Examination of Quantum Pseudo-Telepathy Games and Implementation in Qiskit

Jincheng Zhang Team Duality

Columbia University jz2918@columbia.edu

Abstract. In my project, general quantum games, especially quantum pseudo-telepathy games are examined. Parity Game and Magic Square Game are discussed, where classical strategy and quantum strategy are compared. As a special case of Parity Game, the Mermin GHZ Game is implementated in Qiskit and run on IBM Q Experience backend(3 out of 5 qubits). Execution results both on local simulator and online backend of the Mermin GHZ Games are analyzed. Conclusion is mainly that in theory and running on local simulator quantum strategy has a guaranteed winrate while on real quantum machines, due to noise, statistically quantum strategy has a worse winrate compared to the classical one so far.

Keywords: Quantum Computing · Quantum Teleportation · Pseudo-Telepathy · Entanglement · GHZ state

1 Introduction

1.1 Project Background

Because of the quantum mechanics superposition of qubits, Qubit register can hold much more information compared to a classical bit register. In the early experiments and implementations of quantum games, according to the IBM Q Experience, a battleship game is built to run on a 5-qubit machine, each qubit to store a health state of a battleship. Since one battleship health power (HP) can be up to 3, a classical bit state (containing 0 or 1) cannot be enough to hold. But operating on a qubit and with Partial NOT gates rotating along the X axis, the state holding can be achieved and post multiple trials of state measurement yield the correct retrieval of the HP state. Now in each qubit, measurement of 0 means healthy and 1 means destroyed, by rotating the state in between them gives the power to hold all the possible HPs. The quantum games like the battleship utilizes the property of a single qubit and therefore provide some good educational quantum programming tutorials. While for demonstration of how entanglement works, we have the bell state encoding and quantum teleporation. While later a certain class of entanglement (not only on just two qubits) named quantum pseudo-telepathy games provides more comprehensive educational implementation of entanglement on multiple qubits. More specifically, for example, the Mermin GHZ Game provides a well done demonstration of the three qubit entanglement, the GHZ state. In the following section general quantum pseudo-telepathy games are discussed.

1.2 Quantum Pseudo-Telepathy Games

For simplicity, consider two parties(Alice and Bob) playing a Pseudo-Telepathy game. During the pre-game stage, Alice and Bob have the chance to exchange any information: for example it can be some classical bits information or quantum entangled states. Once the process is done they can't exchange any more information. The next stage is the input stage: Alice and Bob are separately receiving a sequence of inputs. Based on those inputs and previously shared states, each player makes some outputs and if those outputs always meet the winning condition, we say the players can somehow "communicate" with each other, which is not the case in reality, so we call it Pseudo-Telepathy. Below is a formal definition of two player Pseudo-Telepathy game.

Definition A two-party game is defined as a sextuple $G = \langle X, Y, A, B, P, W \rangle$ where X, Y, A and B are sets, $P \subseteq X \times Y$ and $W \subseteq X \times Y \times A \times B$. X and Y are input sets, A and B are output sets, P is predicate on $X \times Y$ as promise and W is the winning condition.

2 The Parity Game

2.1 Overview

This is a family of games for n players, $n \geq 3$, with the property that the player's outputs are single bits, and the winning condition depends on their parity.

2.2 Game settings

Input For each player i: a bit-string $x_i \in \{0,1\}^l$ (also binary string, l denotes length)

Input Constraints

$$\sum_{i=1}^{n} x_i \mid 2^l$$

Output For each player i: a single bit a_i

Winning Conditions

$$\sum_{i=1}^{n} a_i \equiv \frac{\Sigma_{i=1}^{n} x_i}{2^l} \mod 2$$

When $n = 3$ and $l = 1$, this is what is known as Mermin-GHZ game.
When $l = 1$ and $n \geq 3$ be arbitary, this is known as Mermin's parity game.

2.3 Classical Strategy

To prove that classical strategy cannot achieve a 100% winrate deterministicly, we take the Mermin-GHZ game as the special case of parity game, which is the easiest for illustration.

In the Mermin-GHZ game, consider three players called Alice, Bob and Charlie, respectively. They are each receiving a bit, named x, y, and z. Their outputs (based on inputs correspondingly) are denoted as a_x, b_y, and c_z.

According to the input constraints, $x + y + z$ should divide 2, which means the combinations of x, y, and z can only be $000, 011, 101, 110$. Now let us look at the winning condition. $l = 1$ so that for the 000 case $\Sigma_{i=1}^{n} a_i$ should divide 2 (labeled even for later notation, also in my code) and for other cases $\Sigma_{i=1}^{n} a_i$ should yield 1 modulo 2 (odd).

This can be summarized in the equations below:

$$a_0 + b_0 + c_0 \equiv 0 \mod 2$$

$$a_0 + b_1 + c_1 \equiv 0 \mod 2$$

$$a_1 + b_0 + c_1 \equiv 0 \mod 2$$

$$a_1 + b_1 + c_0 \equiv 0 \mod 2$$

Adding the four equations together we can see on the left hand side, the terms of the same name all come in pairs: two of a_0, two of b_1, ... etc.
While on the right hand side it becomes $3 \mod 2 \Rightarrow 1 \mod 2$, which results in a contradiction.
Therefore we can conclude that with no sharing information and player just respond deterministicly, the classical strategy cannot always lead to a win.

2.4 Quantum Strategy

This strategy is general for any l and $n \geq 3$.
During preparation(pre-game), the players first share entangled state

$$\frac{1}{\sqrt{2}}|0^n\rangle + \frac{1}{\sqrt{2}}|1^n\rangle$$

Then apply to his or her entangled state the unitary transformation (first i is for $\sqrt{-1}$):

$$|0\rangle \rightarrow |0\rangle$$
$$|1\rangle \rightarrow e^{\pi i x_i}|1\rangle$$

Then apply Hadamard transform, defined by:

$$|0\rangle \rightarrow \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$
$$|1\rangle \rightarrow \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle$$

Finally measure the qubit in the computational basis to obtain a_i and output a_i

2.5 Implementation in Qiskit

To implement the quantum winning strategy, we first observe that for n players they share an entangled state in total:

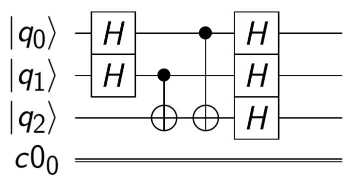
$$\frac{1}{\sqrt{2}}|0^n\rangle + \frac{1}{\sqrt{2}}|1^n\rangle$$

So first step, we need to prepare this entangled state.

Since currently only ibmqx2 or ibmqx4 are usable, we only have 5 qubits in maximum to manipulate. So we can only implement the case when $3 \leq n \leq 5$, which means implementing the Mermin-GHZ Game provides the similarity to those n cases. So here we just implement the Mermin-GHZ game.
So in the beginning we need to get the entangled GHZ state:

$$\frac{1}{\sqrt{2}}|000\rangle + \frac{1}{\sqrt{2}}|111\rangle$$

The circuit for this GHZ state is just an additional H and CNOT gate compared to the Bell state preparation (first entangle a pair then entangle another):



After the entangled pair we put a barrier (to avoid optimize two consecutive Hs) and then we do the phase shift: $|0\rangle \rightarrow |0\rangle$, $|1\rangle \rightarrow e^{\pi i x_i}|1\rangle$. For Mermin-GHZ game specifically, x_i can only be 0 or 1. So here comes the trick:
For quantum gates we have the general phase shift gate:

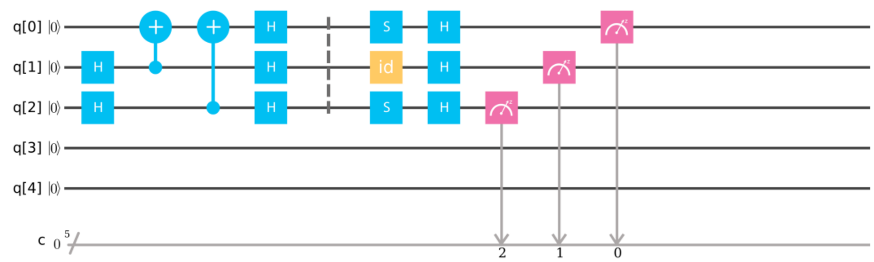
$$R_\phi = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\phi} \end{bmatrix}$$

In our case when $x_i = 0, e^{i\phi} = 1$, it is just the identity transform. When $x_i = 1, \phi = \pi/2$ which can written as U1($\pi/2$) or $\sqrt{S}$ Gate in OPENQASM 2.0.
Verified by looking up in the underlying implementation of the S Gate:

```
// 1-parameter 0-pulse single qubit gate
gate u1(lambda) q { U(0,0,lambda) q; }
// Clifford gate: sqrt(Z) phase gate
gate s a { u1(pi/2) a; }
```

So the core encoding is done, and finally apply H Gate back will get us the correct output.
The whole circuit is (Shown as the 101 case: $x_0 = 1, x_1 = 0, x_2 = 1$ and we can

see the position of the *id* and S Gate as follows):



2.6 Execution Results

This code is available in the appended python notebook. Simulator post selection is done and yields 100% correct results(see output in the notebook). This circuit is also tested on the IBM Q Experience backend 'ibmqx4'. With 2 rounds, each with 1024 shots, results turn out to be as the following: Results come with three classical bit output, to verify we count Σa_i and see if it is even or odd. Fidelity is therefore calculated.

Table 1. Execution Results

Round	Total Shots	000	001	010	011	100	101	110	111	Even total	Odd total	Fidelity
1	1024	106	180	208	20	258	51	41	160	218	806	78.71%
1	1024	113	223	166	37	223	57	49	156	256	768	75.00%

From the results we can see due to noise, using this quantum computing device will yield lower winrate statistically than the randomized classical strategy.

3 The Magic Square Game

This game is only examined, not been proved and implemented since it has a full implementation online at Qiskit Tutorial.

3.1 Game settings

A *magic square* is a 3×3 matrix with entries $\{0, 1\}$ or denoted as $\{+, -\}$
Alice is asked to give the entries of a row $x \in \{1, 2, 3\}$ (a input)

Bob is asked to give the entries of a column $y \in \{1, 2, 3\}$ (b input)
Winning condition: parity of the row must be even, the parity of the column must be odd, and the intersection of the given row and column must agree.
There are nine possible questions (row by column), which are all need to be legitimate.

3.2 Classical Strategy

To prove classical strategy cannot yield 100% winrate. We take the example:
If deterministically reserve square as:

1	1	0
0	1	1
0	1	?

When Alice receives 2 and Bob receives 3 ($a = 1$ and $b = 2$): the second row is satisfied and third column is also satisfied, but on $a = 3$ and $b = 3$ the ? field cannot be agreed at once. The observation shows this all 9 possible combinations can't be fulfilled at the same time.

3.3 Quantum Strategy

Alice: a, c Bob: b, d two shared entangled pair encoded as

$$|\phi\rangle = \frac{1}{\sqrt{2}}(|+\rangle_a \otimes |+\rangle_b + |-\rangle_a \otimes |-\rangle_b) \otimes \frac{1}{\sqrt{2}}(|+\rangle_c \otimes |+\rangle_d + |-\rangle_c \otimes |-\rangle_d)$$

which is

$$|\psi\rangle = \frac{1}{2}|0011\rangle - \frac{1}{2}|0110\rangle - \frac{1}{2}|1001\rangle + \frac{1}{2}|1100\rangle$$

Then the output can be:

$+X \otimes I$	$+X \otimes X$	$+I \otimes X$
$-X \otimes Z$	$+Y \otimes Y$	$-Z \otimes X$
$+I \otimes Z$	$+Z \otimes Z$	$+Z \otimes I$

4 Conclusion

The magic square game is already implemented, posted on the qiskit tutorial. The Mermin GHZ game also seems to be a very good tutorial for implementation of demonstration on the GHZ state. And statistics show that for current quantum machines backend, it only achieves 75%80% winrate. But using a local simulator, we still see the potential of pseudo-telepathy showing the nonlocality of quantum mechanics.

References

1. Quantum Pseudo-Telepathy https://arxiv.org/abs/quant-ph/0407221v3
2. Quantum Pseudo-Telepathy Wikipedia https://en.wikipedia.org/wiki/Quantum_pseudo-telepathy
3. Quantum Pseudo-Telepathy Qiskit Tutorial https://github.com/QISKit/qiskit-tutorial/blob/master/appendix/etc/quantum_magic_square.ipynb
4. IBM Q Experience https://quantumexperience.ng.bluemix.net/
5. Qiskit API Reference https://qiskit.org/documentation/_autodoc/qiskit.QuantumCircuit.html
6. Open Quantum Assembly Language https://arxiv.org/pdf/1707.03429.pdf
7. GHZ State Tutorial https://quantumexperience.ng.bluemix.net/proxy/tutorial/full-user-guide/003-Multiple_Qubits_Gates_and_Entangled_States/060-GHZ_States.html