The Journal of Systems and Software 217 (2024) 112165

Contents lists available at ScienceDirect

The Journal of Systems & Software

journal homepage: www.elsevier.com/locate/jss

Exploring the trade-off between computational power and energy efficiency: An analysis of the evolution of quantum computing and its relation to classical computing ☆

Elena Desdentado [a,*], Coral Calero [a,b], Mª Ángeles Moraga [a,b], Manuel Serrano [a,b], Félix García [a,b]

[a] Institute of Technology and Informations Systems, University of Castilla-La Mancha, Camino Moledores s/n, Ciudad Real, 13005, Spain
[b] aQuantum. Green Quantum Algorithms & Software Research, Spain

ARTICLE INFO

Keywords:
Software
Quantum computing
Green software
Green quantum
Energy consumption
Energy efficiency

ABSTRACT

Quantum computing is considered a revolutionary technology due to its ability to solve computational problems that are beyond the capabilities of classical computers. However, quantum computing requires great amounts of energy to run. Therefore, a factor in deciding whether to use quantum computing should be not only the complexity of the problem to be solved, but also the energy required to solve it. This paper presents an empirical study developed with the aim of comparing classical and quantum computing in terms of energy efficiency to determine whether the increased power of quantum computers is offset by their higher energy consumption. To achieve this, a variety of problems with different levels of complexity were tested on both types of computers. Specifically, we used the IBM Quantum computers with a maximum of 5 qubits and an Intel i7, as a classical computer. In addition to this we have also analysed the evolution of the quantum computers, performing measurements on three time periods. Our empirical study showed that there is a variability of results obtained in the three time periods and that quantum computing is not recommended for low-complexity problems, given its high energy consumption, particularly when compared to traditional computing.

1. Introduction

Thanks to technology, our lives have improved noticeably, and problems that in the past required large numbers of people to work for a long period of time can now be solved in seconds. However, some problems cannot be solved due to their complexity. For example, NP-complete problems, such as *Grover's Algorithm*, *N-Queens Problem* and so on, could not be solved until now using classical computing. The objective of quantum computing is to solve these types of problems.

Research into quantum computing has been steadily increasing since 1994 (Dowling Jonathan and Milburn Gerard, 2003), with IBM, Massachusetts Institute of Technology, Harvard University, Max Planck Society, Google Research, among others, dedicated to the field (Swayne, 2022; Anon, 0000a; Hundt, 2022).

According to Ezratty (2022), two quantum revolutions can be identified. The first revolution took place after World War II and involved the implementation of quantum physics in electronics. The second quantum revolution started in 1981 (Gutiérrez, 2020) and involved the introduction of quantum laws to computation. It covers technologies that combine all or part of the ability to control individual quantum objects (atoms, electrons, photons), use quantum superposition and/or entanglement (Ezratty, 2022).

Despite all the favourable aspects, there are also some less positive elements that should be considered. For example, the enormous amount of energy that quantum computing requires is a downside that needs to be taken into account. This energy consumption comes from different sources, mainly from the cooling unit of the quantum processor, since the quantum computing hardware requires a constant temperature of 15 millikelvin (−273 °C) (Hsu, 2015). There are some previous studies that highlight the importance of energy consumption in the quantum computing context, for example, analysing the energy consumption of the gates themselves (Ikonen et al., 2017), or evaluating theoretically the energy advantage that can be obtained through quantum computing (Meier and Yamasaki, 2023). The main differences between these works and the one presented in this paper is that this paper is focused on comparing classical and quantum consumption and that the measurements performed on the quantum computers are not based solely on the gates of the circuits, but on the overall circuit considering

☆ Editor: Dr. Patricia Lago.

* Corresponding author.
E-mail addresses: Elena.DFernandez@uclm.es (E. Desdentado), Coral.Calero@uclm.es (C. Calero), MariaAngeles.Moraga@uclm.es (M.Á. Moraga), Manuel.Serrano@uclm.es (M. Serrano), Felix.Garcia@uclm.es (F. García).

https://doi.org/10.1016/j.jss.2024.112165
Received 21 November 2023; Received in revised form 21 June 2024; Accepted 25 July 2024
Available online 29 July 2024

the total energy consumption derived from the execution time of each execution.

There is a term employed firstly by Preskill (2018) that reflects the current status and limitations of quantum computers, where noise from errors and interference affects their functionality said term is Noisy Intermediate-Scale Quantum (NISQ). The noise limits the maximum circuit depth, or the sequence in which qubits can execute gates one after the other. Because of this, the implementation of quantum algorithms that need deeper circuits is difficult. The NISQ era presents challenges in algorithm creation and appropriate quantum hardware selection because quantum computers can only manage a finite number of qubits and gates.

As quantum computer results are not always reliable, it is necessary to apply other techniques. For example, quantum error correction (Fellous-Asiani, 2022), is a technique that preserves quantum information and improves the fidelity of operations, which, of course, also require some time and therefore energy consumption. As pointed out in Fellous-Asiani (2022), the more precise we want an operation performed on a qubit, the more we have to protect it and the more energy we have to expend to do so.

Another important limitation of these types of computers is the access to them. As we have said there are many companies and research laboratories invested in this technology and some offer access to their quantum computers but with an important restriction, the majority of the users can only access quantum simulators or access-level quantum computers with a small number of qubits. As we previously mentioned, this limits the complexity of the algorithms that we can execute.

Considering that it has been estimated that by 2030, 20% of the global energy consumption will be used by ICTs (Anon, 2018), quantum computing cannot become a new source to increase this figure. In this sense, in early 2022, the quantum energy initiative emerged, proposing a manifesto (Anon, 0000b). As stated in this manifesto, "as quantum technologies drive strong expectations from governments and industries, we believe that a responsible way to use them must include the study of their potential for energy savings and contained environmental footprint".

Therefore, the development and maintenance of quantum software must follow the engineering principles to produce high-quality and sustainable solutions, and therefore quantum software can benefit from applying the knowledge and lessons learned from the software engineering field. This is the main purpose of the "The Talavera Manifesto for Quantum Software Engineering and Programming", which collects some principles and commitments about the field of quantum software engineering and programming, as well as some calls to action (Anon, 0000b).

In this study, we advocate for the application of software engineering, focusing on the energy efficiency of the solutions produced. The aim is to raise awareness of the importance of making the right choice when addressing a software engineering problem by correctly selecting the classical or quantum paradigm. Although there are studies in which quantum computing is believed to be the solution to the great energy needs of ICTs (Anon, 2022), to our knowledge there are no studies that focus on the measurement and comparison of energy efficiency between classical and quantum computing. This study, which can be classified as a laboratory experiment (Stol and Fitzgerald, 2018), was intended to analyse two aspects.

- On the one hand, to compare quantum and classical computing to determine whether the use of quantum computers implies more energy consumption than classical computing to solve the same problem. To pursue this objective, a comparison between the consumption of different algorithmic problems was performed.
- On the other hand, and considering that quantum computers are in constant evolution, we wanted to compare the evolution of the execution time, the energy consumption, and the overall quality of results obtained for the quantum computers throughout time, making the same measurements at three different points of time.

To achieve this, we worked with the next problems:

- *2 taxis, 3 people*. The first problem tested in our study is called "*2 taxis, 3 people*" (Sugi, 0000). Its definition is as follows: A travel agency must group 3 people in 2 taxis. They must assign a taxi considering that person 1 and person 2 get along, while person 3 does not cope with any of them.This is a allocation problem being important the moment how the passengers take the taxi but not the destination.
- *2 taxis, 5 people*. The second problem under test has two versions, two different solutions based on the same premise: a travel agency has to group people in 2 taxis, but this time, there are 5 people. As previously, in this problem the importance is on how the passengers take the taxi and not the destination. In the first case, in version *friends*, person 1 and person 2 can only go together; therefore, the other 3 people have to go in the other taxi. In the second case, in version *enemies*, the only restriction is that person 1 and person 2 cannot go together in the same taxi. Therefore all the combinations that satisfy the mentioned restraint are possible. For both versions there is also a common limitation, 1 taxi can only fit 3 people.
- *The secret number problem*: This problem was first introduced in 1997 (Bernstein and Vazirani, 1997) and it can be seen as an extension of the Deutsch–Jozsa algorithm (Deutsch and Jozsa, 1992). In this scenario, there is a box that takes a bit string as input and returns an answer. The box follows a secret rule, which involves multiplying the input string by another secret bit string to produce the result. The objective is to guess the secret string with a single question as quickly as possible, similar to deciphering a secret combination. For this matter, there is a function (Anon, 0000c), as a black box that takes some bits that represent a number as input and returns a bit, 0 or 1 to express if this number has been discovered or not. There are two versions of this problem, later discussed in this paper: one which employs 3 qubits and an extra qubit for control purposes and another one which employs 4 qubits and an extra qubit for control.

All information related to this study is available at https://github.com/GrupoAlarcos/Exploring-Trade-Off-Computational-Power-Energy-Efficiency-Quantum-Classical.

In the following section we will justify which quantum computers we will use to carry out the study, and we will present the details and results of the work done in quantum computing. The evolution of the measurements over time is presented in Section 2. Section 3 presents the corresponding information from the energy consumption measurements done for classical computing. Section 4 shows a comparison of the consumption required for both approaches (quantum and classical). Section 5 identifies threats to the validity of the studies performed. Finally, the last section outlines the conclusions from the comparison of the results and draws some future work.

2. Energy consumption In Quantum Computers

Quantum computing systems have evolved considerably over the past few years and are considered a major revolution in the field of information technology. Many companies have provided opportunities to use quantum computers. IBM Quantum was selected for this study because its roadmap seemed promising with a long-term plan of improvement and evolution. This platform offers normal users the usage of some of their computers for free, implying some limitations on the capabilities that can be used; for example, when this study was conducted, normal users could only access 5 qubit computers.

Table 1 shows the computers used in this study, all of which are part of the Falcon family of processors, with 5 qubits and different values of quantum volume (Anon, 2021). It is also disclosed the versions of the computers for the three time period measurements performed. Quantum volume is a value 2n, n being the maximum number of

Table 1
IBM Quantum computers and some of their characteristics.

Computer	Processor	Qubits	Quantum volume	Versions
ibmq_lima	Falcon r4T	5	8	1.0.14; 1.0.35; 1.0.36
ibmq_quito	Falcon r4T	5	16	1.1.9; 1.1.28; 1.1.30
ibmq_santiago	Falcon r4L	5	32	1.3.25; 1.4.1; 1.4.3
ibmq_manila	Falcon r5.11L	5	32	1.0.9; 1.0.29; 1.0.30

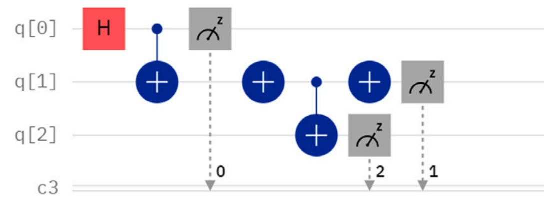
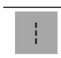
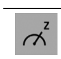


Fig. 1. Quantum circuit to solve the problem *2 taxis, 3 people*.

Table 2
Quantum gates used to create the circuits that solve the problems of this study.

Gate	Description
H	The Hadamard gate is a unitary transformation that maps qubit operations in z-axis to x-axis and vice versa.
⊕	The NOT gate flips |0 > state to |1 >, and vice versa.
	The controlled-NOT gate, also known as the controlled-x (CX) gate, acts on a pair of qubits, with one acting as 'control' and the other as 'target'. It performs a NOT on the target whenever the control is in state |1 >.
I	The identity gate is the absence of a gate. It ensures that nothing is applied to a qubit for one unit of gate time.
Z	The quantum gate Z changes the phase of a qubit without affecting its amplitude. If the qubit is at |0 >, it remains at |0 > but with a changed phase; if it is at |1 >, it also remains at |1 > but with its phase changed.
	To improve the efficiency of your quantum program, the compiler will try to combine gates. The barrier is an instruction to the compiler that prevents these combinations. It is also useful for visualisation purposes.
	Measurement can be used to implement any kind of measurement when combined with gates.

qubits, and steps that can be performed with errors below a specific level (Anon, 0000c). This refers to something we have already talked about in the previous Section 1, quantum computers nowadays can perform a limited number of gates before they reach a state in which errors and interference are common.

Currently, there are several quantum computing paradigms. IBM uses a quantum gate-based programming system, that is, it uses gates to create circuits that solve problems. Quantum logic gates are unitary operators described as unitary matrices relative to certain bases and with a distinctive feature; they are reversible operations. In quantum circuits, quantum gates are represented mathematically using unitary transformation matrices, that is, matrices of complex numbers whose inverse is equal to their conjugate transpose (Serrano et al., 2022). Table 2 contains the gates used in this study (Anon, 0000d).

Therefore, we implemented the circuits that solve the problems mentioned in the Introduction using the quantum gate-based quantum computing paradigm. In following subsections, the measurement protocol used will be discussed along with the results obtained after executing the circuits. Finally, as quantum technology evolves rapidly, there is a subsection that covers the evolution of the results obtained following the same protocol and methodology in three different time periods to observe if there is a change, and if so, to analyse it.

It should be noted that all the information is available at https://github.com/GrupoAlarcos/Exploring-Trade-Off-Computational-Power-Energy-Efficiency-Quantum-Classical.

2.1. Quantum implementation of the problems

In this section, quantum circuits of the problems measured in this study are presented. Note that the beginning state of all qubits is 0 for all circuits. As previously stated in the introduction, three problems have been addressed in this study: the *2 taxis 3 people* problem (Sugi, 0000), two versions of the *2 taxis, 5 people* problem and two implementations of *the secret number* problem, employing the solutions presented by EETHAN Bernstein and Umesh Vazirani (Anon, 0000c).

In Fig. 1, the quantum circuit designed to solve the *2 taxis, 3 people* problem is presented.

The second problem (*2 taxis, 5 people*) has two different approaches. The difference lies in the form of resolution and the considerations of the people in the taxis; one of them requires a larger number of gates and has more possible solutions. Therefore, since it has a larger complexity, we want to check whether this affects the execution time and energy consumption needed to solve the problem. In the first case, in version *friends*, person 1 and person 2 can only go together; therefore, the other 3 people have to go on the other taxi. In the second case, version *enemies*, person 1 and person 2 cannot go together in the same taxi, which means that all combinations taking into account this restriction should be implemented. For both versions there is also a common restriction, only 3 persons can be in each taxi.

In Fig. 2, the quantum circuits designed for the two proposed solutions are presented.

In the third problem, *the secret number* problem, also two versions were designed, the first one employs 3 qubits and another one for control (*3Q+1Q*) and the second one employs 4 qubits and another one for control (*4Q+1Q*).

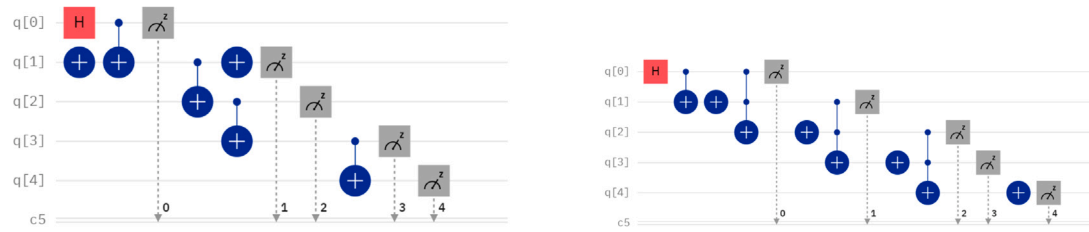
In the *3Q+1Q* version, there are 3 qubits and another extra qubit that is used to control whether or not the solution has been found. In this case, the number to be discovered is 001 and the quantum circuit that solves this problem is shown in Fig. 3.

In the *4Q+1Q* version, there are 4 qubits and one extra control qubit with the same mean, that is, to check whether the solution has been found. In this case the number to be discovered is 0001 and the quantum circuit that solves this problem is shown in Fig. 4.

2.2. Measurement protocol

This section describes the protocol used for estimating the energy consumption required by quantum computers for each one of the problems previously presented. The quantum circuits that implement the problems, *2 taxis, 3 people*; *2 taxis, 5 people* (2 versions) and *the secret number problem* (*3Q+1C version* and *4Q+1C version*), were executed on all quantum computers (*ibmq_manila, ibmq_santiago, ibmq_quito* and *ibqm_lima*).

As sometimes, under the same conditions, the same computer can have different results or outcomes, we decided to repeat the execution as many times as needed to obtain 35 executions. Moreover, as the IBM system only allows the execution of circuits with a maximum of 5 executions on the same computer at once, each problem was executed on each computer on batches of 5 executions, until we achieved the 35 executions already mentioned. For each execution, the time required to execute each circuit (called *time in system* on IBM) was recovered and the mean execution time was calculated for the 35 executions. This time was used to calculate the energy consumption required to solve each problem. IBM quantum computers have a constant energy consumption per hour of 25kWh (Boger, 2023; Castro, 2023). Therefore, the energy consumed to solve a given problem is calculated by multiplying the average time required to solve the problem by 25kWh.



(a) 2 taxis, 5 people *friends*.   (b) 2 taxis, 5 people *enemies*.

Fig. 2. Quantum circuits to solve the problem *2 taxis, 5 people*, versions *friends* and *enemies*.

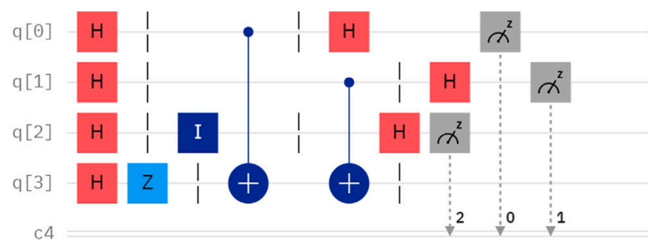


Fig. 3. Quantum circuit implementing *the secret number* problem with 3 qubits and an extra qubit for control (*3Q+1C*).

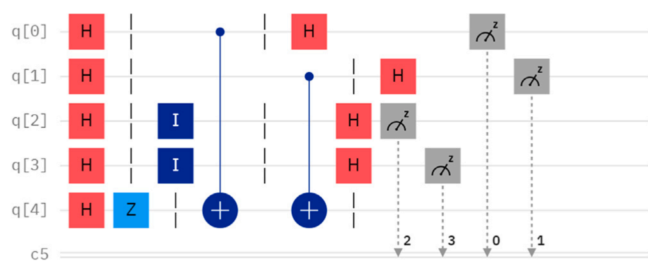


Fig. 4. Quantum circuit implementing *the secret number* problem with 4 qubits and an extra qubit for control (*4Q+1C*).

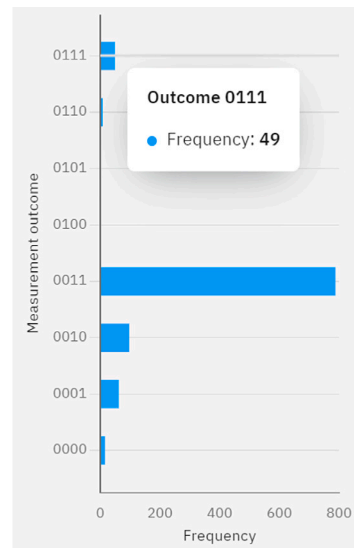


Fig. 5. Example of frequency of the outcomes after an execution of a quantum circuit for *the secret number* problem, *3Q+1C* version.

2.3. Time and consumption results

Once the circuit is executed, the IBM system shows the time it took to do so, which they call *time in system*. In Table 3, the results of execution time and energy consumption for each problem and in the different quantum computers are shown (all these executions were performed in the second week of May 2022). As we can see, the quantum computer that takes the longest time to execute the circuits is *ibmq_santiago* and the fastest in most cases is *ibmq_quito*.

We can see that the pairs *ibmq_lima* and *ibmq_quito* have a similar execution time, and this also happens for the pairs *ibmq_santiago* and *ibmq_manila*. It can also be observed that between these two pairs, the difference is important, *ibmq_santiago* and *ibmq_manila* almost doubles the execution time of *ibmq_lima* and *ibmq_quito*.

The main difference between the pairs is in the Volume (see Table 1), while *ibmq_santiago* and *ibmq_manila* have a volume of 32, *ibmq_lima* and *ibmq_quito* have 8 and 16, respectively. This correlation could indicate that as the Volume increases, implying the possibility deeper circuits, the execution time and energy consumption increase as well.

2.3.1. Success rate

As a counterpart to classical computing, quantum computers executions are prone to errors due to noise, incoherencies and other problems related to the gates, or to the non-isolation of the qubits (Fellous-Asiani, 2022). Therefore, in one single execution, it is common to launch the quantum circuit multiple times (referred to as *shots* in our context), providing us with the distribution of the outcomes obtained. It is worth noting that the displayed execution time corresponds to a single run of the circuit. The default value for each execution set by IBM is 1024 shots, which was maintained throughout the study.

The success rate of each execution was calculated by considering the frequency of each of the possible outcomes, gathering all the shots that ended in a correct solution, and calculating the total frequencies for these correct executions, that is, the success value. Fig. 5 shows the distribution of the outcomes obtained for a certain execution.

We have to keep in mind that we are calculating this metric ourselves and we are not considering the reliability of the gates nor the specific layout of the quantum computers employed; this will be discussed in the future work on Section 6.

The correct outputs obtained by each computer for each one of the problems included in this study are shown in Table 4, in this case, the computer with the highest overall number of correct outputs is *ibmq_lima* and the worst computer in this matter is *ibmq_santiago*.

The success rate was calculated for each one of the 35 executions performed on each quantum computer for all the problems considering the correct outputs and the total shots executed, which are 1024 shots for every execution; then the average success rate as a percentage was obtained; see Table 5.

For the first problem, *2 taxis, 3 people*, we can observe that the quantum computer that has a higher success rate is *ibmq_lima*, achieving a success rate of 92%. For the second problem, *2 taxis, 5 people*, *friends* version, we can observe that *ibmq_manila* achieves the highest success rate (82%). For the *enemies* version of this problem *ibmq_quito* is the quantum computer that has a higher success rate of 63%. Finally,

Table 3
Execution time in seconds and energy consumption (EC) in Wh for the five problems proposed on the four quantum computers tested.

Quantum computer	*ibmq_manila*		*ibmq_santiago*		*ibmq_quito*		*ibmq_lima*	
Problem	Time(s)	EC (Wh)	Time(s)	EC (Wh)	Time(s)	EC (Wh)	Time(s)	EC (Wh)
2 taxis, 3 people	5.48	38.07	6.37	44.24	3.46	24.02	3.43	23.81
2 taxis, 5 people friends	5.57	38.71	6.44	44.71	4.16	28.88	4.28	29.74
2 taxis, 5 people enemies	5.64	39.15	6.41	44.46	4.2	29.11	4.3	29.94
SN (3Q+1C)	5.51	38.25	6.5	45.17	4.09	28.41	4.13	28.67
SN (4Q+1C)	5.63	39.04	6.45	44.75	4.08	28.33	4.1	28.49

Table 4
Correct outputs obtained from the circuits executed on each quantum computer selected.

		CORRECT OUTPUTS			
	Quantum computer	*ibmq_manila*	*ibmq_santiago*	*ibmq_quito*	*ibmq_lima*
Problem	*2 taxis, 3 people*	910	811	897	937
	2 taxis, 5 people friends	837	728	732	753
	2 taxis, 5 people enemies	613	628	649	602
	SN (3Q+1C)	628	622	799	863
	SN (4Q+1C)	544	600	736	677
		706	678	763	766

Table 5
Success rate of the circuits executed on each quantum computer selected.

		SUCCESS RATE			
	Quantum computer	*ibmq_manila*	*ibmq_santiago*	*ibmq_quito*	*ibmq_lima*
Problem	*2 taxis, 3 people*	89%	79%	88%	92%
	2 taxis, 5 people friends	82%	71%	72%	74%
	2 taxis, 5 people enemies	60%	61%	63%	59%
	SN (3Q+1C)	61%	61%	78%	84%
	SN (4Q+1C)	53%	59%	72%	66%
		69%	66%	75%	75%

for *the secret number* problem, first version with 3 qubits and another control qubit, it can be observed that *ibmq_lima* has the highest success rate (84%), being *ibmq_quito* the quantum computer that achieves the highest success rate (72%) for the second version (4 qubits and another control qubit).

Overall we can say that the computers which have the highest success rate are both *ibmq_quito* and *ibmq_lima* with a 75%.

2.4. Comparative analysis over time

In a previous study (Fernández et al., 2021), the energy consumption measurements were performed each month, the variability of the data obtained during this study was not too significant; therefore, for this study it was decided that the energy consumption measurements would be made over a longer period of time, around 6 months. However, after two measurements in September 2021 and April 2022 it was announced that one of the computers used was going to be decommissioned, *ibmq_santiago*. Consequently, we decided to perform another energy consumption measurement in May 2022 to have at least three sets of data to compare. The same measurement protocol, explained in Section 3.2, was used to calculate the energy consumption the three times. With the data obtained, we carried out a comparative study to assess variations in execution time and energy consumption, as well as the overall quality of the results obtained.

The obtained results were compared from different aspects: the execution time and energy consumption; the success rate of the results obtained; and the quantum efficiency (together with the quantum computer efficiency and the quantum problem efficiency).

2.4.1. Execution time and energy consumption evolution

Firstly, the evolution of the execution time and energy consumption of the computers in September of 2021, April of 2022 and May of 2022 is analysed. As the energy consumption of these quantum computers according to IBM is constant and equal to 25kWh, the execution time and energy consumption are directly proportional, so in this section we will focus on energy consumption.

Table 6 shows, for the first problem (*2 taxis, 3 people*), that the computer with the lowest energy consumption overall is *ibmq_lima*, (needing 23.81 Wh to execute the circuit in the third time period, May 2022).

Note that in this case the names of the computers has been simplified for better visualisation.

For the second problem, *2 taxis, 5 people*, for both the first and second versions, the computer with the lowest energy consumption (28.88 Wh- first version, 29.11 Wh- second version; both in May 2022) is *ibmq_quito*. Finally, for the secret number problem, it can be seen that for both versions, the computer with the lowest energy consumption is also *ibmq_quito*, (needing 28.41 Wh and 28.33 Wh for the first and second versions, respectively; both in May 2022).

If we then observe the computers for each time period individually, in September, the quantum computer with the lowest average energy consumption is *ibmq_lima*, (which needed 25.366 Wh). In April the lowest average energy consumption was obtained by *ibmq_quito*, (needing 28 Wh). And finally, in May, the computer with the lowest average energy consumption is also *ibmq_quito*, (needing 27.75 Wh). However, in both April and May, the differences between *ibmq_quito* and *ibmq_lima* are very small (only a 0.78% and a 1.36% greater for *ibmq_lima* in April and May respectively).

To simplify the data and compare the quantum computers used, the mean energy consumption of each one of them throughout the three time periods for every problem was calculated, this can be seen in Fig. 6. In general, the computer that needs the highest energy consumption to execute is *ibmq_santiago*, immediately followed by *ibmq_manila*. As can also be observed, both *ibmq_quito* and *ibmq_lima* have a significantly lower energy consumption than the other two, *ibmq_lima* achieving a lower energy consumption. Also, it can be observed that *ibmq_quito* and *ibmq_lima* have very similar consumption behaviour.

These results seem to reaffirm what has already been stated previously that there might be a correlation between the Volume of a computer and its energy consumption.

2.4.2. Success rate

As mentioned in Section 2.3.1, the success rate was calculated for each of the 35 executions performed on each quantum computer for all the problems and over the three time periods under analysis. To obtain the success rate, it was considered both the correct outputs obtained and the total number of *shots* performed per execution (the default number of shots determined for all executions is 1024). These correct outputs obtained for each quantum computer on each one of the problems are shown in Table 7.

In general, we can observe that *ibmq_santiago* has the highest overall number of correct outputs in September of 2021 with a mean of 811.9 out of the 1024 total *shots*. It also has the highest number in April 2022, with 733.2 correct outputs. But finally in May 2022, it is debunked by *ibmq_lima* which has 766.4 correct outputs and *ibmq_santiago* becomes the computer with the least amount of correct outputs in this period with just 677.8. Keep in mind that these are the means of the number of correct outputs obtained for each quantum computer.

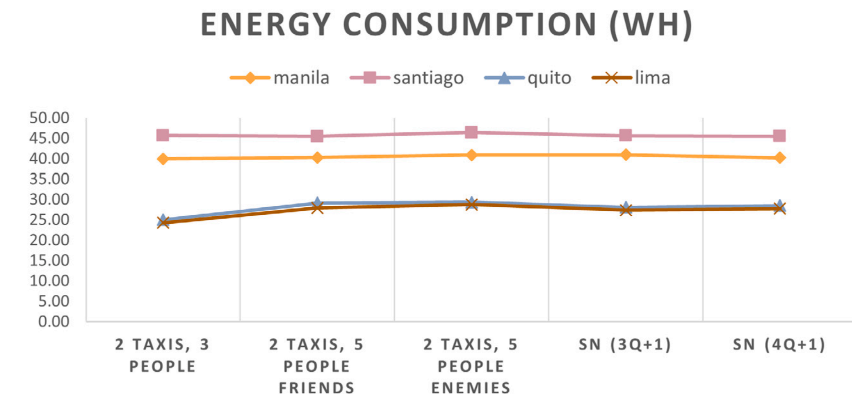


Fig. 6. Average energy consumption (Wh) throughout the three time periods on each one of the quantum computers for every problem.

Table 6
Energy consumption of the executions made on quantum computers on three time periods.

	ENERGY CONSUMPTION (Wh)											
	September 2021				April 2022				May 2022			
Computer	*manila*	*santiago*	*quito*	*lima*	*manila*	*santiago*	*quito*	*lima*	*manila*	santiago	*quito*	*lima*
2t, 3p	44.47	46.81	27.48	24.90	37.46	46.01	23.65	24.24	38.07	44.24	24.02	23.81
2t, 5p friends	44.86	46.35	28.97	25.22	37.36	45.45	29.54	28.88	38.71	44.71	28.88	29.74
2t, 5p enemies	45.44	48.47	29.42	26.35	38.29	46.32	29.42	30.00	39.15	44.46	29.11	29.94
SN 3Q+1C	47.82	45.89	27.44	24.76	36.86	45.87	28.39	28.75	38.25	45.17	28.41	28.67
SN 4Q+1C	44.86	44.60	28.27	25.60	36.83	47.24	29.00	29.22	39.04	44.75	28.33	28.49
	45.49	46.42	28.32	25.37	37.36	46.18	28.00	28.22	38.64	44.67	27.75	28.13

Table 7
Correct outputs of the circuits executed on each quantum computer selected on three time periods.

	CORRECT OUTPUTS											
	September 2021				April 2022				May 2022			
Computer	*manila*	*santiago*	*quito*	*lima*	*manila*	*santiago*	*quito*	*lima*	*manila*	*santiago*	*quito*	*lima*
2t, 3p	905	966	801	930	899	915	917	779	910	811	897	937
2t, 5p friends	854	852	767	729	838	834	616	576	837	728	732	753
2t, 5p enemies	754	741	629	673	674	638	542	484	613	628	649	602
SN (3Q+1C)	652	778	736	839	693	724	766	732	628	622	799	863
SN (4Q+1C)	617	721	557	687	558	555	699	655	544	600	736	677
	756.4	811.6	698	771.6	732.4	733.2	708	645.2	706.4	677.8	762.6	766.4

To facilitate the interpretation of the results, the success rate as a percentage was obtained by calculating the average of the 35 executions for each case considering the number of *shots*, 1024. The resulting values are shown in Table 8.

If we consider the computers for each one of the time periods individually, in September the computer with the highest average success rate is *ibmq_santiago* (79%). In April the highest average success rate corresponds to both *ibmq_manila* and *ibmq_santiago*, achieving a success rate of 72%. Finally, we can see that in May the computers with the highest average success rates are both *ibmq_quito* and *ibmq_lima* (75%).

It seems that there is no specific pattern that allows us to identify a better quantum computer based on its success rate.

This means that, as mentioned previously, the success rate is a problem that appears to be present in all the quantum computers and almost on the same amount (the maximum difference is 10%).

2.4.3. Quantum efficiency

In the context of this study, Quantum Efficiency measure was obtained to ascertain how well a quantum computer or problem performs with the least number of resources, in this case energy consumption. Quantum Efficiency will be used to better compare the performance of the study in general, and the problems and quantum computers in particular.

The Quantum Efficiency was calculated as the ratio of the useful work output (obtained by using the correct outputs values previously discussed) and the energy consumption needed to obtain that output, (obtained as the energy consumption of an execution in a determined number of shots; in our case the number of shots is 1024, the default value set by IBM). Therefore, in this context a quantum computer or problem is considered more efficient than another if it is able to produce a higher output success rate by using the same amount of energy consumption. To clarify how this measure was obtained, the mathematical formula employed is as follows (1):

$$Quantum\ efficiency = \frac{Correct\ outputs}{(Energy\ Consumption \times Total\ number\ of\ shots)} \tag{1}$$

Table 9 displays the values obtained for each problem on each computer for each time period. Focusing on May 2022, which is the most recent data, it appears that both *ibmq_manila* and *ibmq_santiago* have worse performances than *ibmq_quito* and *ibmq_lima*.

This is aligned with previous results, where the difference in consumption between the identified pairs (*ibmq_manila-ibmq_santiago* and *ibmq_lima-ibmq_quito*) was high but the differences between the success rates of the computer were low.

Table 8
Success rate of the circuits executed on each quantum computer selected on three time periods.

	SUCCESS RATE											
	September 2021				April 2022				May 2022			
Computer	*manila*	*santiago*	*quito*	*lima*	*manila*	*santiago*	*quito*	*lima*	*manila*	*santiago*	*quito*	*lima*
2t, 3p	88%	94%	78%	91%	88%	89%	90%	76%	89%	79%	88%	92%
2t, 5p friends	83%	83%	75%	71%	82%	81%	60%	56%	82%	71%	72%	74%
2t, 5p enemies	74%	72%	61%	66%	66%	63%	55%	47%	60%	61%	63%	59%
SN (3Q+1C)	64%	76%	77%	82%	68%	71%	75%	72%	61%	61%	78%	84%
SN (4Q+1C)	60%	70%	54%	67%	57%	57%	68%	64%	53%	59%	72%	66%
	74%	79%	69%	75%	72%	72%	70%	63%	69%	66%	75%	75%

Table 9
Quantum efficiency obtained for the quantum computers tested for each problem in three time periods.

	QUANTUM EFFICIENCY											
	September 2021				April 2022				May 2022			
Computer	*manila*	*santiago*	*quito*	*lima*	*manila*	*santiago*	*quito*	*lima*	*manila*	*santiago*	*quito*	*lima*
2t, 3p	1.9789	2.0081	2.8384	3.6546	2.3492	1.9344	3.8055	3.1353	2.3378	1.7857	3.6636	3.8639
2t, 5p friends	1.8502	1.7909	2.5889	2.8152	2.1949	1.7822	2.0311	2.4931	2.1183	1.5880	2.4931	2.4882
2t, 5p enemies	1.6285	1.4858	2.0734	2.5047	1.7237	1.3601	1.8695	1.5667	1.5326	1.3720	2.1642	1.9706
SN (3Q+1C)	1.3384	1.6561	2.8061	3.3118	1.8448	1.5479	2.6418	2.5034	1.5948	1.3505	2.7455	2.9299
SN 4Q+1C)	1.3375	1.5695	1.9102	2.6172	1.5477	1.2066	2.3448	2.1903	1.3576	1.3184	2.5415	2.3166
	1.6267	1.7020	2.4434	2.9807	1.9320	1.5662	2.5385	2.3779	1.7882	1.4829	2.7216	2.7139

Table 10
Quantum computer efficiency obtained in three time periods grouped by a quantum computer.

	Quantum computer	*ibmq_manila*	*ibmq_santiago*	*ibmq_quito*	*ibmq_lima*
Time period	September 2021	0.01619	0.01711	0.02480	0.02977
	April 2022	0.01914	0.01557	0.02729	0.02276
	May 2022	0.01799	0.01484	0.02725	0.02717

Once the overall quantum efficiency is analysed, in the following two subsections, we analyse the efficiency of the quantum computers used and the efficiency of the problems.

2.4.3.1. Quantum computers efficiency. As previously mentioned, in the quantum computing atmosphere, it is known that the results of a quantum computer are not always reliable. In our case, as the result or possible results of each quantum circuit are previously known, we could obtain the success rate from each quantum computer evaluated. Bear in mind once again that the success rate values are calculated by us considering the total number of shots and the correct outputs, in this case we are not considering the reliability of the gates employed nor the specific layout of the computers tested.

To obtain the efficiency of a given quantum computer, the mean of the computer for each time period was calculated independently of the problem executed. As shown in Table 10, the difference in performance is noticeable; in September 2021, the worst performance was obtained by *ibmq_manila* (0.01619), and the best performance was obtained by *ibmq_lima* (0.02977), which is a difference of 0.01358 points. In April 2022, the worst performer was *ibmq_santiago* (0.01557), and the best performer was *ibmq_quito* (0.02529), which is a difference of 0.00972 points. Finally, in May 2022, the worst performance corresponds to the quantum computer *ibmq_santiago*, (0.01484), and the best performance is once again obtained by *ibmq_quito* (0.02725), which is a difference of 0.01241 points.

These differences are quite significant and follow a similar pattern for the three time periods; therefore, we might say that, for our cases, it is best to execute the circuits on *ibmq_quito* or *ibmq_lima* as they provide a better result if the same amount of energy is to be employed as *ibmq_manila* or *ibmq_santiago*.

2.4.3.2. Quantum problem efficiency. The problem efficiency was obtained similarly to the computer's efficiency, in this case the mean of each problem, regardless of the computer, was calculated for each time period. From the values obtained (see Table 11), having a circuit with less qubits increases the performance; the *2 taxis, 3 people* problem, which only needs 3 qubits, has the highest efficiency for all three time periods, with the circuit for the *2 taxis, 5 people* version 2 the worst overall performer.

Note that a different implementation of a circuit to solve the same circuit can have a high impact on the performance of the computer, as indicated in the Introduction, in Section 1, deep circuits are more prone to reach decoherence, inconsistencies and other types of errors.

If we observe the efficiencies of both versions of the *2 taxis, 5 people* problem, the first version has a higher performance than the second version. Then, this leads us to conclude that there can be a way to improve the quantum circuits that implement a problem so that they have better efficiency, and therefore, a lower energy consumption.

2.5. General discussion on the evolution of quantum computers

As previously explained, the IBM quantum computers have been evolving and changing throughout the three time periods in which the study was executed. As already mentioned, the energy consumption of the computers selected has not changed noticeably. Let us observe again the Table 6 from Section 2.4.1 specifically if we examine the differences, *ibmq_manila* lowers a 17.7% its energy consumption from September 2021 to May 2022. *ibmq_santiago* also lowers its energy consumption, by a 3.8%. This also happens to *ibmq_quito*, which lowers its energy consumption a 2%. Finally, *ibmq_lima* has a 10.9% increase in energy consumption.

But something that has changed throughout the months included in this study is the success rate, if we observe again the Table 8 in Section 2.4.2, comparing the months of September 2021 and May 2022, *ibmq_manila* lowered its success rate by 5 percentage points. The same thing happened to *ibmq_santiago*, it decreases its success rate but in this case 13 percentage points. This computer had the best success rate of all in September 2021 but ended in May 2022 being the computer, from the ones selected for this study, with the lowest success rate.

If we look at the success rate of *ibmq_quito* we can observe that it increases 6 percentage points. And finally, *ibmq_lima* has the same success rate percentage in September 2021 and in May 2022.

From all these data, we can conclude that, in general, quantum computers have changed in the months tested, but this change has not been

Table 11
Quantum problem efficiency obtained in three time periods grouped by problem.

	Problem	*2t, 3p*	*2t, 5p friends*	*2t, 5p enemies*	*SN, (3+1)*	*SN, (4+1)*
Time period	September 2021	0.02595	0.02303	0.01926	0.02290	0.01869
	April 2022	0.02809	0.01997	0.01615	0.02130	0.01794
	May 2022	0.02907	0.02181	0.01768	0.02165	0.01886

reflected so much in their energy consumption as in their success rate, since the reliability of some of them has dropped considerably. This needs to be studied further in the future to verify these measurements and observe possible trends, a further discussion on our future work is disclosed on Section 6.

3. Energy consumption in classical computers

In this section, the measurement of the energy consumption of problems using classical computers is presented.

To carry out the energy consumption measurement, we used GSMP (Green Software Measurement Process) (Mancebo et al., 2021a), a specific process for measuring the energy consumption of software.

GSMP is the methodological component of FEETINGS (Framework for Energy Efficiency Testing to Improve eNvironmental Goals of the Software) (Mancebo et al., 2021b), which is a framework to promote reliable capture, analysis, and interpretation of software energy consumption data. FEETINGS is composed, in addition to the aforementioned methodological component, of two other components: a conceptual component (GSMO-Green Software Measurement Ontology) with terminology related to the measurement of software energy consumption and a technological component composed of EET (Energy Efficiency Tester), EETHAN and ELLIOT. EET is a validated hardware device that includes internal sensors to capture the energy consumption of the processor, the graphics card, the monitor and the hard disk, as well as the overall energy consumption of the computer (namely DUT-Device Under Test) when running software. EETHAN is a tool that helps us in the process of performing the measurements, it automates the execution of the software to be measured given the number of executions per measurement and the measurements to be performed. Finally, it collects the data captured which are then analysed using the ELLIOT v5 tool.

As in the quantum computers study, all the information about the classical computers study is available at https://github.com/GrupoAlarcos/Exploring-Trade-Off-Computational-Power-Energy-Efficiency-Quantum-Classical.

3.1. Classical implementation of the problems

The problems selected to measure their energy consumption were the same as for quantum computers, that is, *2 taxis, 3 people*; *2 taxis, 5 people* (two versions) and *the secret number* problem (two versions, *3Q+1C* and *4Q+1C*). All the problems in the study were programmed in the C programming language. We chose this programming language because, according to Pereira et al. (2017), it is one of the least consuming.

Next, in code listings 1, 2, 3 we can observe excerpts of the code in C, which implements the solution for the *2 taxis, 3 people* problem. Code listing 1 shows the declaration of the functions to be used and the main function of the problem.

```
void get_taxi_assignment(int taxi_assignment[], int
     num_passengers);
void print_taxi_assignment(int taxi_assignment[], int
     num_passengers);

void main(){
    time_t t;
    srand((unsigned) time(&t));
    int taxi_assignment[3] = {0,0,0};
    int num_passengers = sizeof(taxi_assignment)/sizeof
     (taxi_assignment[0]);

    get_taxi_assignment(taxi_assignment, num_passengers
     );
}
```

Listing 1: Declaration of the functions and the main function of the *2 taxis, 3 people* problem.

Code listing 2 shows the function that calculates a random position assignment and then verifies whether it follows the constraints defined in the problem (person 3 cannot share a taxi with any of the other people).

```
void get_taxi_assignment(int taxi_assignment[], int
     num_passengers){
    int i;

    for(i=0; i<num_passengers; i++){
        taxi_assignment[i] = rand() % 2;
    }

    // Constraint check
    if (taxi_assignment[2] != taxi_assignment[0] &&
     taxi_assignment[2] != taxi_assignment[1]){
        print_taxi_assignment(taxi_assignment,
     num_passengers);
    } else {
        get_taxi_assignment(taxi_assignment,
     num_passengers);
    }
}
```

Listing 2: A function that calculates an assignment for the passengers and checks its validity.

Code listing 3 shows a function that prints the passengers position arrangement once the solution has been reached.

```
void print_taxi_assignment(int taxi_assignment[], int
     num_passengers){
    int j,k;

    printf("\nThe taxi assignment is:");
    for (j=0; j<2; j++){
        printf("\n  -Taxi %d:", j+1);

        for (k=0; k<num_passengers; k++){
            if(taxi_assignment[k] == j){
                printf("\n   Person %d", k+1);
            }
        }
    }
}
```

Listing 3: A function that prints the final assignment of passenger taxi positions.

The other algorithms were coded in the same manner, trying to mimic the behaviour of quantum computers, obtaining results based on some parameters and then performing a constraint check.

3.1.1. Measurement protocol

As already mentioned, to carry out the energy consumption measurement, we used GSMP (Green Software Measurement Process), a specific model for measuring the energy consumption of software (Mancebo et al., 2021a). Fig. 7 shows the seven phases defined in GSMP.

Phase I. Scope Definition

First of all, the aim of the study is defined. In this case, the aim is to assess the energy consumption required to execute the problems

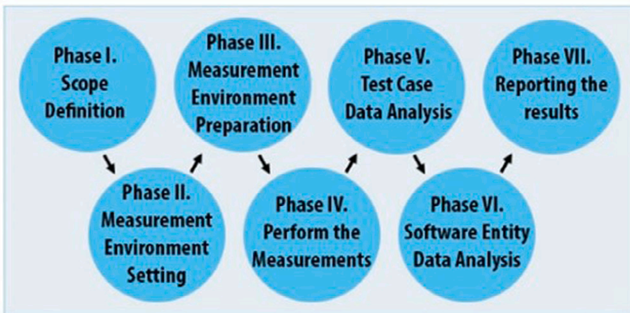


Fig. 7. GSMP phases for evaluating the energy efficiency of a software.

defined. It is necessary to define the software entity class, the software entities, and the test cases in the study, see Fig. 8.

Phase II. Measurement Environment Setting

We used FEETINGS and the measuring device EET. The Device Under Test (DUT), which is the computer where the algorithms for each test case will be executed, had the following specifications (see Fig. 7):

- Asus Prime B460 Plus motherboard.
- Intel i7 10700 2900MHz processor.
- 2 modules of 16 GB Kingston HyperX Fury DDR4 memory.
- Western Digital Blue 500 GB SSD hard disk drive.
- Saphire ATI Radeon X1950 GT, 256MB RAM DDR3 graphics card.
- Power supply 3Go PS580S – 580 W.

Although EET captures the energy consumption of several hardware components, as the objective is to compare the classical results with the quantum results, we will just retrieve the overall energy consumption (DUT consumption) and the execution time.

Phase III. Measurement Environment Preparation

We performed 35 measurements of each test case (see Fig. 8) to do a fair comparison with the quantum study and to mitigate any atypical values related to time or consumption. Moreover, as in this study the tested algorithms have very low execution times, it was required to perform an additional step to determine how many executions per measurement were needed to obtain reliable measurements. Therefore, it was determined that each algorithm should be executed 1800 times for each one of the 35 repetitions, such as detailed in https://github.com/GrupoAlarcos/Exploring-Trade-Off-Computational-Power-Energy-Efficiency-Quantum-Classical. Therefore, the results of each measurement (time and consumption values) were the average values, i.e. each one of the 35 repetition values was obtained as the average of the 1800 runs of the corresponding algorithm.

The operating system used for this study was Linux, Ubuntu. The computer was prepared in order to prevent any background executions from running so that the only thing being measured was the execution of the algorithms. Then, the EET device used to measure the energy consumption of the DUT was prepared so as to correctly collect the consumption values by the EETHAN tool.

Phase IV. Perform the measurements

This phase involves running the different algorithms that solve the problems described, on the specified DUT, following the defined protocol and taking measurements using the EET. EET has a sampling frequency of 100 Hz, that is, it captures 100 energy data points per second. The results (energy consumption values for each time point) are subsequently stored in a log that is processed in the next phase.

Phase V. Test case data analysis

The main goal of the fifth phase is to process and analyse the energy consumption data of the test cases defined in the first phase. To do this, we used ELLIOT v5, a software tool that processes the data collected by the EET.

Table 12
Execution time in miliseconds sorted from lowest (top) to highest (bottom) for the five problems proposed on the DUT, classical device under test.

Execution time (ms)	
SN (3Q+1C)	4.65317
2 taxis, 3 people	4.65321
2 taxis, 5 people enemies	4.65324
SN (4Q+1C)	4.65341
2 taxis, 5 people friends	4.65377

Table 13
Energy consumption in joules sorted from lowest (top) to highest (bottom) for the five problems proposed on the DUT, classical device under test.

Energy consumption (J)	
2 taxis, 3 people	0.55191
2 taxis, 5 people enemies	0.57610
2 taxis, 5 people friends	0.57815
SN (3Q+1C)	0.58049
SN (4Q+1C)	0.58444

First, the average values of each measurement were calculated. This is necessary when working with a single value that is derived from a large number of values stored by the EET in the log. During this process, we had to clean the data, check for possible outliers (unusual values), and remove invalid executions if any, detected by ELLIOT v5.

Next, using the average of all the power consumption values recovered by the EET and the execution time obtained from the computer clock, the consumption was calculated.

Phase VI. Software Entity Data Analysis

In this phase, the results of the energy consumption for the test cases were obtained. As shown in Table 12, all the algorithms developed using classical computing that solve the problems tested have similar execution times, between 4.65317 and 4.65377 miliseconds, which corresponds to the execution of the algorithms that solve the *SN (3Q+1)* and *2 taxis, 5 people friends*, problems respectively.

Table 13 shows the energy consumption in joules (Ws) of the algorithms performed in classical. The algorithm that consumes the least amount of energy is the *2 taxis, 3 people* problem, and the one that consumes the most is the *secret number (4Q+1C)* problem. As it can be observed there is no relationship between execution time and energy consumption. It is worth mentioning that this did not happened on the quantum study because quantum computers have a constant consumption, being then a direct relationships between time and consumption.

The consumption results allow us to conclude that there are no noticeable differences depending on whether friends or enemies are involved, although depending on the number of people differences can be found. Also the consumption of the two secret number algorithms vary slightly, being the *4Q+1C* version the most consumer one.

Therefore, we might say that different ways of solving the same problem can have an impact on the execution time, and therefore, on the energy consumption of an algorithm execution. This also happens when there is an algorithm that needs more resources from the computer.

4. Classical and quantum computing comparison

Once the consumption of both quantum and classical computers have been analysed separately, this section aims to compare the time and energy consumption required to solve a problem using a quantum computer or a classical–classical computer. Since our goal is to do the comparison with the most up-to-date versions of quantum computers, from the three different studies of quantum computers, we are going to use the results obtained in the May 2022 study.

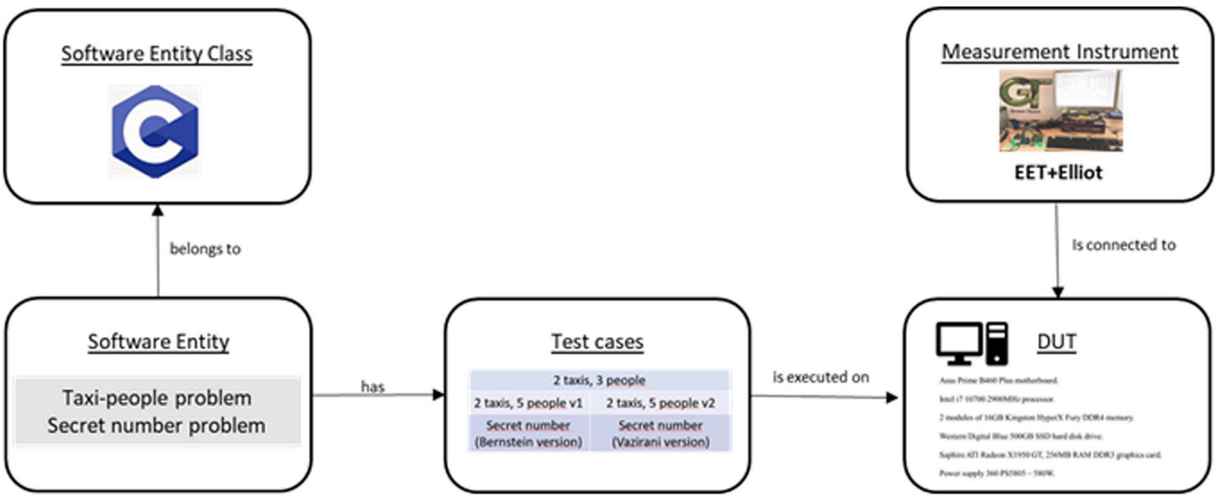


Fig. 8. GSMO instantiation for classical computers.

Table 14
Execution time in seconds, comparison for the five problems between quantum and classical computers.

Execution time (s)	Quantum				Classic
Problems	*manila*	*santiago*	*quito*	*lima*	DUT
2 taxis, 3 people	5.48	6.37	3.46	3.43	0.004653205
2 taxis, 5 people friends	5.57	6.44	4.16	4.28	0.004653774
2 taxis, 5 people enemies	5.64	6.41	4.20	4.30	0.004653241
SN (3Q+1C)	5.51	6.50	4.09	4.13	0.004653175
SN (4Q+1C)	5.63	6.45	4.08	4.10	0.004653405

Table 15
Energy consumption in Wh, comparison for the five problems between quantum and classical computers.

Energy consumption (Wh)	Quantum				Classic
Problems	*manila*	*santiago*	*quito*	*lima*	DUT
2 taxis, 3 people	38.07	44.24	24.02	23.81	0.000153308
2 taxis, 5 people friends	38.71	44.71	28.88	29.74	0.000160597
2 taxis, 5 people enemies	39.15	44.46	29.11	29.94	0.000160029
SN (3Q+1C)	38.25	45.17	28.42	28.67	0.000161246
SN (4Q+1C)	39.04	44.75	28.33	28.49	0.000162346

Table 14 shows the time execution results of the quantum and the classical computers. As can be seen, for all of the cases, the classical computer takes the shortest time to execute each of the problems tested. In fact, there is a great difference, with between 679 and 903 times more time needed for the quantum versions.

From the perspective of the energy consumption (Table 15), Quantum computers have a considerable higher consumption. For instance, the quantum computer with the lowest consumption for the easiest problem, 2 taxis 3 people, which is *ibmq_lima*, consumed roughly 155,308 times more energy than the classical computer.

In the next subsections the results are compared by problem.

4.1. PROBLEM 1 - 2 taxis, 3 people

This was one of the simplest problems included in this study. Regarding the execution time, as shown in Table 14, the classical computer takes considerably less time to execute the algorithm that solves this problem than the quantum computer to execute the circuit.

The execution time of the classical algorithm is 0.004653205 seconds, and, for the quantum computers, it varies between 3.43 s on *ibmq_lima*, and 6.37 s on *ibmq_santiago*.

Table 15 shows that quantum computers consume substantially more energy to solve all the problems. If we compare the consumption of the classical computer and the quantum computer that consumes the least amount of energy for this problem, which is *ibmq_lima*, the classical computer consumes approximately 155,308 times less energy to solve the problem.

4.2. PROBLEM 2 - 2 taxis, 5 people : both versions

Firstly, let us observe the *friends* version in Table 14. In this case, the execution time of the classical algorithm is 0.004653774 seconds. The *enemies* version takes 0.004653241 seconds to execute. Referring now to quantum computers, the fastest computer for both problems is *ibmq_quito*, which takes approximately 4.2 s and the slowest, also for both, is *ibmq_santiago*, which needs approximately 6.4 s to execute the circuits.

Looking again at the consumption, a classical computer consumes less energy to execute the algorithm that solves both problems, see Table 15.

If we compare the consumption of the classical computer for the *friends* version of this problem, the DUT (0.000160597 Wh), and the quantum computer that consumes the least amount of energy for this problem, which is *ibmq_quito*, the classical device consumes 179,828 times less energy to solve the problem. Let us now compare the *enemies* version of this problem: the consumption of the classical computer, the DUT (0.000160029 Wh), and the quantum computer that consumes the least amount of energy for this problem, which is *ibmq_quito*, the classical device consumes 181,905 times less energy to solve the problem.

4.3. PROBLEM 3 - the secret number problem: both versions

Finally, let us observe the execution time of the two last problems, which are two versions of the secret number problem.

The execution time of these two versions of the secret number problem in the classical computer is 0.004653175 seconds for the first version, *3Q+1C*, and 0.004653405 seconds for the second version, *4Q+1C*. Checking quantum computers, as before, *ibmq_quito* is the fastest to execute, taking 4.09 and 4.08 s respectively. The quantum computer *ibmq_santiago* is still the slowest computer, taking 6.5 s for the fourth problem and 6.45 s for the fifth problem. This can be seen in Table 14.

And, just as we observed before, the energy consumption of these quantum executions is substantially higher than the energy needed to execute the algorithms in a classical computer. As shown in Table 15, the quantum computer that needs less energy to execute these problems is still *ibmq_quito*, and *ibmq_santiago* remains as the slowest.

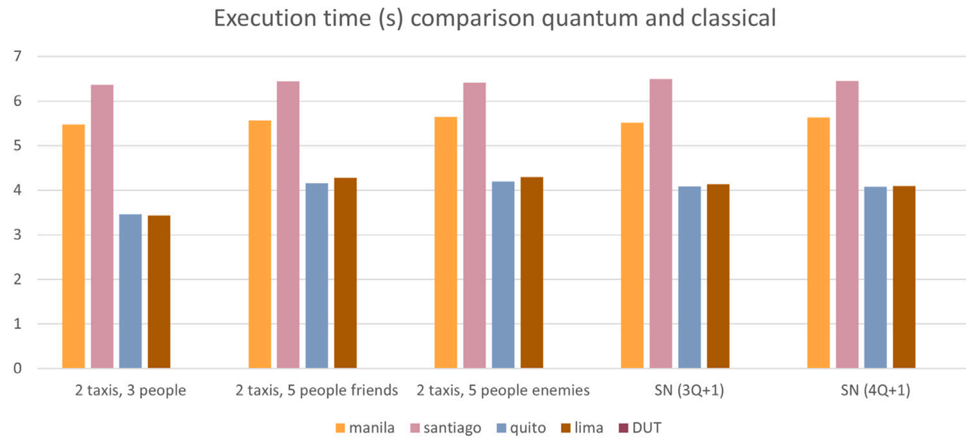


Fig. 9. Execution time comparison in seconds between quantum and classical computing executions.

If we compare the consumption of the classical computer for the first version, *3Q+1C*, of this problem, the DUT (0.000161246), and the quantum computer that consumes the least amount of energy for this problem, which is *ibmq_quito*, the classical device consumes 176,252 times less energy to solve the problem. Let us now compare for the second version, *4Q+1C*, of this problem the consumption of the classical computer, the DUT (0.000162346), and the quantum computer that consumes the least amount of energy for this problem, which is *ibmq_quito*, the classical device consumes 174,504 times less energy to solve the problem.

This also happens when there is an algorithm that needs more resources, such as more qubits or gates.

4.4. Overall comparison between classical and quantum computing

Let us compare considering both the execution time and energy consumption derived from the resolution of the problems defined in this study using classical and quantum computing. As before, if we observe the execution time obtained for all the cases and we calculate the mean of execution time of the quantum computing executions, (which is 5.0115 s), they need 107,596.38% more time to execute than the mean of execution time of the classical computing executions, (which is 0.004653360 seconds).

To perform this calculation, the following mathematical formula for the relative difference between quantum execution time and classical execution time, relative to the classical execution time (QTvsCT) was used (2):

$$QTvsCT = \frac{Quantum\ execution\ time - DUT\ execution\ time}{DUT\ execution\ time} * 100 \quad (2)$$

The individual differences of execution time in seconds for the problems described on each one of the devices tested are shown in Fig. 9. In the following figure, we are unable to discern the data obtained from the DUT owing to its infinitesimal value in comparison to data obtained from quantum computers.

Let us now take an overall look at the energy consumption of both, considering as before the mean of the data gathered, the calculation of this value for the quantum computers equals 34.798 Wh and for que DUT classical device this calculation equals 0.000159505 Wh. Comparing these results, we can see that quantum computers have, on average, 21,816,116.9% more energy consumption than the classical computer.

To perform this calculation, the following mathematical formula was used, which gives the percentage that quantum energy consumption represents in relation to the energy consumption of the DUT (QEvs.EC) (3):

$$QEvsEC = \frac{Quantum\ energy\ consumption - DUT\ energy\ consumption}{DUT\ energy\ consumption} * 100 \quad (3)$$

Individual differences in energy consumption in Wh for the problems described in each of the devices tested are shown in Fig. 10. In the following figure, we are unable to discern the data obtained from the DUT owing to its infinitesimal value in comparison to data obtained from quantum computers.

We can say that for these cases and considering the energy consumption and time needed to solve the problems described, quantum computing is not optimal. There are some cases in which the use of this technology can make sense, for example, with problems in which time is crucial and solving them on a classical computer takes far too long.

We acknowledge that with the limitations imposed by the access to this technology, that let us use quantum computers with a small amount of qubits, we could test simple quantum circuits. Nevertheless, the data and information gathered from the results of this study reinforce the idea that quantum technology must be used only for high complexity problems or some that take classical computing too much time to execute and that research on this field, relating quantum computing and energy consumption needs to be addressed.

5. Threats to validity

This section tackles the threats to the validity of the two empirical scenarios by following the recommendations in Wohlin et al. (2012):

- [•] Construct and conclusion validity. The first point is about the reliability of the measurements that we are using in the two empirical scenarios. As for the quantum one, to calculate the consumption of the quantum circuits we used the data provided in the literature about the constant consumption of IBM Quantum computers. Therefore, we trust on the data source and although we cannot guarantee that this figure is totally reliable, the obtained conclusions can still be considered valid to rank the computers and problems with better or worse energy behaviour. This is due to the fact that consumption values would be altered equally (energy value affects equally all the results as it is used as a constant value in quantum scenario). In the case of the comparison between classical and quantum, if the data change, the comparison would also change but as there was a big difference between the two paradigms in our experiment it is not likely that the conclusions would vary. In the classical scenario we used EET to measure consumption, which is a hardware device that obtains real consumption values; it has been validated (Mancebo et al., 2021b) and previously used in other measurements of this type.
- [•] Internal validity. Regarding uncontrolled factors that may affect the results of the experiment, the most remarkable are the ones related with the conditions in which the executions

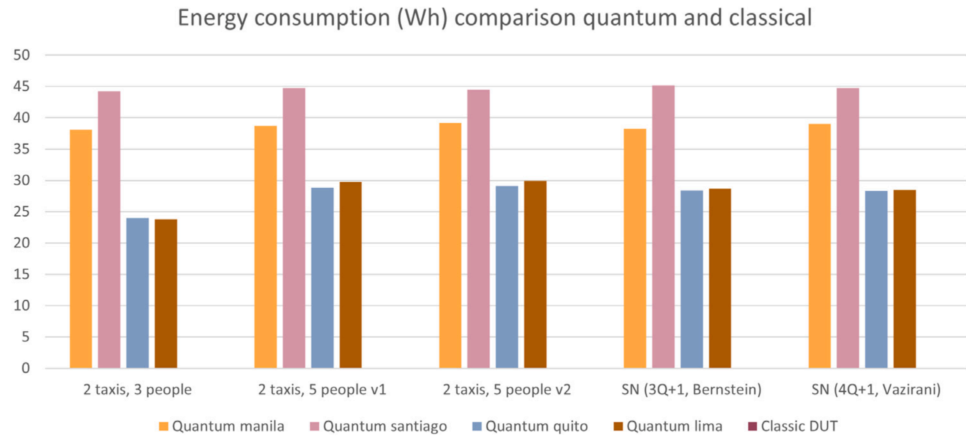


Fig. 10. Energy consumption comparison in Wh between quantum and classical computing executions.

were performed. First, the problems to be solved had similar implementations in both the quantum and classical scenarios. On the other hand, as previously stated, several executions (35) were performed in both scenarios to mitigate the possible atypical values related to time or consumption. Due to the low execution time of the algorithms in the classical scenario, 1800 repetitions per measurement have been carried out to obtain more accurate results. As for the DUT used in the classical scenario, we used a computer with characteristics similar to those that can be purchased in the market and measures were taken to ensure that the DUT was always in the same conditions for the different executions.

- [•] Threats to external validity. Finally, regarding the potential for generalisation of the results obtained in this experiment, it is important to note that the main objective of this work is to give a first indication of the need for an adequate trade-off between computing power and energy when making the decision to use quantum or classical computing. In this sense, the external validity of this study is strongly affected by the availability of quantum computers, as we discussed earlier, the access to this computers is limited as you can employ most of the times just quantum simulators and in some cases quantum computers with a small amount of qubits, this limited the complexity of the problems included in our study. Which were three specific problems (*2 taxis, 3 people*; *2 taxis, 5 people*; *the secret number* problem), we selected them considering the computers we had access to, which had a maximum of 5 qubits. This limitation highly restricted the problems we could design, execute and measure, but even with this limitation we could gather relevant conclusions as we could also observe the differences obtained with distinct depth circuits. There are also other quantum technologies besides gate-based quantum computing, such as quantum annealing, that are planned to be analysed in the future. Therefore, we can consider this study as a starting point for future research in this field.

6. Conclusions and future work

When conducting research, it is important to gather empirical evidence in support of the ideas which guide the research. This work is the first step towards that direction, it is one of the first studies that provides empirical evidence on the high energy consumption that comes with quantum computing and laying the foundation for future investigations.

In this laboratory experiment we analysed two aspects; we compared the quantum computers individually to determine whether the energy consumption employed, and the correctness of the results obtained change over time; and then we compared both quantum and classical computers to determine whether one or the other consumes more energy to solve the same problem.

For the simple problems tested, we can conclude that it appears that quantum computers are not the solution as classical devices can solve them in a shorter time, with less energy efficiency and a better correctness. As seen throughout this study, quantum computers do not provide a correct solution every time the circuit is executed. In this case, the quantum utility (Herrmann et al., 2023) is reduced, as the classical advantage is much higher. Therefore, considering all of the aforementioned factors, quantum computers can be a good solution when implementing the solution on a classical device is either not cost efficient because it takes a lot of time to run, or when this is even impossible to execute.

An interesting result of the quantum study is that, for the same problem, different circuits appear to vary in time and thus in consumption. This reinforces our idea that it is possible to define circuits considering the energy consumption that their execution will have (as in classical computing).

One of the problems with quantum computers is that IBM Quantum computers and their processors and features are constantly being updated as an emerging technology. Consequently, they do not always take the same amount of time to execute a quantum circuit, which is reflected in their energy consumption. It is then necessary to execute a circuit several times to have a better dimension of the real time it requires to be executed. However, there seems to be a relationship between the volume of the quantum computer and the power consumption needed to execute a circuit: the larger the volume, the higher the consumption. However, we did not find a pattern that allows us to identify a better quantum computer from its success rate, which can be interpreted as the fact that the success rate is a problem present in all quantum computers and almost at the same value (the maximum difference in our study is 10%). We have to keep in mind again that we are talking about the success rates obtained from the executions we performed and we are not considering the reliability of the gates themselves nor the specific layout of the quantum computers employed, we will address this topic in the future work, see Section 6.

Among the quantum computers used in the study, it seems that it is best to execute the circuits on *ibmq_quito* or *ibmq_lima*, since they provide a better result if equal amount of energy is used compared to *ibmq_manila* or *ibmq_santiago*.

From the evolution point of view, we can conclude that, overall, quantum computers changed in the months tested, but this change was not reflected so much in energy consumption as in the success rate.

In general, from the perspective of energy efficiency for simple problems, the energy consumption of the analysed quantum computers is significantly higher than that of the classical DUT computer, as shown in Section 5. This indicates that when given the choice, it would be better to continue using traditional computers for these types of problems. However, it is important to note that quantum computers have the potential to be much more powerful than classical computers for certain types of tasks. For example, they can solve certain problems that are impractical or impossible to solve on classical computers. As quantum computers continue to advance and become more energy-efficient, they may eventually become the preferred option for a wider range of problems. However, at present, the higher energy consumption and variable execution times of quantum computers means that for many simple problems, classical computers may still be the better choice.

To continue the work and research in this field, it would be advisable for all quantum computing companies to disclose the energy consumption of their devices so that studies can be conducted with the goal of improving the energy efficiency of quantum software, which would ultimately lead to more energy-efficient quantum computers. This would allow quantum software to be built from the outset with a view to sustainability and energy efficiency.

We would like to repeat the study in the future but including some differences.

The current study was limited to the quantum computers that were available for our use in the IBM Quantum platform, which, by that time, had a maximum of 5 qubits; currently normal users can access 127 qubit computers. Therefore, this study could be replicated using those and implementing circuits that solve problems that require that number of qubits and in general, test problems with a higher complexity. This study could also be replicated adding the use of quantum simulators as well as some types of quantum programming other than quantum gate based (for example, quantum annealing) in order to compare them, check their differences and their performance in terms of execution time and energy consumption.

We would also like to conduct a study to test whether there is an implication of the number of qubits as well as the gates used (or other elements) to develop a circuit, and the success rate, time, energy consumption, and others, obtained upon its execution.

CRediT authorship contribution statement

Elena Desdentado: Writing – review & editing, Writing – original draft, Investigation, Data curation. **Coral Calero:** Writing – review & editing, Validation, Supervision. **Mª Ángeles Moraga:** Writing – review & editing, Validation, Supervision. **Manuel Serrano:** Writing – review & editing, Validation, Supervision. **Félix García:** Writing – review & editing, Validation, Supervision.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: All authors report financial support was provided by Junta de Comunidades de Castilla-La Mancha. All authors report financial support was provided by Spain Ministry of Science and Innovation. All authors report financial support was provided by FEDER funds (Fondo Europeo de Desarrollo Regional). All authors report financial support was provided by the European Union NextGenerationEU/PRTR If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

I have shared the link to the data in the manuscript.

Acknowledgements

This work is part of the following projects:

- OASSIS: PID2021-122554OB-C31/ AEI/ 10.13039/5011000 11033/ FEDER, UE.
- SEEAT: This work is part of the SEEAT project (PDC2022-133249-C31), funded by MICIU/AEI /10.13039/501100011033 and by the European Union NextGenerationEU/ PRTR.
- PLAGEMIS: This work is part of the PLAGEMIS project (TED2021-129245B-C22), funded by MICIU/AEI /10.13039/501100011033 and the European Union NextGenerationEU/ PRTR.
- EMMA: Project SBPLY/21/180501/000115, funded by CECD (JCCM) and FEDER funds.

In addition, the first author has a pre-doctoral FPI (Research Staff Training) contract at the University of Castilla-La Mancha, Spain, for which funding has been received from the European Union through the European Social Fund Plus (ESF+).

References

Anon, 0000a. Universities with research groups - Quantum computing report, Quantum Computing Report By CGI, URL https://quantumcomputingreport.com/universities/.

Anon, 0000b. The talavera manifesto for quantum software engineering and programming, aQuantum software engineering, URL https://www.aquantum.es/manifesto/.

Anon, 0000c. qiskit-textbook/content/ch-algorithms/bernstein-vazirani.ipynb at main · qiskit-community/qiskit-textbook, GitHub, URL https://github.com/qiskit-community/qiskit-textbook/blob/main/content/ch-algorithms/bernstein-vazirani.ipynb.

Anon, 0000d. Operations glossary, IBM Quantum, URL https://quantum-computing.ibm.com/composer/docs/iqx/operations_glossary.

Anon, 2018. Between 10 and 20% of electricity consumption from the ICT sector in 2030? Enerdata, URL https://www.enerdata.net/publications/executive-briefing/between-10-and-20-electricity-consumption-ict-sector-2030.html.

Anon, 2021. What is quantum volume, anyway? Qiskit, URL https://quantum-computing.ibm.com/composer/docs/iqx/operations_glossary.

Anon, 2022. Quantum computing promises to solve data center energy drain. VentureBeat, URL https://venturebeat.com/datadecisionmakers/quantum-computing-promises-to-solve-data-center-energy-drain/.

Bernstein, E., Vazirani, U., 1997. Quantum complexity theory. SIAM J. Comput. 26 (5), 1411–1473. http://dx.doi.org/10.1137/S0097539796300921.

Boger, Y., 2023. The dual-pronged energy-saving potential of quantum computers. URL https://www.datacenterdynamics.com/en/opinions/the-dual-pronged-energy-saving-potential-of-quantum-computers.

Castro, A.d., 2023. Quantum computing in the roadmap to greener calculations. URL https://medium.com/pasqal-io/quantum-computing-in-the-roadmap-to-greener-calculations-6adf15478509.

Deutsch, D., Jozsa, R., 1992. Rapid solution of problems by quantum computation. Proc. Royal Soc. A 439 (1907), 553–558. http://dx.doi.org/10.1098/rspa.1992.0167, URL https://royalsocietypublishing.org/doi/abs/10.1098/rspa.1992.0167.

Dowling Jonathan, P., Milburn Gerard, J., 2003. Quantum technology: the second quantum revolution. Royal Soc. 361 (1809), http://dx.doi.org/10.1098/rsta.2003.1227.

Ezratty, O., 2022. Understanding quantum technologies 2022. Opinions Libres - Le blog d'Olivier Ezratty, URL https://www.oezratty.net/wordpress/2022/understanding-quantum-technologies-2022/.

Fellous-Asiani, M., 2022. Could energy efficiency be quantum computers' greatest strength yet? The Conversation, URL http://theconversation.com/could-energy-efficiency-be-quantum-computers-greatest-strength-yet-191989.

Fernández, E.D., de la Rubia, M.Á.M., Martín, M.S., 2021. Studying the consumption of IBM quantum computers. In: International Conference on Information Systems Development. ISD, URL https://aisel.aisnet.org/isd2014/proceedings2021/sustainable/3.

Gutiérrez, J.R., 2020. La segunda revolución cuántica y cómo va a impactarnos. Delfino, URL https://delfino.cr/2020/09/la-segunda-revolucion-cuantica-y-como-va-a-impactarnos.

Herrmann, N., et al., 2023. Quantum utility – definition and assessment of a practical quantum advantage. http://dx.doi.org/10.1007/10.48550/arXiv.2303.02138, arXiv.

Hsu, J., 2015. How much power will quantum computing need? IEEE Spectrum, URL https://spectrum.ieee.org/how-much-power-will-quantum-computing-need.

Hundt, R., 2022. Quantum computing for programmers. Cambridge CB2 8BS, URL https://www.cambridge.org/us/academic/subjects/computer-science/algorithmics-complexity-computer-algebra-and-computational-g/quantum-computing-programmers?format=HB.

Ikonen, J., Salmilehto, J., Möttönen, M., 2017. Energy-efficient quantum computing. npj Quantum Inf. 3 (1), http://dx.doi.org/10.1038/s41534-017-0015-5.

Mancebo, J., Calero, C., García, F., 2021a. GSMP: Green software measurement process. In: Calero, C., Moraga, M.A., Piattini, M. (Eds.), Software Sustainability. Springer International Publishing, Cham, pp. 43–67, URL https://link.springer.com/chapter/10.1007/978-3-030-69970-3_3.

Mancebo, J., et al., 2021b. FEETINGS: Framework for energy efficiency testing to improve environmental goal of the software. Sustain. Comput. Inf. Syst. 30, http://dx.doi.org/10.1016/j.suscom.2021.100558.

Meier, F., Yamasaki, H., 2023. Energy-consumption advantage of quantum computation. http://dx.doi.org/10.48550/arXiv.2305.11212, arXiv:2305.11212.

Pereira, R., et al., 2017. Energy efficiency across programming languages: how do energy, time, and memory relate? In: Proceedings of the 10th ACM SIGPLAN International Conference on Software Language Engineering. Association for Computing Machinery, New York, NY, USA, pp. 256–267. http://dx.doi.org/10.1145/3136014.3136031.

Preskill, J., 2018. Quantum Computing in the NISQ era and beyond. Quantum 2, 79. http://dx.doi.org/10.22331/q-2018-08-06-79.

Serrano, M.A., Pérez-Castillo, R., Piattini, M., 2022. Quantum Software Engineering. SpringerLink, URL https://link.springer.com/book/10.1007/978-3-031-05324-5.

Stol, K.-J., Fitzgerald, B., 2018. The ABC of software engineering research? ACM Trans. Softw. Eng. Methodol. 27 (3), 1–51. http://dx.doi.org/10.1145/3241743.

Sugi, Y., 0000. What is a quantum computer? Explain with a simple example - Programmer Sought, ProgrammerSought, URL https://programmersought.com/article/62945919474/.

Swayne, M., 2022. 18 Leading quantum computing research institutions in 2022. The Quantum Insider, URL https://thequantuminsider.com/2022/05/16/quantum-research/.

Wohlin, C., et al., 2012. Experimentation in Software Engineering. Springer, Heidelberg, http://dx.doi.org/10.1007/978-3-642-29044-2.

Elena Desdentado is a Ph.D. student at the Escuela Superior de Ingeniería Informática, University of Castilla-La Mancha, Spain. She is a member of the Alarcos Research Group, and her research interests include software sustainability and quantum computing. Elena.Desdentado@alu.uclm.es/Elena.DFernandez@uclm.es.

Coral Calero is full professor, member of the Alarcos Research Group at UCLM Director of the Green Quantum Algorithms & Software area in aQuantum and Director of the Green Algorithms area of OdiseIA (Observatory on the social and ethical impact of artificial intelligence). She works on software sustainability.
Coral.Calero@uclm.es

Mª Ángeles Moraga is an associate professor and member of the Alarcos Research Group at UCLM. She works on software quality and measures, and software sustainability.
MariaAngeles.Moraga@uclm.es

Manuel A. Serrano is M.Sc. and Ph.D. in Computer Science by the University of Castilla-La Mancha. He is working on cybersecurity, quantum computing, data quality, software quality and measurement, and business intelligence.
Manuel.Serrano@uclm.es

Félix García is full professor at UCLM. He is a member of the Alarcos Research Group, and his research interests include software sustainability, business process management, software processes, and software measurement.
Felix.Garcia@uclm.es