Quantum Games

Jiarui Liu

1. Introduction

Nowadays Quantum mechanics has been applied to many fields to solve complex problems. The studies on quantum information have succeeded in such as quantum computation, quantum circuit, quantum cryptography, quantum communication complexity, and so on. Quantum game theory is part of quantum study which is used to make decision on a specific game to maximum the probability to win based the information and operation on quantum computing.

In quantum games, however, players employ quantum strategies instead of mixed strategies, over a larger set of quantum states ρ in Hilbert space H. It has been shown that quantum strategies lead to advantages over classical strategies in some particular game examples.

Three games are included in our study, each of them can be optimized to maximum the winning probability.

2. Games and Strategies

2.1 Single coin flip game

Alice plays a game against Computer to flip a coin, each move Alice or computer can choose to flip the coin or not. Initially the coin is prepared in state with “head” on the top.

The game follows the process:

1. Quantum Computer plays a move, but it is not revealed to the Opponent [Human].
2. Alice plays a move and it is also not revealed to the Quantum Computer.
3. Finally, Quantum Computer plays a move.
4. Computer wins if the coin is in state with “head” on the top.

Strategy:

1. The qubit is initialized to the $|0\rangle$ "heads" state.
2. The computer plays, applying the H Gate to the qubit. The coin enters the $|+\rangle$ state.
3. Alice plays, no matter whether she choose to flip (Which means apply X Gate to qubit) or not, the qubit will stay $|+\rangle$ state.
4. The computer plays, applying the H Gate to the coin.
5. The final state will always be $|0\rangle$, Computer wins.

Implement in Q#:

```
operation ComputerPly (q : Qubit) : Unit{
    H(q);
}
operation peoplePlay (q : Qubit) : Unit{
    let flip = RandomInt(2);
    if(flip == 1){
        X(q);
    }
}
```

2.2 Two coins flip game

Alice prepares 2 coins and puts them into a box in the state of all the coins being heads, i.e., (H, H). Suppose that any player cannot see the inside of the box.

1. Bob flips some coins (or not)
2. Alice flips m coins(0<=m<=2) under keeping the value of m secret from Bob.
3. Finally, Bob flips some coins (or not).

Open the box, Bob wins if the coins are in the following state: the state of the coins is (H, H) if is even, or the state of the coins is (T, H) if m is odd. Otherwise Alice wins.

Strategy:

1. Bob prepares the $|00\rangle+|11\rangle$ state according to the process: H (q[0]), CNOT (q[0], q[1]);
2. Alice choose to flip some coins (Apply X gate);
3. Bob Apply H gate to all qubit, CNOT(q[0],q[1]);
4. Measure the qubits.

Implement in Q#:

```
operation ComputerPly_Kcoins (qs : Qubit[]) : Unit{

    H(qs[0]);
    let n = Length(qs);
    ApplyToEachA(CNOT(qs[0],_),qs[1..n-1]);
}
operation peoplePlay_Kc0ins (qs : Qubit[], flipPattern: Bool[]) : Unit{
    let n = Length(flipPattern);
    for( i in 0..n-1){
        if(flipPattern[i]) {
            X(qs[i]);
        }
    }
}
operation ComputerAnswer (qs : Qubit[]) : Unit{
    let n = Length(qs);
    ApplyToEachA(H,qs[0..1]);
    ApplyToEachA(CNOT(qs[0],_),qs[1..n-1]);
    H(qs[0]);
}
```

2.3 GHZ game

The referee chooses a three bits string r s t uniformly from the set {000, 011, 101, 110} and sends r to Alice, s to Bob, and t to Charlie. Their answers must be bits: a from Alice, b from Bob, and c from Charlie. They win if $a \oplus b \oplus c = r \vee s \vee t$ and lose otherwise.

Strategy:

1. Prepare shared the entangled state.

$$|\psi\rangle = \frac{1}{2}|000\rangle - \frac{1}{2}|011\rangle - \frac{1}{2}|101\rangle - \frac{1}{2}|110\rangle \, .$$

2. Make the output based on entangled pair and assigned number.

3. If the assigned number is q = 1, then the player performs a Hadamard transform on their qubit of the above state.
4. Measure the Qubits

Implement in Q#:

```
operation CreateEntangledTriple_Reference (qs : Qubit[]) : Unit is Adj {
    X(qs[0]);
    X(qs[1]);

    H(qs[0]);
    H(qs[1]);
    Controlled Z([qs[0]], qs[1]);
    (ControlledOnBitString([false, true], X))([qs[0], qs[1]], qs[2]);
    (ControlledOnBitString([true, false], X))([qs[0], qs[1]], qs[2]);
}
operation QuantumStrategy_Reference (input : Bool, qubit : Qubit) : Bool {
    if (input) {
        H(qubit);
    }
    return ResultAsBool(M(qubit));
}
operation PlayQuantumGHZ_Reference (strategies : (Qubit => Bool)[]) : Bool[] {

    using (qs = Qubit[3]) {
        CreateEntangledTriple_Reference(qs);
        mutable abc = new Bool[3];
        for (i in 0..2) {
            set abc w/= i <- strategies[i](qs[i]);
        }
        ResetAll(qs);
        return abc;
    }
}
```

2.4 CHSH game

The referee chooses questions rs ∈ {00, 01, 10, 11} uniformly, and Alice and Bob must each answer a single bit: a for Alice, b for Bob.
They win if $a \oplus b = r \wedge s$

Strategy:

Alice and Bob will answer correctly with probability cos2 (π/8) ≈ 0.85, which is better than an optimal classical strategy that wins with probability ¾.

Here is the first way. The entangled state shared by Alice and Bob will be $|\psi\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$. For a given angle $\theta \in [0, 2\pi)$, define

$$|\phi_0(\theta)\rangle = \cos(\theta)\,|0\rangle + \sin(\theta)\,|1\rangle\,,$$
$$|\phi_1(\theta)\rangle = -\sin(\theta)\,|0\rangle + \cos(\theta)\,|1\rangle\,.$$

If Alice receives the question 0, she will measure her qubit with respect to the basis

$$\{|\phi_0(0)\rangle\,, |\phi_1(0)\rangle\},$$

and if she receives the question 1, she will measure her qubit with respect to the basis

$$\{|\phi_0(\pi/4)\rangle\,, |\phi_1(\pi/4)\rangle\}.$$

Bob uses a similar strategy, except that he measures with respect to the basis

$$\{|\phi_0(\pi/8)\rangle\,, |\phi_1(\pi/8)\rangle\} \quad \text{or} \quad \{|\phi_0(-\pi/8)\rangle\,, |\phi_1(-\pi/8)\rangle\},$$

depending on whether his question was 0 or 1.

Implement in Q#:

```
// Task 2.2. Quantum strategy
operation QuantumStrategy_CHSH (input : Bool, qubit : Qubit) : Bool {
    if (input) {
        H(qubit);
    }
    return ResultAsBool(M(qubit));
}
operation AliceQuantum (bit : Bool, qubit : Qubit) : Bool {
    // Measure in sign basis if bit is 1, and
    // measure in computational basis if bit is 0
    let basis = bit ? PauliX | PauliZ;
    return ResultAsBool(Measure([basis], [qubit]));
}
operation RotateBobQubit (clockwise : Bool, qubit : Qubit) : Unit {
    let angle = 2.0 * PI() / 8.0;
    Ry(clockwise ? -angle | angle, qubit);
}
operation BobQuantum(bit : Bool, qubit : Qubit) : Bool {
    RotateBobQubit(not bit, qubit);
    return ResultAsBool(M(qubit));
}
operation PlayQuantumCHSH (r:Bool, s:Bool ) : (Bool, Bool) {
    using ((a, b) = (Qubit(), Qubit())) {
        CreateEntangledTriple_CHSH([a, b]);
        let aliceResult = AliceQuantum(r,a);
        let bobResult = BobQuantum(s,b);
        Reset(a);
        Reset(b);
        return (aliceResult, bobResult);
    }
}
```

Reference

1.SY Zhang. Quantum strategic game theory. The 14th International Workshop on Quantum Information Processing, 2011. (arXiv:1012.5141)

2. R. B. Myerson. Game Theory: An Analysis of Conflict. MIT Press, Cambridge, 1991.

3. R. A. Fisher. The Genetic Theory of Natural Selection. Clarendon Press, Oxford, 1930.

4. John Maynard Smith. Evolution and the theory of games. Cambridge University Press, 1982.