Qiskit Quantum Hardware Testing via Implementations of QKD Algorithms

Daniel Escánez-Expósito[1] Pino Caballero-Gil[1] Francisco Martín-Fernández[2]

[1]University of La Laguna, Tenerife, Spain
jdanielescanez@gmail.com, pcaballe@ull.edu.es
[2]IBM Research, NY, USA
paco@ibm.com

SCAN ME

Introduction

This work proposes an in-depth study on the performance of many different quantum algorithms executed on the hardware made available by Qiskit [1] from IBM. In particular, it describes the results of the quantum library QuantumSolver [2] based on the Qiskit SDK, which allows to operate with a wide variety of quantum algorithms that can be run from a command line or a web interface, on real quantum hardware and simulators.

Figure 1: QuantumSolver Logo

Specifically, in its current version, the library contains six easy-to-execute predefined quantum algorithms. In order to corroborate the correct operation of all the developed algorithms and to test IBM's real quantum hardware and simulators, the obtained results have been compared with those presented by IBM [3]. For all of them, two executions of the experimental mode are carried out, one in the local simulator "aer_simulator" and the other in the IBM quantum processor "ibmq_quito", which has five qubits in its superconducting architecture. Both are run 20,000 times with the same parameters.

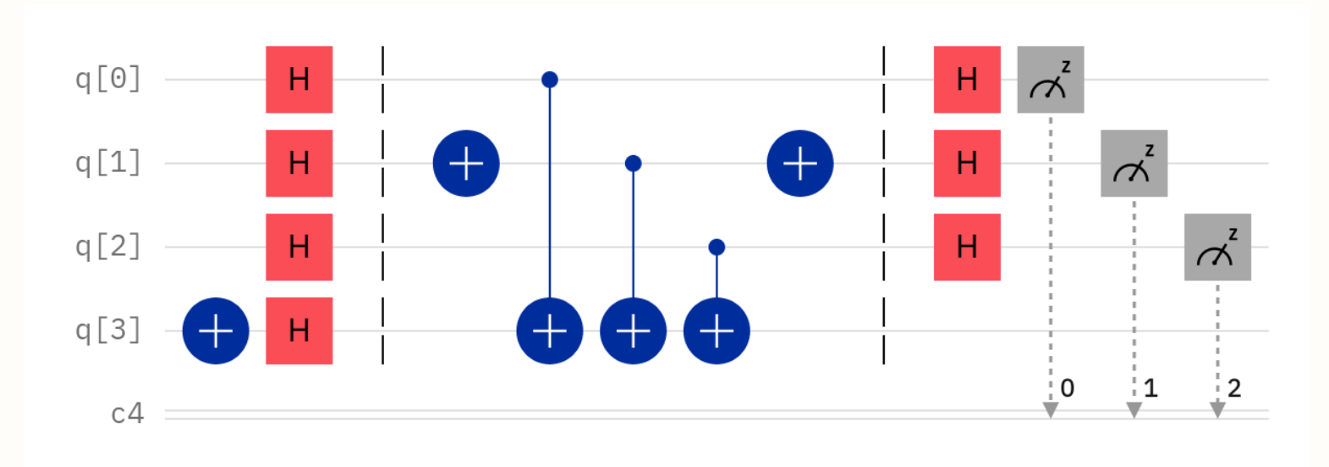


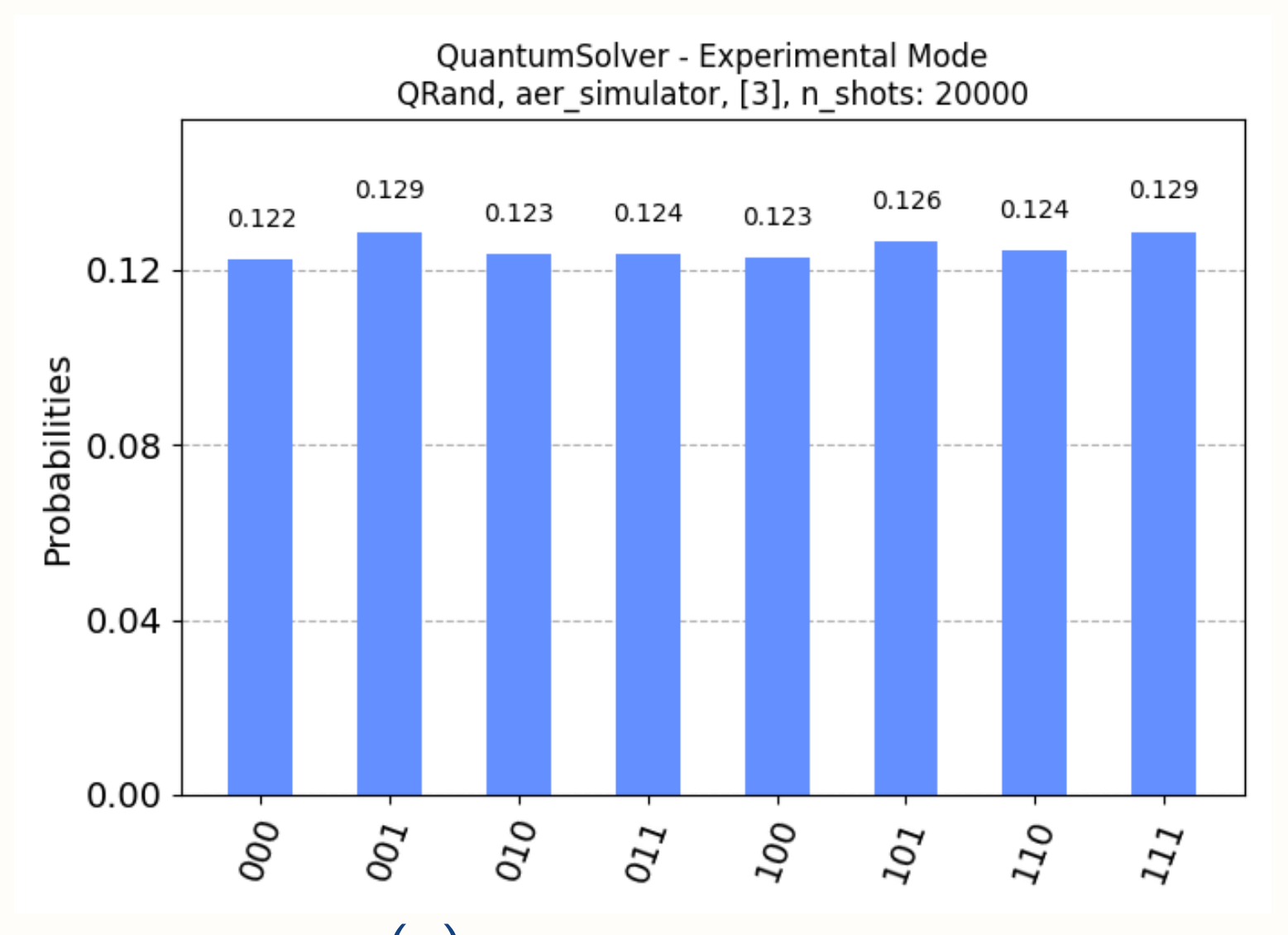
Figure 2: Example of a developed quantum circuit

Equation of the circuit example in Eq. 1.

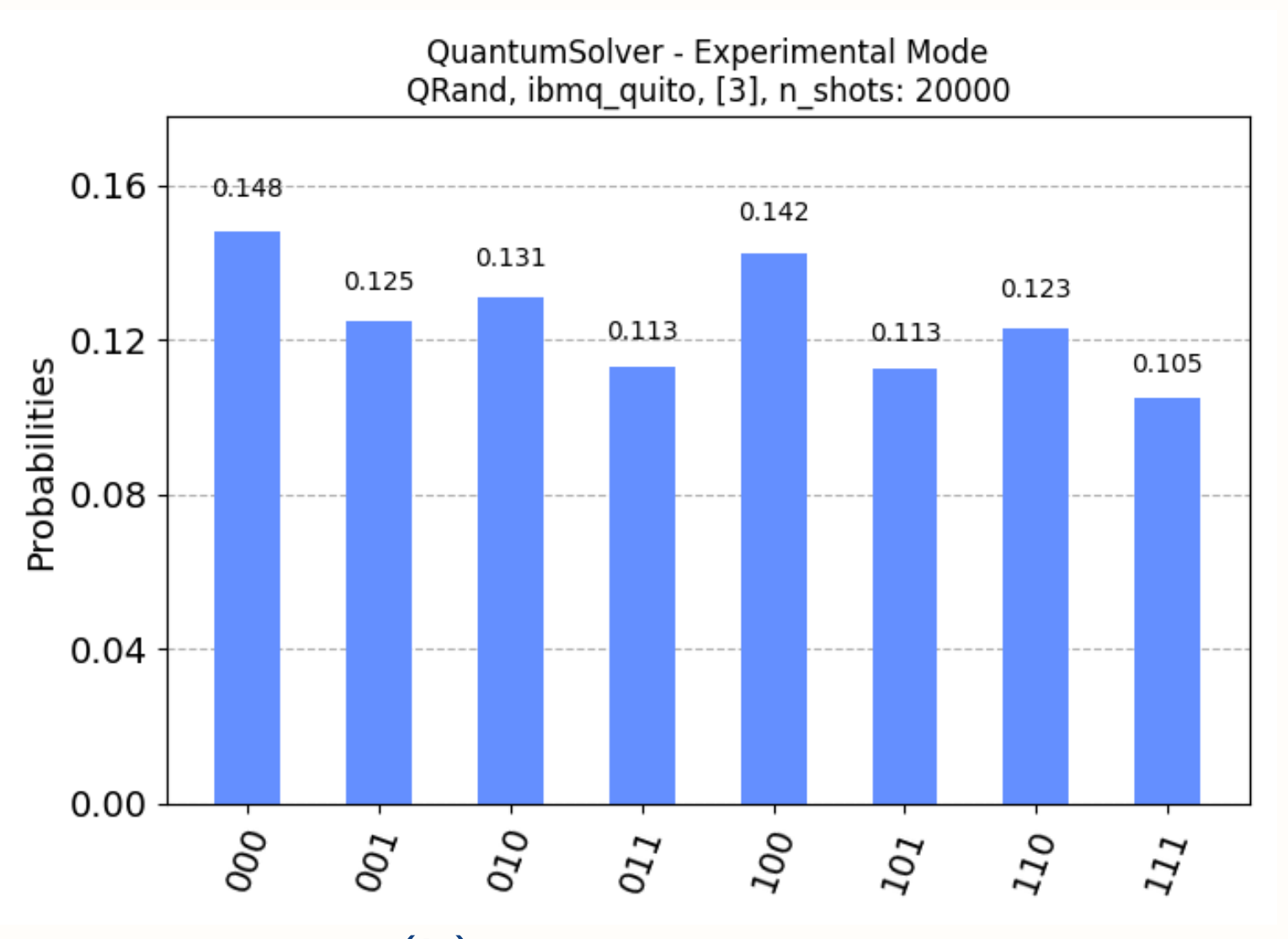
$$
\begin{aligned}
|\psi\rangle &= \frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\left[\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}|y\rangle\right] \\
&= \frac{1}{2^n}\sum_{y=0}^{2^n-1}\left[\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot y}\right]|y\rangle
\end{aligned}
\qquad (1)
$$

Random Number Generation

In random number generation, the expected result is that the number of repetitions of each element is uniformly distributed. In this case, the parameter of three qubits has been used, so that numbers from 0 to 7 should be randomly generated. We can verify that in the two performed runs (Fig. 3a and Fig. 3b) the results are similar; although evidently, in the real quantum case there are inaccuracies that give rise to the errors typical of these physical systems.



(a) Using "aer_simulator"

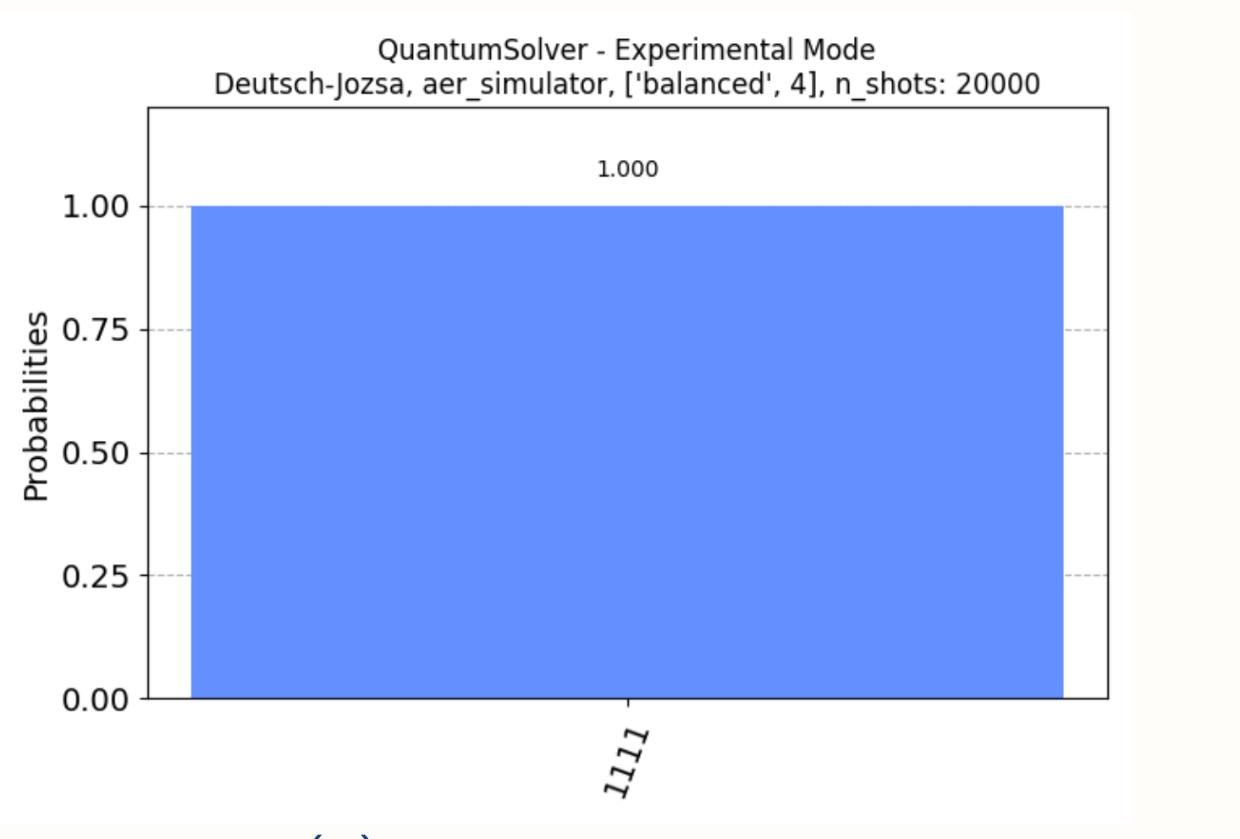


(b) Using "ibmq_quito"

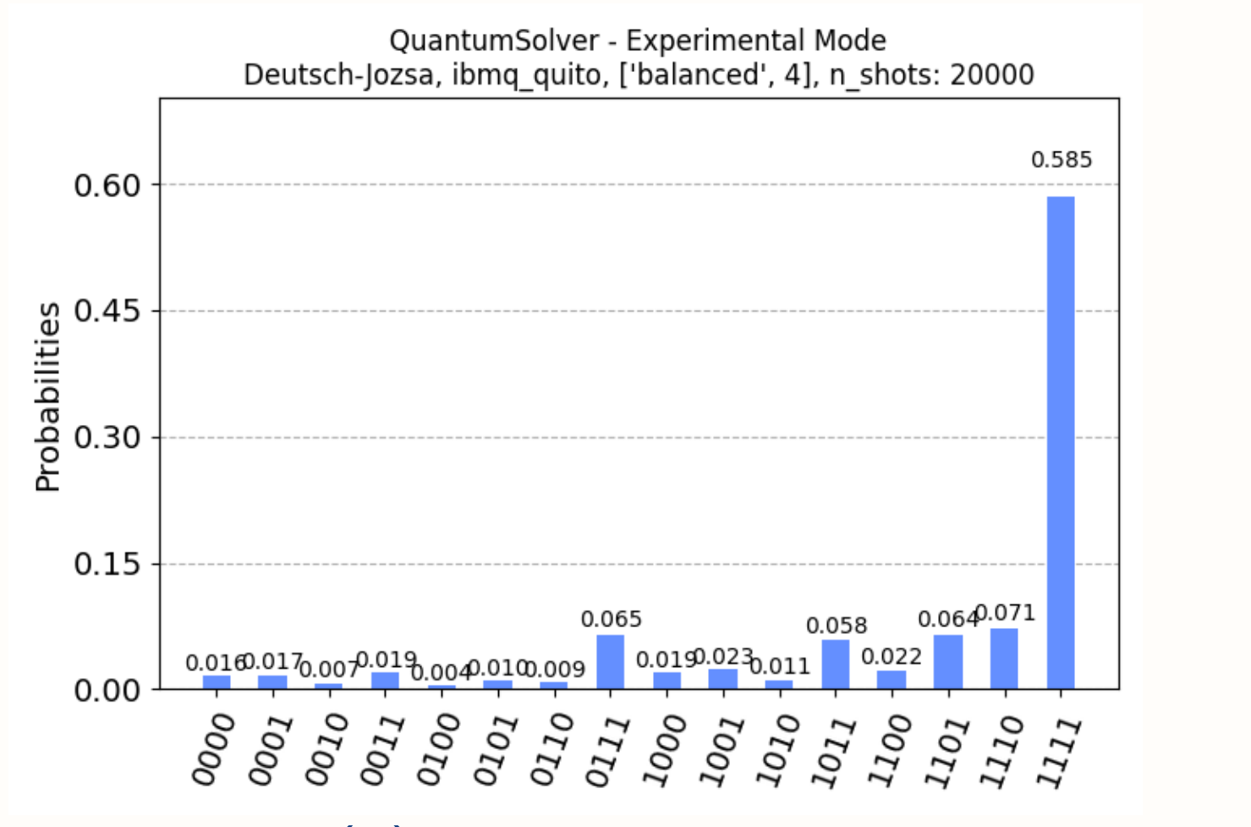
Figure 3: Histograms of the QRand Algorithm

Deutsch-Jozsa Algorithm

The Deutsch-Jozsa algorithm returns a string of n zeros if the implemented oracle encodes a constant function. Obtaining any other case should be interpreted as a balanced function. In the simulation (Fig. 4a), the 100% of executions correctly determine the function type. In real quantum hardware (Fig. 4b), considering that the function is determined as constant only when n zeros are obtained, the obtained probability of success is 99.984%.



(a) Using "aer_simulator"

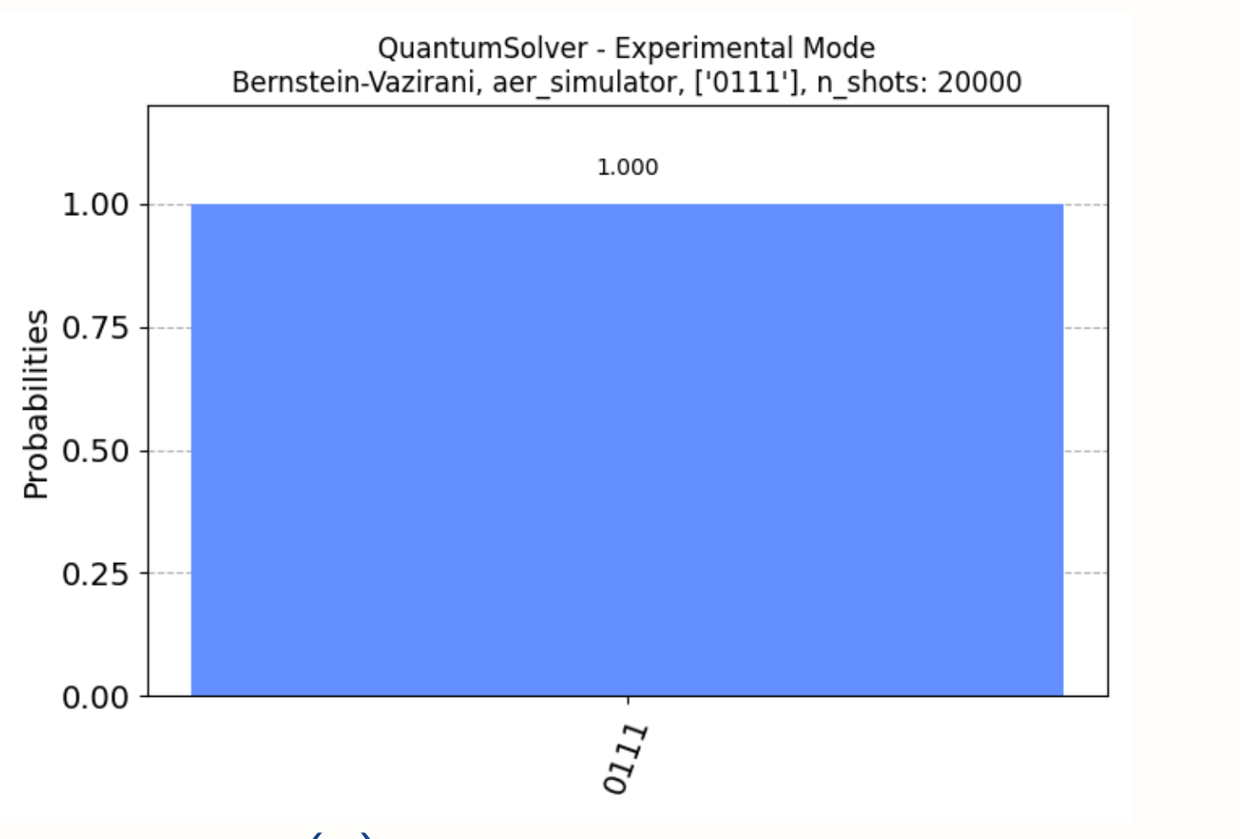


(b) Using "ibmq_quito"

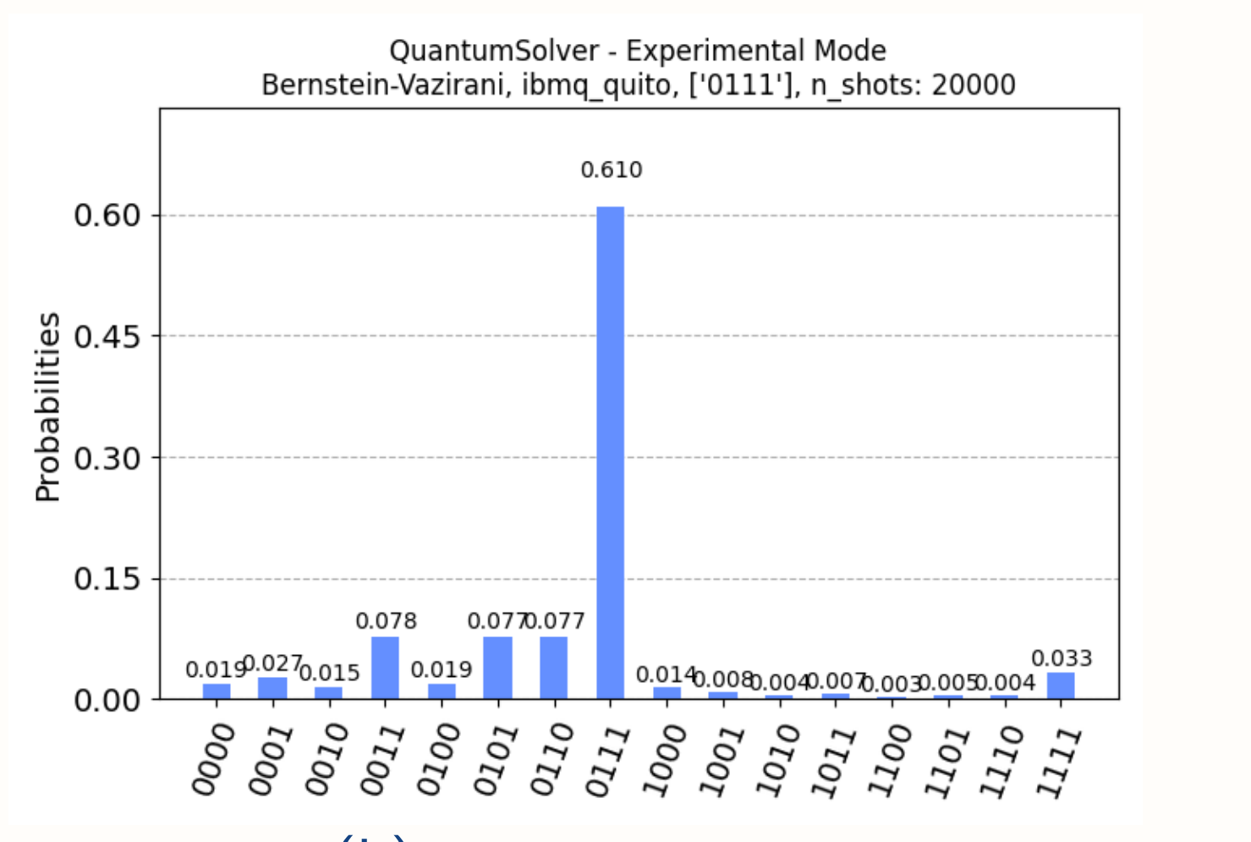
Figure 4: Deutsch-Jozsa Histogram

Bernstein-Vazirani Algorithm

In the executions of the Bernstein-Vazirani algorithm, the simulator (Fig. 5a) hits in all of them the key hidden in the oracle. On IBM hardware (Fig. 5b), it only reaches 61% attempts.



(a) Using "aer_simulator"

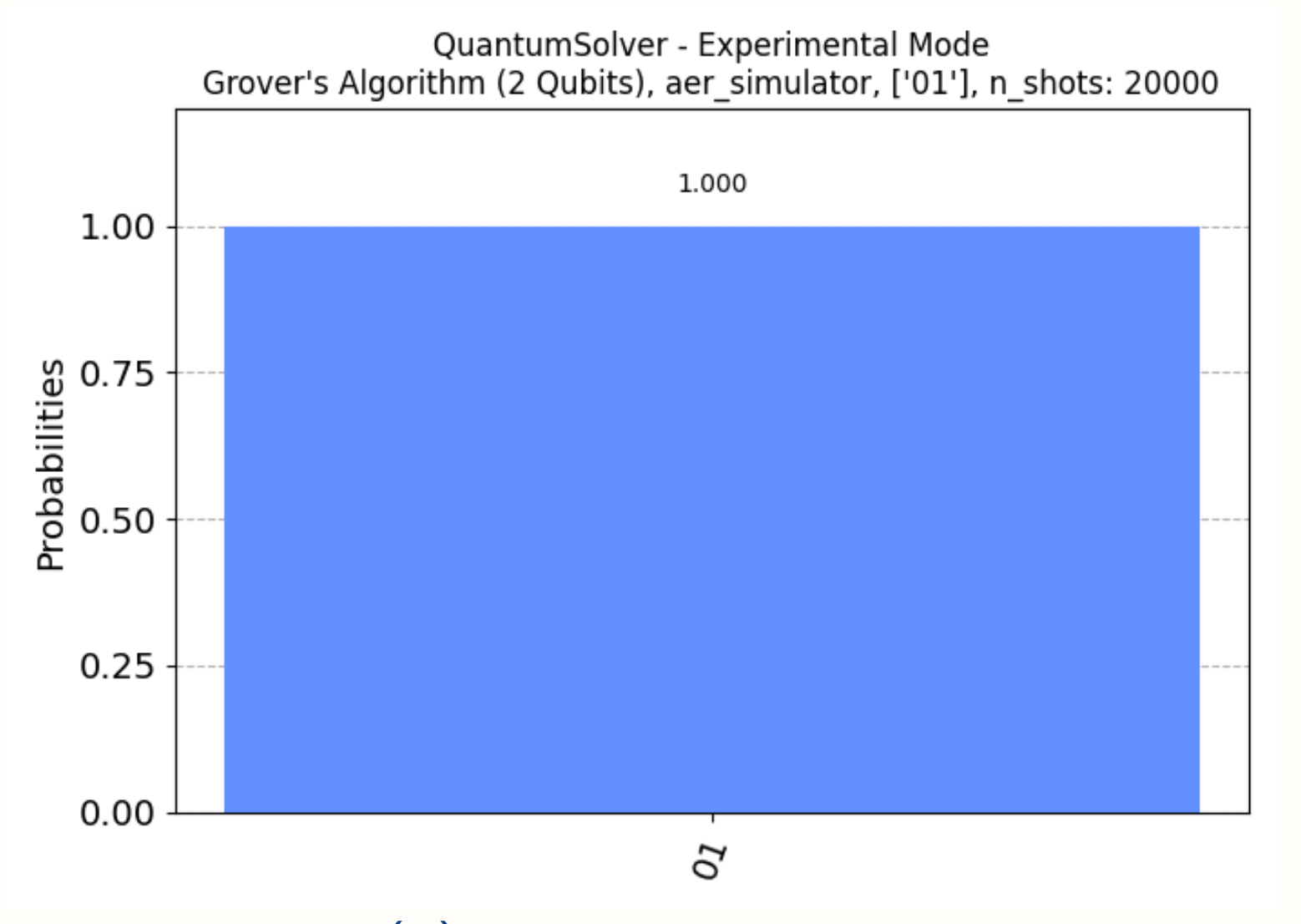


(b) Using "ibmq_quito"

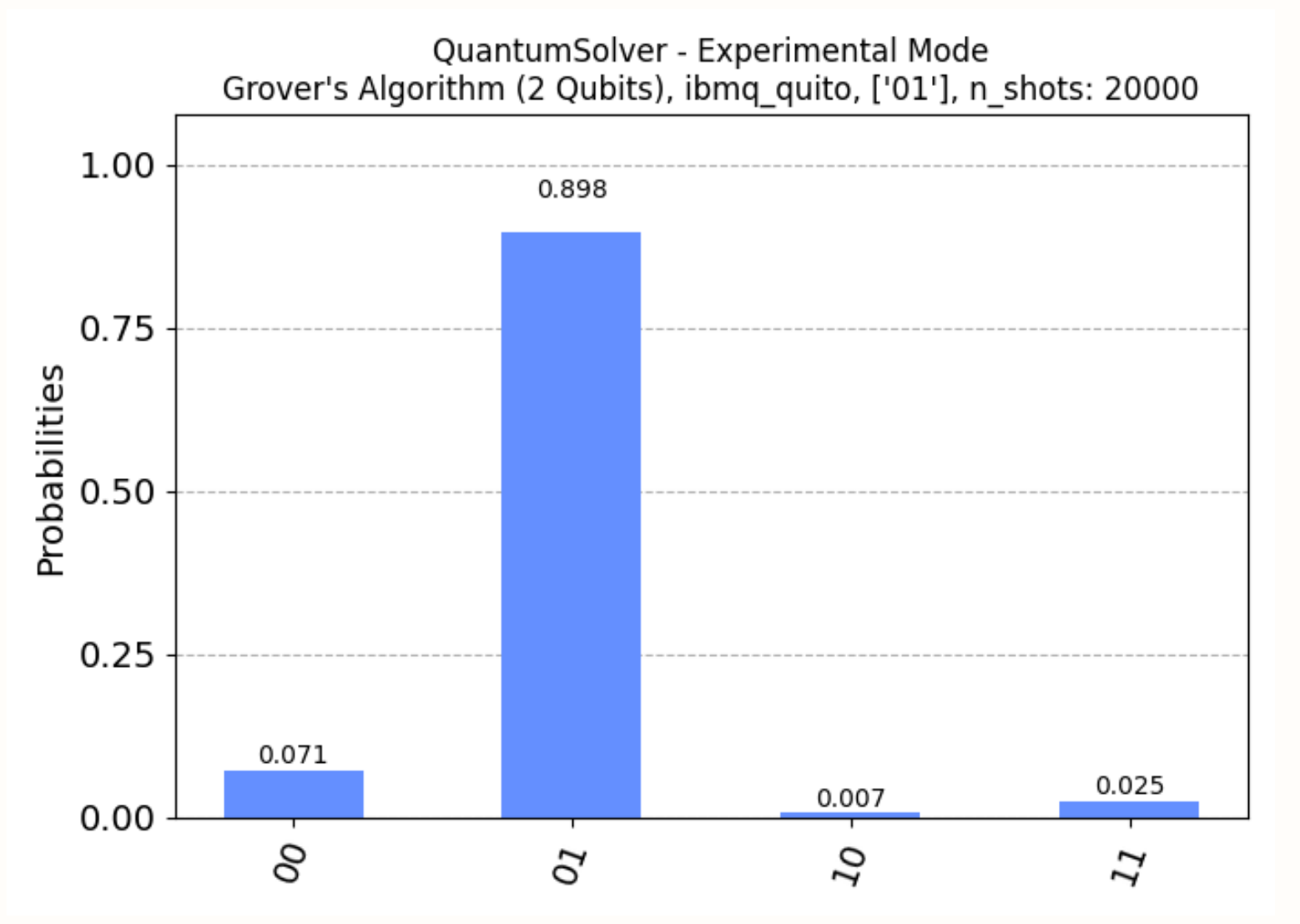
Figure 5: Bernstein-Vazirani Histogram

Grover's Algorithm

Grover's algorithm with parameter defining the state marked "01" should find that element and return it. Again, in the simulation (Fig. 6) a success rate of 100% is obtained, while on current quantum hardware it has been successful in 89.8% cases. The remaining cases have returned untagged states.



(a) Using "aer_simulator"

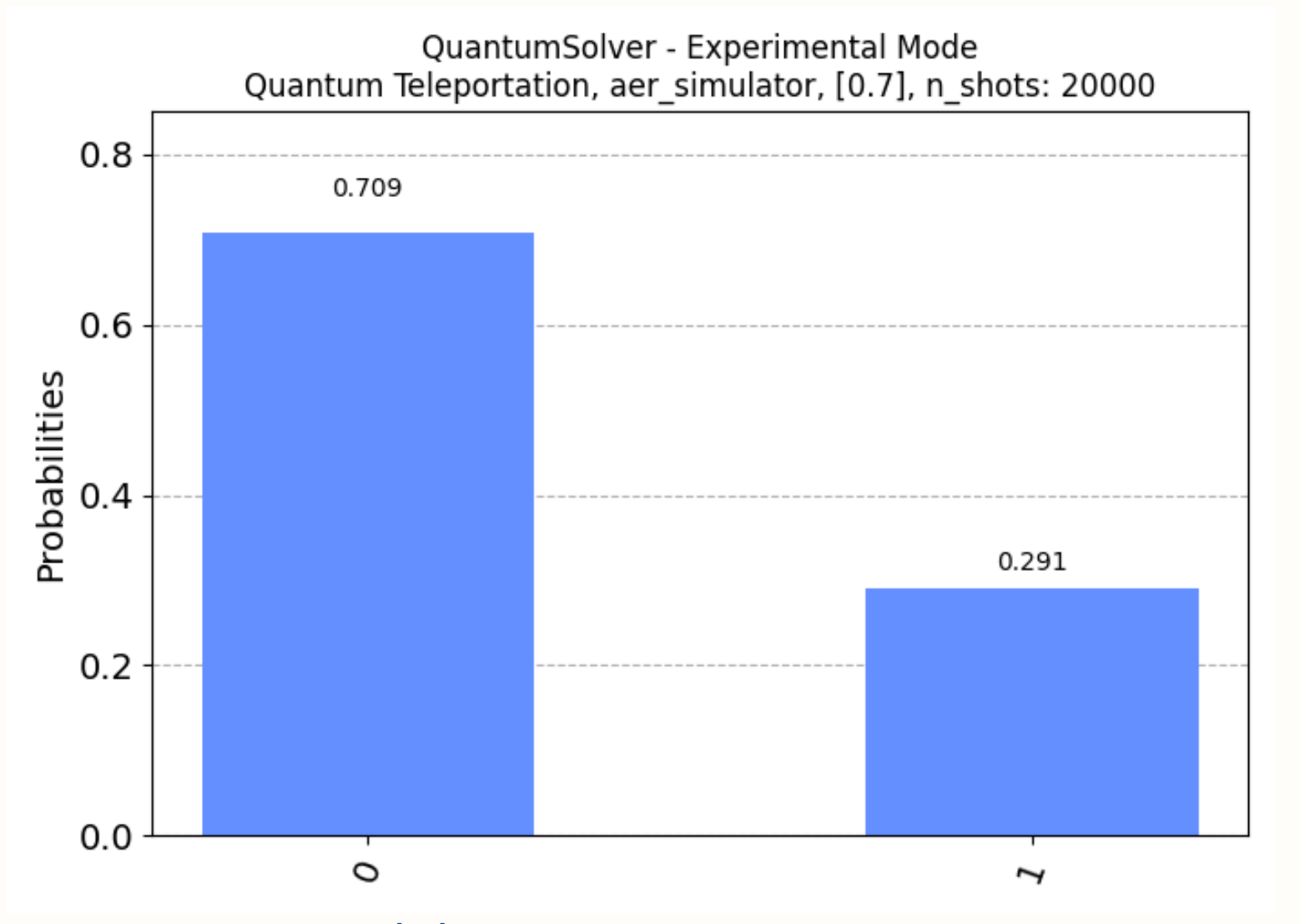


(b) Using "ibmq_quito"

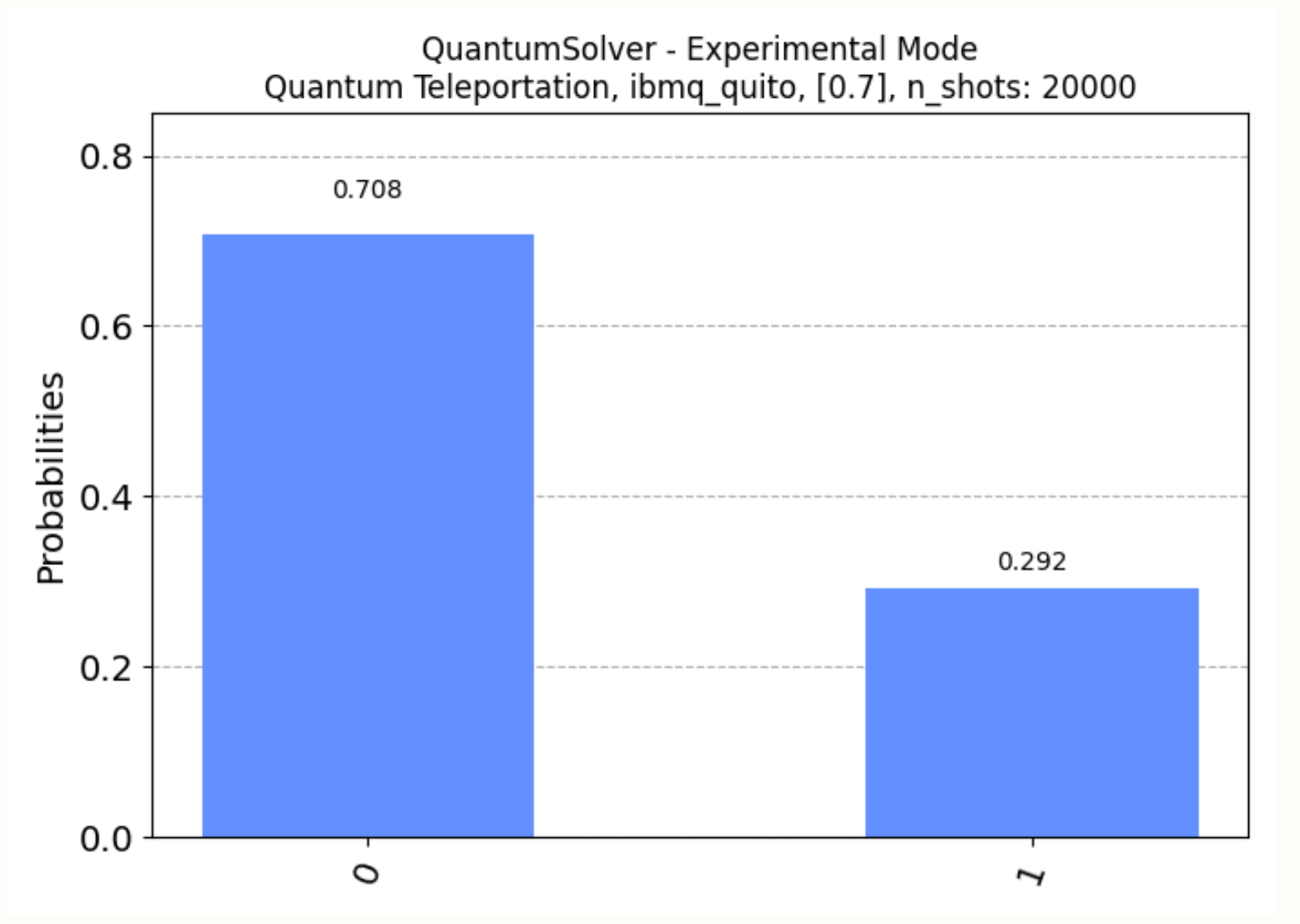
Figure 6: Grover's Algorithm Histogram

Quantum Teleportation

On this occasion, we can observe practically identical results in both execution cases (Fig. 7a and Fig. 7b). This is due to the considerations seen in its implementation, to make it suitable for IBM's hardware [4].



(a) Using "aer_simulator"



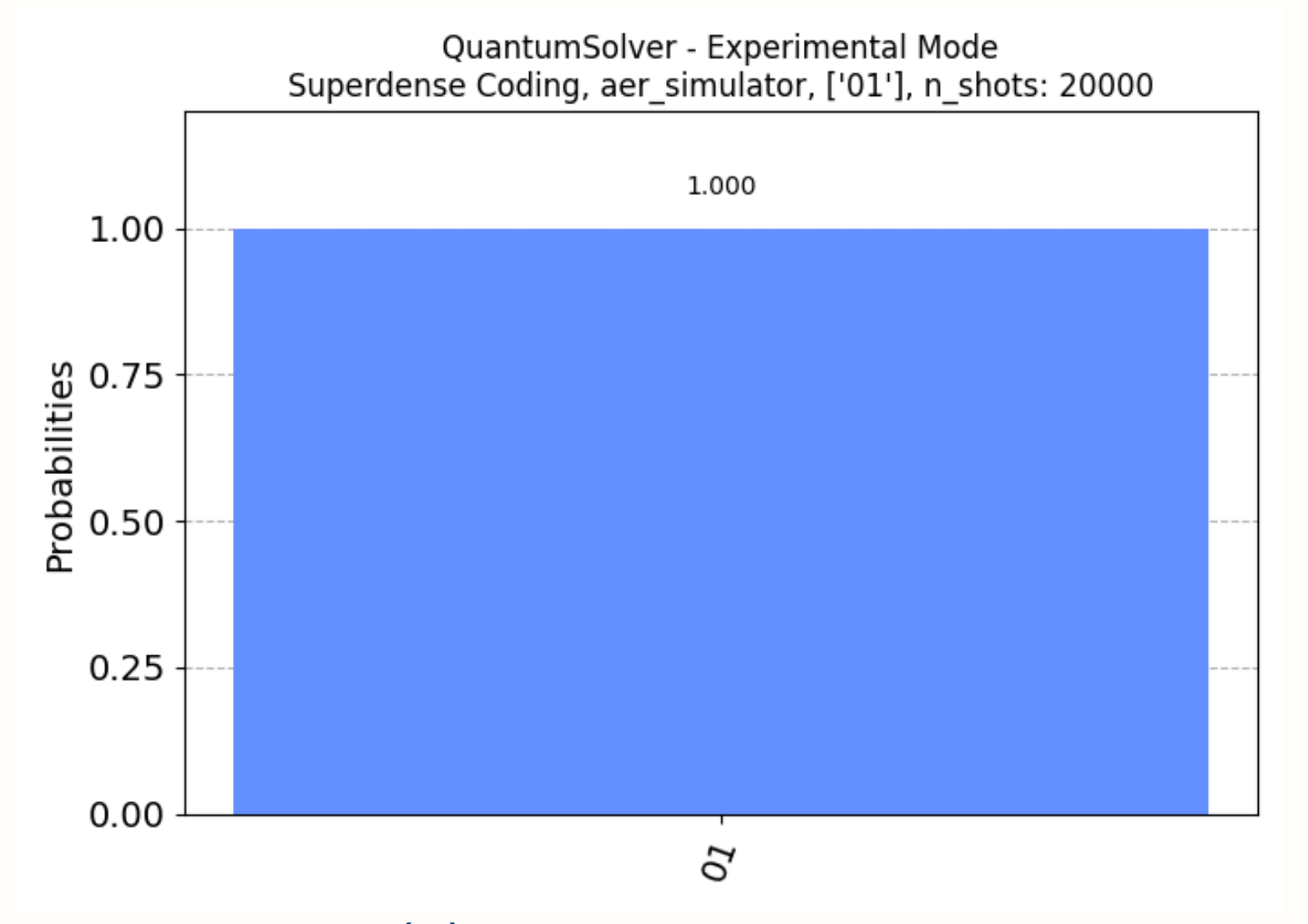
(b) Using "ibmq_quito"

Figure 7: Quantum Teleportation Histogram

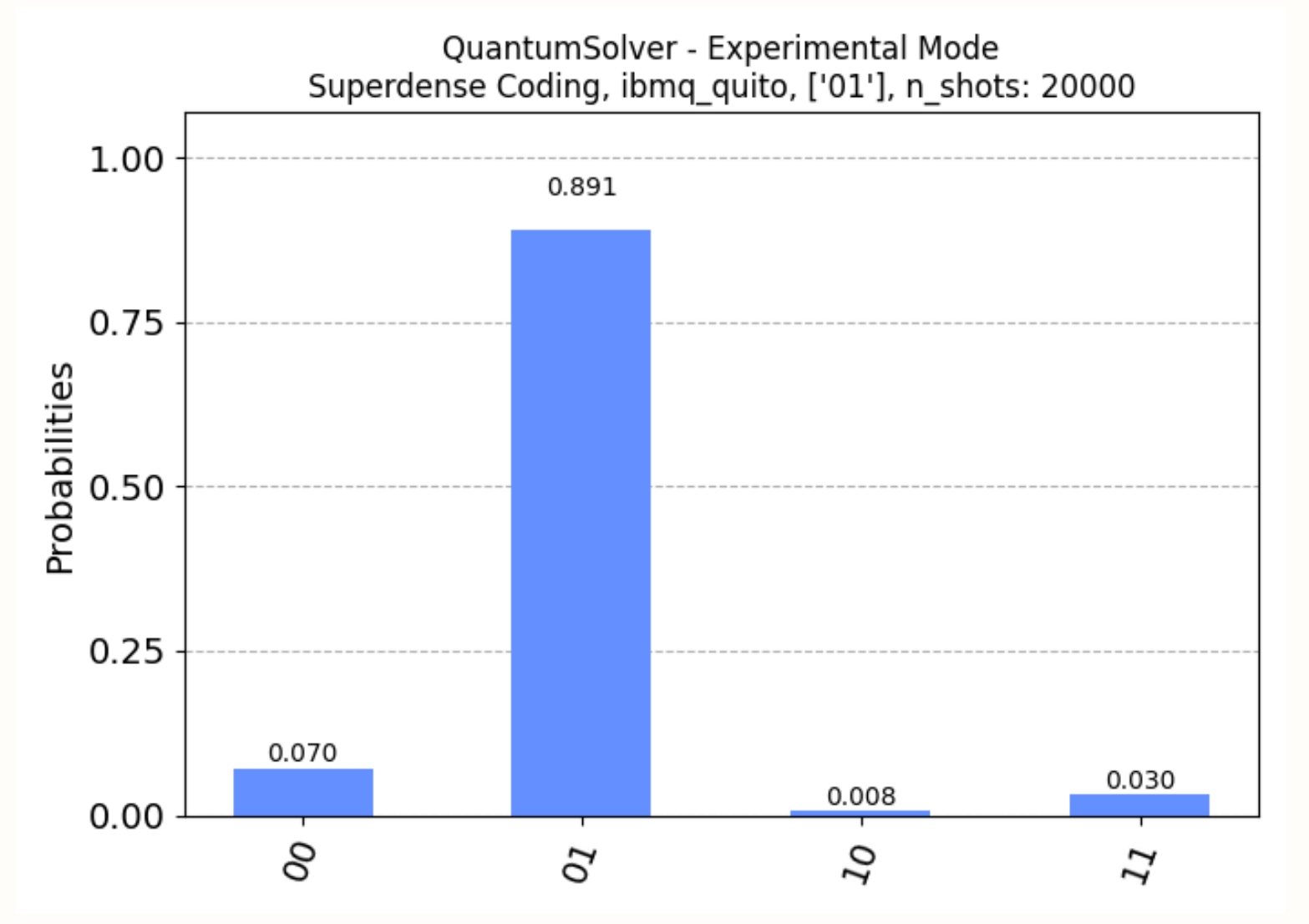
A qubit with a 70% probability of being measured at 0, and a 30% probability of being determined at 1, has been successfully teleported. By performing the experiment a considerably large number of times, in this case 20,000, it can be seen how the probability of obtaining these values has been correctly encoded.

Superdense Coding Protocol

Finally, using the superdense coding protocol (SDC), we try to transmit the value "01", using a single communication qubit. Again, the simulator succeeds in all iterations of the execution while the executions performed on the IBM hardware obtain an 89.1% success rate (Fig. 8), achieving very similar results to those obtained in the execution of Grover's algorithm.



(a) Using "aer_simulator"



(b) Using "ibmq_quito"

Figure 8: SDC Histogram

Conclusions

Through various infographics, it has been possible to develop a deep analysis of the implementations, illustrating the results of the test runs and comparing executions on simulators and real quantum hardware.

As future work, it is proposed to include the optimization of the aforementioned implementations and an analysis of several specific features of their design, such as topology, calibration and use of coupling map, which are important aspects for decision-making in the selection of the used backends. Finally, thanks to the open-source framework Qiskit Metal, several verifications will be carried out to validate the correct operation of the implementations through different graphs and intuitive interfaces that motivate learning.

Acknowledgment

This research has been supported by the Spanish Ministry of Science, Innovation and Universities, the State Research Agency and the European Regional Development Fund under project RTI2018-097263-B-I00, and the Cybersecurity Chair of Binter at the University of La Laguna.

References

[1] IBM, "Qiskit": https://qiskit.org/.

[2] D. Escánez-Expósito, P. Caballero-Gil, F. Martín-Fernández, "QuantumSolver": https://github.com/alu0101238944/quantum-solver/.

[3] IBM, "IBM Quantum": https://quantum-computing.ibm.com/.

[4] IBM, "Quantum Teleportation - Deferred Measurement": https://qiskit.org/textbook/ch-algorithms/teleportation.html#deferred-measurement.