(R) + 0.40$	$G_{q[1]}(\frac{\pi}{2},\frac{3\pi}{2},\gamma') = 0.45(R) + 0.25$

International Journal of Applied Sciences and Smart Technologies

Volume 5, Issue 2, pages 213-228

p-ISSN 2655-8564, e-ISSN 2685-9432

Table 4. Measured average gain per game for the player-q[1] using the circuit in figure 2, and also calculated using eq. (15). $G_{q[1]} = P_1 + R(P_2 - P_3)$ where P_1 is the probability to obtain $|1\rangle_{q[0]} \otimes |1\rangle_{q[1]}$, P_2 and P_3 are those that correspond to $|1\rangle_{q[0]} \otimes |0\rangle_{q[1]}$ and $|0\rangle_{q[0]} \otimes |0\rangle_{q[1]}$ respectively. Also, the probability of the "erroneous" state $|0\rangle_{q[0]} \otimes |1\rangle_{q[1]}$ was calculated for the 20 program runs of one "shot" in each one (left column) and the average of 3 program runs of 1000 "shots" in each one (right column).

Measured $G_{q[1]}$ vs. theoretical $G_{q[1]}$ for $\gamma' = \pi/2$

Type- $\pi/3$				
Settings $(q[0], q[1])$	**From "Oslo", 20 program runs of one "shot" in each one** $G_{q[1]} =$	**Theoretical** $G_{q[1]} =$	**Prob. Of** $\|0\rangle_{q[0]} \otimes \|1\rangle_{q[1]}$ (Not used in Table 3)	**Prob. Of** $\|0\rangle_{q[0]} \otimes \|1\rangle_{q[1]}$ (Not used in Table 2)
$(\pi/3,2\pi/3)$	$-0.45(R) + .25$	$-0.4331(R) + .1875$	0.00	0.038 ± 0.003
$(\pi/3,4\pi/3)$	$-0.10(R) + .25$	$+0.4331(R) + .1875$	0.05	0.026 ± 0.007
$(-\pi/3, 2\pi/3)$	$+0.4(R) + 0.15$	$+0.4331(R) + .1875$	0.05	0.033 ± 0.007
$(-\pi/3,4\pi/3)$	$-0.45R + 0.05$	$-0.4331(R) + .1875$	0.00	0.036 ± 0.002
$(0, 4\pi/3)$	0	0	0.00	0.030 ± 0.003
$(\pi/3, \pi)$	.25	.25	0.05	0.030 ± 0.010
$(0, 2\pi/3)$	$-0.10(R)$	0	0.00	0.025 ± 0.002
$(-\pi/3, \pi)$	$-0.10(R) + 0.4$	.25	0.00	0.038 ± 0.002

Type- $\pi/2$				
Settings $(q[0], q[1])$	**From "Oslo", 20 program runs of one "shot" in each one,** $G_{q[1]} =$	**Theoretical** $G_{q[1]} =$	**Prob. Of** $\|0\rangle_{q[0]} \otimes \|1\rangle_{q[1]}$ (Not used in Table 3)	**Prob. Of** $\|0\rangle_{q[0]} \otimes \|1\rangle_{q[1]}$ (Not used in Table 2)
$(-\pi/2,3\pi/2)$	$-0.65(R) + .30$	$-0.7072(R) + 0.25$	0.05	0.029 ± 0.002
$(-\pi/2, \pi/2)$	$+0.75(R) + .25$	$+.7072(R) + 0.25$	0.00	0.026 ± 0.006
$(\pi/2, 3\pi/2)$	$+0.45(R) + 0.25$	$+0.7072(R) + 0.25$	0.00	0.029 ± 0.003
$(\pi/2, \pi/2)$	$-0.35R + 0.35$	$-0.7072(R) + 0.25$	0.00	0.029 ± 0.010
$(0, 3\pi/2)$	$-0.25(R)$	0	0.05	0.040 ± 0.007
$(\pi/2, \pi)$	$-0.15(R) + 0.40$	0.5	0.05	0.054 ± 0.012
$(0, \pi/2)$	$-0.10(R)$	0	0.00	0.030 ± 0.005
$(-\pi/2, \pi)$	0.3	0.5	0.10	0.048 ± 0.009
N.E.				
$(0, \pi)$	$+0.05(R) + 0.05$	0	0.00	0.034 ± 0.007

International Journal of Applied Sciences and Smart Technologies

Volume 5, Issue 2, pages 213-228

p-ISSN 2655-8564, e-ISSN 2685-9432

The results for the two types of games with one "shot" per program run, as it would be in an actual game, repeated 20 times, show that the Nash equilibrium point coincides with theory for a limited range of R. The point $(\alpha' = 0, \beta' = \pi, \gamma' = \frac{\pi}{2})$ at the centers of Type-$\pi/2$ and Type-$\pi/3$ data in Table 3 become the Nash equilibrium shown in Table 1 if

(i) a. $G_{q[1]}\left(-\frac{\pi}{3}, \pi, \gamma'\right)$ & $G_{q[1]}\left(\frac{\pi}{3}, \pi, \gamma'\right) \geq G_{q[1]}(0, \pi, \gamma')$,

b. $G_{q[1]}\left(-\frac{\pi}{2}, \pi, \gamma'\right)$ & $G_{q[1]}(\frac{\pi}{2}, \pi, \gamma') \geq G_{q[1]}(0, \pi, \gamma')$,

(ii) a. $G_{q[1]}(0, \pi, \gamma') \geq G_{q[1]}\left(0, \frac{2\pi}{3}, \gamma'\right)$ & $G_{q[1]}\left(0, \frac{4\pi}{3}, \gamma'\right)$,

b. $G_{q[1]}(0, \pi, \gamma') \geq G_{q[1]}\left(0, \frac{\pi}{2}, \gamma'\right)$ & $G_{q[1]}(0, \frac{3\pi}{2}, \gamma')$.

Given the data in Table 3, if $R > 0$ in the inequalities (i) & (ii), then $(0, \pi. \gamma')$ is a Nash equilibrium point when $0 \leq R \leq 2.3$, for Type-$\pi/3$ games, and $0 \leq R \leq 1.75$ for Type-$\pi/2$. In theory, $(0, \pi. \gamma')$ is a Nash equilibrium point without restrictions in R. Eq. (15) implies that the coefficients of R, in the center rows and columns corresponding to the two types of games in Table 3, are zero (theoretical gains are shown in Table 4); however, this is not the case in Table 3. The coefficients of R (as well as the constant terms) have variations which imply an even narrower range to confirm the theoretical Nash equilibrium point reliably. On the other hand, R is selected by the players. The closer they select R to zero the more likely that they can verify the Nash equilibrium point with a few games (assuming the error in the constant term does not fluctuate considerably when playing a small number of games). Diminishing R necessarily makes its coefficient less significant in the center rows and columns as expected in theory.

Table 2 also shows that R can be a small fraction to confirm that $(0, \pi. \gamma')$ is the Nash equilibrium point. Considering all the measurement errors and the inequalities (i) & (ii), the point is reliably the equilibrium after thousands of outcomes for $0 < R \leq 0.041$ in type-$\pi/3$ games. In type-$\pi/2$ games, the same can be concluded for $0 < R \leq 0.20$. These are the ranges implied by the most extreme measurement error fluctuations possible. In this way, these can be the ranges from which players select R from the start, even for a few games, if players seek a theoretical Nash equilibrium.

Volume 5, Issue 2, pages 213-228
p-ISSN 2655-8564, e-ISSN 2685-9432

On the other hand, for $R \gg 1$, the gains (losses) in the corners of the matrices corresponding to the two types of games in Table 2 are much greater than those in the center rows and columns as predicted by theory. If $\left|\Delta G_{q[1]}^{C}\right|$ is the absolute change in gain (loss) from a non-corner one to a corner one, and $\left|\Delta G_{q[1]}^{N}\right|$ is the absolute change from one that is a non-corner one to another non-corner one, then $\left|\Delta G_{q[1]}^{C}\right| \gg \left|\Delta G_{q[1]}^{N}\right|$ for all data in Table 2 for $R \gg 1$. In other words, in this R range, the gains (losses) when at least one player maintains the equilibrium parameter, regardless of the other's selection, are notably less than those corresponding to the other parameter combinations given thousands of "shots". Considering the measurement errors in Table 2, there may not be significant gain improvement for a player changing the equilibrium parameter when the other maintains it. Consequently, in this case, a player does not have a strong incentive to change the parameter when the other does not change it; a player considerably minimizes the losses (gains) by keeping it constant regardless of the other's selection. $(0, \pi. \gamma')$ is a Nash-like equilibrium point for $R \gg 1$ given a large number of games.

4 Conclusions

The results suggest that two players can confirm reliably that $(0, \pi, \gamma')$ is a Nash equilibrium point for two qubits of the game protocol shown in Fig. 1, with the circuits in Fig. 2, when R is a small fraction of a coin, testing thousands of times each of the setting combinations from either type-$\pi/3$ or type-$\pi/2$ games; also, the data shows that it is probable that the same could be confirmed with at least 20 repetitions. Now, consistent with theory, for $R \gg 1$, given thousands of "shots", the notably greatest gains (losses) correspond to those when both players do not set their equilibrium parameters, which means that a player can considerably minimize the losses (gains) by not changing it. In this case, there may not be significant gain improvement for the one that changes the equilibrium parameter if the other does not; consequently, there is no strong incentive for a player to change it when the other is maintaining it, suggesting Nash-like equilibrium. Thus, under the same restrictions for R after more than 20 games with the same set of setting combinations, the gains (losses) would be expected functions of participants' choices if two played remotely in the cloud. In the future, as NISQ devices

Volume 5, Issue 2, pages 213-228
p-ISSN 2655-8564, e-ISSN 2685-9432

improve, it is likely that a Nash-equilibrium point can be attained with less restrictions if two played the protocol introduced.

References

[1] Zhang P., Zhou XQ, Wang YL, et al. Quantum gambling based on Nash-equilibrium. npj Quantum inf. 3 (24) (2017).

[2] Lu Zhou, Xin Sun, Chunhua Su, Zhe Liu, Kim-Kwang, Raymond Chou, Game theoretic security of quantum bit commitment, Information Science 479 (2019).

[3] Daniel Centeno, German Sierra, General quantum chinos game, 6 (7).

[4] Yuyang Han, Cheat-sensitive coin flipping and quantum gambling, Quantum information Processing 21(170) (2022).

[5] Luyao Wang, Hai Cheng, Pseudo-Random number generator based on logistic chaotic system, Entropy 21(10), 960 (2019).

[6] Fei Yu, Lixiang Li, Qiang Tang, Shuo Cai, Yun Song, Quan Xu, A survey on true random number generator base on chaos, discrete Dynamics in nature and society, (2019).

[7] U. Ansari, A. K. Chaudhary and S. Verma, True Random Number Generator (TRNG) Using Sensors for low Cost IoT Applications, 2022 International Conference on Communication, Computing and Internet of Things (IC3IoT), Chennai, India, (2022) 1-6.

[8] Christina Chamon, Shahriar Ferdous, and Laszlo B. Kish, Deterministic random number generator attack against the kirchhoff-law-Johnson-Noise secure key exchange protocol, Fluctuation and Noise Letters, 20(5) (2021) 2150046.

[9] Yutaka Shikano, Unpredictable random number generator, AIP Conference Proceedings 2286 (2020) 040004.

[10] Miguel Herrero-Collantes, Juan Carlos Garcia-Escartin, Quantum random number generators, Rev. Mod. Phys. 89 (2017) 015004.

[11] Jaideep Patnak, Brian Hunt, Michelle Girvan, Zhixin Lu, Edward Ott, Model-free prediction of Large spatiotemporally chaotic system from Data: A reservoir computing approach. Phys. Rev. Lett. 120 (2018) 024102

[12] John Preskill, Quantum Computing in the NISQ era and beyond, Quantum 2, 79 (2018)

[13] McEwen, M., Faoro, L., Arya, K. et al. Resolving catastrophic error bursts from cosmic rays in large arrays of superconducting qubits. Nat Phys. 18 (2022) 107-111.

[14] Lior Goldenberg, Lev Veidman, and Stephen Wiesner, Quantum gambling, Phys. Rev. Lett. 82 (1999) 3356.

[15] Ireneuz Pokula, Quantum gambling using mesoscopic ring qubits Physica Status Solidi (b) Vol. 244(7) (2007) 2513-2515.

[16] Davide Castelvecchi, IBM's quantum cloud computer goes commercial, Nature 543 (7644) (2017).

[17] Charles H. bennett, Gilles Brassard, Quantum cryptopgraphy: Public Key distribution and coin tossing, Theoretical computer Science Volume 560, Part 1, 4 (2014).

[18] Blum, Manuel Coin flipping by telephone a protocol for solving impossible problems. ACM SIGACT News 15(1) 23-27.

[19] Anna Pappa, Paul Jouguet, Thomas Lawson, Andre Chailloux, Matthieu Lagre, Patrick Trinkler, Lordanis Kernidis, Eleni Diamanti, Experimental plug and play quantum coin flipping, Nature Communications 5(3717) (2014).