Battery Network Design Using Quantum Computing

Cristina Barragán Castro

Universidad Pontificia Comillas ICAI, Madrid and *University of Strathclyde, Glasgow*

***Abstract* — Energy storage systems will play a big part in increasing grid flexibility, so having tools that allow for quickly and efficiently designing battery networks can be very useful. Starting from the code developed for project Flexener, which uses genetic algorithms to design a battery network that maximises voltage control and reliability while minimising the needed investment, the possibility of translating it into quantum computing is analysed. After carrying out a scalability analysis, it is concluded that the optimisation is the most complex part of the algorithm, representing 56% of the total computation time. Therefore, this is where quantum computing could be most beneficial. The technology chosen to solve this problem is a Quantum Annealer, which works with a specific formulation: QUBO. Therefore, the problem needs to be reformulated. Two approaches are studied: a quantitative one, which translates into QUBO the original problem, and a heuristic approach which uses strategies such as eigencentrality, energy communities and longest and shortest path problems. The translation into quantum computing is expected to allow for working with bigger networks and for a potential speed-up.**

***Index Terms* — Optimisation, Battery Network, Genetic Algorithm, Quantum Computing, Quantum Annealer, QUBO**

I. INTRODUCTION

The increasing penetration of distributed generation in the distribution grid, particularly of renewable energies, and the growth in the total demand due to the electrification of the economy imply great challenges for grid operation and planning, The introduction of energy storage systems, and in particular batteries, into the power grid brings with it an improvement in flexibility, which is essential to balance generation and demand under uncertainty. This way, they allow for dealing with the variability of renewable generation, charging during peak generation and valley demand periods, and injecting power into the grid in opposite situations. This facilitates the integration of renewable energies into the grid. Moreover, batteries bring many other advantages, such as the possibility of acting as backup generators in case of fault due to their rapid response capabilities, which improves the system's reliability. This is especially important for radial systems, which have a low reconfiguration capacity and, therefore, find it harder to guarantee the power supply. For this project, the improvement in reliability and voltage control capabilities are the most relevant. Voltage control is related to batteries' ability to control their active and reactive power exchange with the system [1]. On the other hand, the improvement in reliability is related to their ability to feed isolated systems.

All these capabilities, combined with the decrease in prices experienced in recent years, make these energy storage systems a very attractive option for investing in the development of the power grid. For this reason, it would be extremely useful to be able to quickly and efficiently design battery networks to be introduced into the distribution grid. Nevertheless, the great number of variables and possible locations for the batteries make this a very computationally complex problem. That is why this paper analyses its implementation using quantum computing. This technology has a huge potential in the energy sector, nevertheless, it is still relatively new and this kind of project can help boost its introduction into the power sector.

This paper starts by briefly explaining the problem treated in Project Flexener and how it was solved. Then, Section III includes an analysis of the computation time as well as a scalability analysis with the number of zones, which allow for identifying which parts of the algorithm are more computationally complex and, therefore, would be the most benefited with the implementation in quantum computing. Next, section IV presents some basic concepts regarding quantum computing, as well as the different technologies available. The most suitable one for solving this problem is then chosen in Section V, as well as the formulation needed to implement this technology. Next, the original problem is reformulated to adapt it to said formulation, which is done in section VI. This is part of the quantitative approach to solving the problem, as explained in section VII. This section presents the two approaches that can be followed to translate this problem into quantum computing. Finally, sections VIII and IX include the conclusions drawn from this project as well as future lines of work, as the problem is yet to be implemented into quantum computing.

II. PROJECT FLEXENER T.3.1.2

The present paper follows project Flexener, more specifically task T.3.1.2, developed by Universidad Pontificia Comillas in close collaboration with i-DE [2]. The information presented next is a compilation of the different project reports, as well as from carrying out reverse engineering on the code developed for the project.

The objective of task T.3.1.2 is to determine the optimum capacity and location of batteries in the grid to meet three objectives: maximising voltage control under normal operation, maximising reliability, and minimising investment. In this case, the optimisation problem is solved by using genetic algorithms. A 20kV radial network with distributed generation and a total of 263 buses, which is shown in the following figure, is used for the analysis.

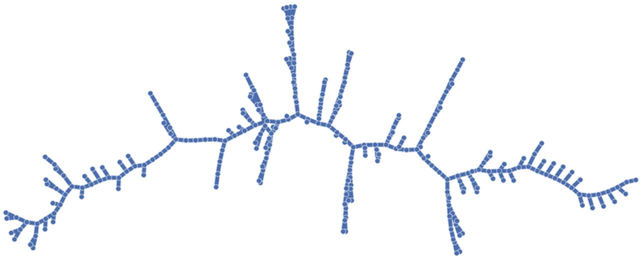


Fig. 1: Network Topology

To solve this problem, a MATLAB algorithm is developed, which can be divided into three parts: input data pre-processing, the optimisation, and post-processing.

Starting with the pre-processing, this step could itself be divided into three main activities, the first being loading the data into the system and preparing it for its later use. This means, transforming it to inner format and rearranging it, to avoid errors when using matpower and facilitate later calculations. Next, the network is divided into zones attending to the placement of tele-controlled switches. For each of these zones, several parameters are calculated as they will be needed later to carry out the optimisation. The following figure shows the zone distribution obtained:

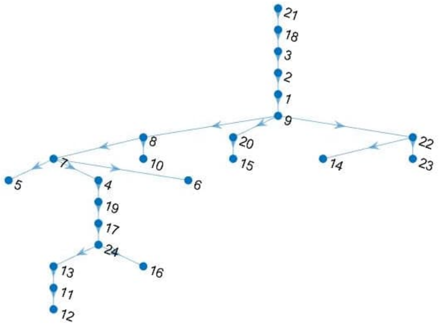


Fig. 2. Simplified Network using Zone Division

On another hand, in case of a fault in one of the zones, said zone will be isolated and the rest will reconfigure forming microgrids in order to guarantee the continuity of supply. The aforementioned microgrids are also identified in this step for every possible faulted zone. Lastly, as voltage control directly depends on the specific node in which batteries are placed, the node where the impact of placing the batteries will be greater regarding balancing voltages is identified.

As for the multiobjective optimisation, it is carried out at a zone level to facilitate the convergence of the genetic algorithm. A continuous optimisation variable which represents which battery capacity is installed in each zone is used. It will be limited by two constraints: one regarding the maximum capacity of one battery and the other one regarding the maximum total capacity that can be installed. The latter is important because voltage control and reliability tend to improve by installing batteries, but there is a maximum investment the company is willing to make. Therefore, this is a way of limiting the investment.

There are three objective functions:

- **Maximising Voltage Control** during normal operation: The following metric is used, which represents the voltage improvement obtained by installing a certain battery capacity:

$$V_{imp-Zona} = abs\left(sum\left(M_{opt}(:,n)\right)\right) * C_z$$

$$V_{imp-Total} = \sum_{zonas} V_{imp-Zona}$$

Eq. 1: Voltage Control Metric

M_{opt} is the sensitivity matrix, which represents how voltage is affected by a variation in the active and reactive power injection.

- **Maximising Reliability:** It is expected to be the most computationally complex function of all. It is modelled as the minimisation of the VOLL, which represents the economic impact of the fault. The metric used is the next, being b2 the faulted zone and b1 the analysed zone:

$$Metric\_Reliab = \sum_{b1=1}^{b1=B} \sum_{b2=1}^{b2=B} fr_{b2} * h\,rep_{b2} * IE_{b1} * ZI_{b1,b2}^{ns}$$

$$ZI_{b1,b2} = 1 - \frac{C + \sum PGmed_Z * hrep}{\sum PDmed_Z * hrep} = 1 - \frac{C}{\sum ENS_Z}$$

Eq. 2: Reliability Metric

As can be seen, for every faulted zone it takes into account the fault rate in that zone, the reparation time, how much it affects the rest of the zones (represented as matrix ZI) and the economic impact it has on them. ZI is an interaction matrix which represents how much of the demand in zone b1 is not going to be satisfied because of a fault in b2. Therefore, the microgrids formed for every fault need to be taken into consideration when calculating this matrix. Moreover, it depends on the optimisation variable, so it needs to be recalculated for every iteration. This is the main source of complexity in this function.

- **Minimising the Investment:** It is modelled taking into account economies of scale, which produces the following metric:

$$I = \frac{unitary\ cost}{cap - 2 \cdot emax} \cdot x^2 - \frac{2 \cdot emax \cdot unitary\ cost}{cap - 2 \cdot emax} \cdot x$$

Eq. 3 Investment Metric

It represents the total cost of installing the batteries.

Lastly, regarding post-processing, genetic algorithms are metaheuristic. This means that, when solving the optimisation, they do not guarantee optimality. They produce reasonable solutions instead. This way, the algorithm does not give a unique solution, but rather a number of them, which are presented in two ways: using a graph that represents the impact on all three metrics of increasing the total capacity installed, and a matrix that contains the values of the optimisation variable for each solution. This way, by looking at the tendencies in both graphs, the user can decide on the final design option.

In the analysed case, the batteries tend to be placed at zone 12, as it is the furthest from the main feeder and, therefore, the most likely to suffer a loss of supply. Moreover, it has one of the highest VOLL and, for this reason, it has a great impact on the total reliability metric.

III. COMPUTATION TIME ANALYSIS AND SCALABILITY

Computation time and scalability of the algorithm developed for project Flexener have been analysed in order to determine which parts are the most computationally complex, and therefore, would experience greater benefits from the introduction of quantum computing.

With the computation time analysis carried out it is determined that, for a model that includes power flow calculations, 56% of the total computation time is used for the optimisation, while 43% is used for the pre-processing. On another hand, in the next figure can be seen that two functions are more complex than the rest. These are the power flow, which belongs to the pre-processing part of the algorithm, and the genetic algorithm itself:

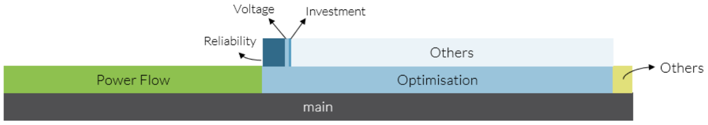


Fig. 3: Computation Time Distribution of Project Flexener's Algorithm

Regarding the genetic algorithm, a big part of the computation time is dedicated to internal functions. Among the different objective functions, the most complex one is reliability, as expected. This is ascertained with the scalability analysis carried out at a zone level, whose results are presented in Fig. 4. Due to a lack of data, it has not been possible to perform a scalability analysis at a bus level. Therefore, it was not possible to determine the growth rate of the power flow. Nevertheless, it is estimated to be lower than the one for the optimisation.

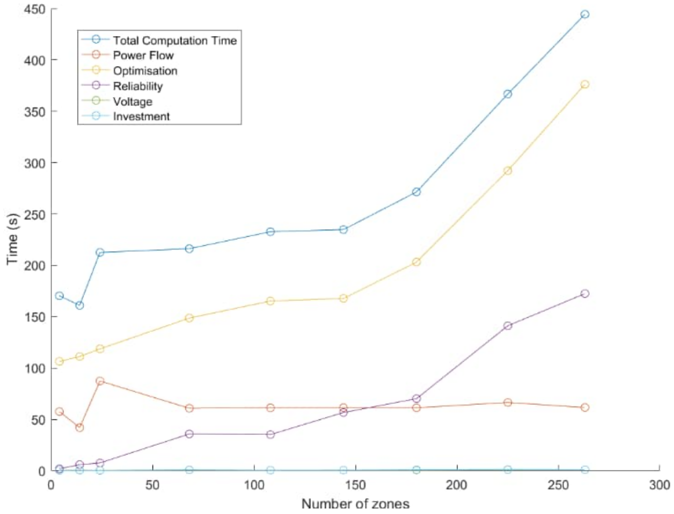


Fig. 4: Scalability Analysis of the Algorithm developed for project Flexener at a Zone Level

It is concluded that the growth in the total computation time is directly related to the growth in the time needed for the optimisation and, in particular, for the reliability function. The voltage control and investment functions continue to represent a very low proportion of the total computation time when increasing the network size.

To know which function the growth rate of the total computation time follows, it is approximated using both a cubic and an exponential function, represented in green and orange respectively in the next figure:

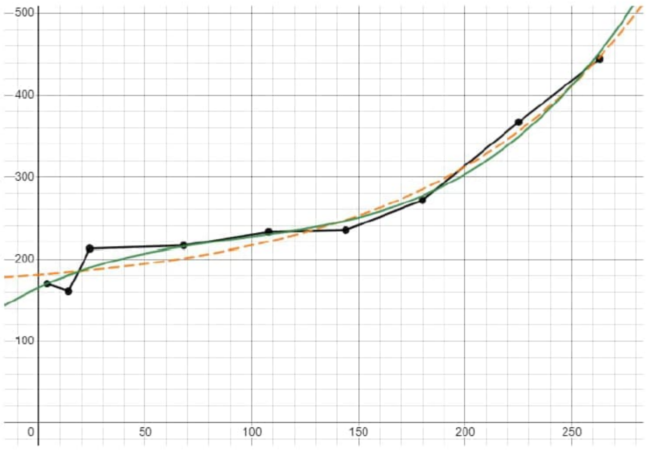


Fig. 5. Growth Rate Estimation of the Total Computation Time

The R^2 obtained for the cubic is 0.9788, and 0.968 for the exponential. Therefore, it seems to grow with a cubic function. However, as the number of points used for the analysis is not very high and genetic algorithms are inherently chaotic, it is hard to approximate the growth rate with a particular function with complete certainty. This way, it is concluded the growth rate of the total computation time must be at least cubic, which is the growth rate of the reliability function.

In conclusion, there are two functions that can be interesting to translate into quantum computing: the power flow and the optimisation. However, current quantum technologies do not allow for the calculation of power flows for large networks, so for the moment it must stay in classical computation. Nevertheless, studies show promising results on this matter and quantum computing could be of use when the technology is more mature. This way, this project focuses on the optimisation.

IV. QUANTUM COMPUTING

Quantum computing is an emerging technology used to solve problems that are too complex for classical computing. This way, they can be extremely useful for power grids, where large and complex optimisation problems with numerous input and output variables are very frequent, each time more and more as the number of DERs increases and the power flow stops being unidirectional.

Quantum computers work by exploiting the laws of quantum mechanics, like superposition, entanglement, and interference, which are explained next [3].

A. Basic Concepts

Qubits are the basic unit of information in quantum computing. While bits can be either 0 or 1 but never both, qubits allow for the superposition of both states [4].

On the other hand, entanglement is the ability of qubits to influence each other, forming a single system. In this case, one qubit's quantum state cannot be described independently of the other's [4]. This way, one qubit is directly affected by changes in another. These relationships are exploited by quantum algorithms to solve complex problems [5].

Finally, interference is related to superposition and refers to the capability of qubits to affect the likelihood of it collapsing in one of the two possible ways. This means qubits lose their quantum state to act like a bit. Quantum computers are designed to reduce interference as much as possible to guarantee good and accurate results [4].

B. Quantum Computing Technologies

There are many quantum technologies, but there are two whose use is more widespread:

- **Gate-Based Quantum Computing:** This kind of computer is characterised by its ability to control the evolution of quantum states [8]. This way, qubit states are initialised according to the input data [7] and then modified using a series of gate operations. They allow for solving a wide variety of problems. However, they are very sensitive to errors like noise and decoherence, and they have a low number of qubits [7].
- **Quantum Annealing:** They take advantage of the fact that every system tends to seek a minimum energy state [9]. This way, they allow for solving sampling and optimisation problems. They work by initialising qubits to a ground state of a trivial Hamiltonian [7]. The system's configuration is then modified so that the energy landscape reflects the problem [6]. This way, the system then evolves to the minimum energy state which represents the solution to the problem [7]. Some of their main advantages are their larger number of qubits and their lower sensitivity to errors, compared to Quantum Gate Computers [6].

Moreover, sometimes it can be useful to have a quantum and a classical computer working together. This is the case of **Hybrid Computing** [10]. This method takes advantage of both the flexibility of classical computing and the ability of quantum computers to solve complex problems [11][8]. In this sense, quantum computers perform hard tasks, and classical computers deal with the rest, such as HMI and high-level applications.

Finally, as an alternative to using quantum computers, there are **Quantum-Inspired Technologies**. These can solve specific problems by taking advantage of the mathematical formulation of quantum mechanics. This way, they simulate the behaviour of quantum computers using classical ones [12]. However, they still have the same limitations as classical computers regarding memory and processing speed [13].

V. CHOICE OF QUANTUM TECHNOLOGY AND FORMULATION

Several quantum technologies allow for solving optimisation problems.

Firstly, there are Quantum Gate Computers, which would use a QAOA algorithm to solve the problem. QAOA is a heuristic algorithm that belongs to the group of Hybrid-Computing [14]. It is designed to use QUBO formulation and can solve combinatorial optimisation problems by generating approximate solutions [15]. This option is discarded as this type of computer is very sensitive to noise, which may compromise the optimality of the solution. Moreover, they do not have a great number of qubits, so they cannot work with large networks.

Another option would be using a Quantum-Inspired solution. The algorithm that would be used is QIOA, which is an analogue of QAOA for this kind of computer [14]. However, this algorithm is not very mature yet and, furthermore, requires a large number of resources. This way, it may have problems working with large networks.

For this reason, a Quantum Annealer is the chosen technology. This is a type of quantum computer that is specifically designed for solving sampling and optimisation problems, as is the one object of study. Moreover, it has a lower sensitivity to noise than Quantum Gate Computers, and a higher number of qubits, which allows for working with large networks. To use this kind of computer, it is necessary to use a specific formulation: QUBO (Quadratic Unconstrained Binary Optimisation) [16]. Therefore, the problem must be reformulated to be expressed as:

$$max/min\ y = x^t \cdot Q \cdot x$$

Eq. 4. QUBO Formulation

X is the optimisation variable, which in this case represents whether a certain battery capacity is installed in a zone or not. Therefore, its dimensions will be N·C x 1, being N the number of zones and C the number of battery capacities analysed. Regarding matrix Q, it is a constant square matrix representing the problem. For the studied problem, a sub-matrix of dimensions N x N will be needed for each battery capacity. They will be the main diagonal of the Q matrix. The off-diagonal elements will represent the interaction between different battery capacities. This is presented in the following graph:

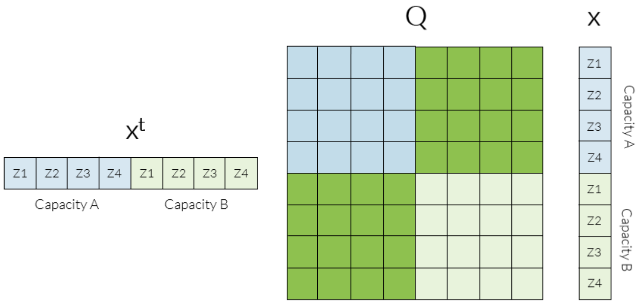


Fig. 6. Representation of the QUBO Adaptation of the Problem

VI. PROBLEM REFORMULATION

As its name indicates, to formulate the problem as a QUBO the optimisation variable must be binary and to the maximum power of 2. For this case, as previously explained, the binary optimisation variable is going to represent whether a certain battery capacity previously set is installed in a zone or not. This way, one of the constraints is no longer needed, as the battery capacities are set by the user. The constraint regarding the total battery capacity installed must be included in the objective function. It is important to note this will not be a hard constraint, therefore, its importance can be modified by changing the parameter d. Moreover, the genetic algorithm is let go and all objective functions are joined into one. When doing this it is important to include a set of parameters a, b and c which represent the importance of each of them, while allowing for unifying the units.

With all of this and using classical formulation, the problem is reformulated:

- **Indexes**

i	Zones (b1= analysed zones, b2=faulted zone)
mg	Microgrids
j	Battery Capacity

- **Parameters**

M	Maximum Installed Battery Capacity
fr_{b2}	Fault rate in zone b2
$hrep_{b2}$	Reparation time for zone b2
U	Unitary Cost of the Batteries
cap_c	Battery capacity at which functions with and without economies of scale cross.
Z_3_i	Zonal table column 3. VOLL for zone i.
Z_8_i	Zonal table column 8. Value of voltage improvement for zone i
Z_9_i	Zonal table column 9. Total ENS in zone i.
Z_10_i	Zonal table column 10. Total generation capacity (not from the batteries) installed in zone i

$MG_{b1,b2,mg} \in \{0,1\}\ \ \forall$ b1, b2, mg

If b1∈mg for fault in b2 ⇒ MG=1

aux ∈ {0,1}

If ENS=0 ⇒ aux=1

G_{b2}	Number of microgrids for a fault in b2
B_j	Battery Capacity j

- **Variables**

$x_{i,j} \in \{0,1\}\ \ \forall$ i, j

$x_{i,j}$ =1 if Battery Capacity j is installed in zone i

s	Slack Variable. Later expanded to a set of binary variables

- **Objective Function:**

$$min\ a \cdot V(x_{I,j}) + b \cdot I(x_{I,j}) + c \cdot VOLL(x_{I,j}) + d \cdot Const(x_{I,j})$$

$$V(x_{i,j}) = -\sum_j \sum_i Z\_8_i \cdot B_j \cdot x_{i,j}$$

$$I(x_{i,j}) = \sum_j \sum_i \frac{U}{(cap\_c - 2 \cdot M)} \cdot x_{i,j} \cdot (B_j^{\,2} - 2 \cdot M \cdot B_j)$$

$$VOLL(x_{i,j}) = \sum_j \sum_{b2=1}^{i} \sum_{mg=1}^{G_{b2}} fr_{b2} \cdot hrep_{b2} \cdot IE_{b2,mg} \cdot ZI_{b2,mg,j}$$

$$ZI_{b2,mg,j} = \left(1 - \frac{C_{b2,mg,j}}{Z\_9_{b2} + aux}\right) \cdot (1 - aux) \qquad \forall\, b2, mg, j$$

$$C_{b2,mg,j} = \sum_{b1=1}^{i} MG_{b1,b2,mg} \cdot (Z\_10_{b1} + B_j \cdot x_{b1,j}) \quad \forall\, b2, mg, j$$

$$IE_{b2,mg} = \sum_{b1=1}^{i} MG_{b1,b2,mg} \cdot Z\_3_{b1} \qquad \forall\, b2, mg$$

$$Const(x_{I,j}) = \left(\sum_j \sum_i x_{I,j} \cdot B_j - M - s\right)^2$$

Eq. 5: Problem Reformulation to Meet QUBO Requirements Using Classical Formulation

The number of binary variables needed for the expansion of the slack variable will depend on M, following the next relation:

$$N = log_2 M$$

Eq. 6. Calculation of the Number of Binary Variables Needed for the Expansion of the Slack Variable

M being the maximum capacity allowed and N being the number of binary variables needed. If the result is not a natural number, the immediate superior must be chosen.

This way, for a maximum capacity of, for example 30kW, 5 binary variables would be needed, and the binary expansion would be the following:

$$Const\,(x_{i,j}) = \left(\sum_j \sum_i x_{i,j} \cdot B_j - M - y_1 - 2 \cdot y_2 - 4 \cdot y_3 - 8 \cdot y_4 - 16 \cdot y_5\right)^2$$

Eq. 7. Constraint including the Binary Expansion of the Slack Variable for a Maximum Total Installed Capacity of 30kW

The next step would be to present this formulation as a typical QUBO problem, calculating the Q matrix.

VII. QUANTUM APPROACH

There are two ways in which the problem can be addressed using quantum technologies:

A. Quantitative Approach:

This approach consists of reformulating the original problem in order to adapt it to QUBO formulation. This way, the problem is first reformulated by using classical formulation for a continuous variable but getting rid of genetic algorithms and unifying the three objective functions into one. Next, the problem is discretised. In this sense, the optimisation variable now represents whether a certain battery capacity previously specified is installed in a zone or not. Lastly, the restrictions are let go. This way, we are left with a problem that meets the QUBO requirements but is expressed in classical formulation, which is the one presented in the previous section. As mentioned, this must later be adapted in order to have the typical QUBO format, by calculating matrix Q.

Starting with the reliability function, which is the most complex, it is translated into QUBO formulation:

$$Metric\_Reliab = \sum_{b1=1}^{b1=B} \sum_{b2=1}^{b2=B} fr_{b2} * h\,rep_{b2} * IE_{b1} * ZI_{b1,b2}^{ns}$$

$$ZI_{b1,b2} = 1 - \frac{C}{\sum ENS_Z} \rightarrow ZI_{b1,b2} = \frac{C}{\sum ENS_Z}$$

As we are trying to minimise the variable x, constant terms can be taken away to facilitate the calculations. This way, instead of minimising the VOLL we would be maximising the losses that would be prevented with the installation of the battery. Joining these two functions, we obtain the new metric of reliability for one battery size:

$$Metric\_Reliab = \sum_{b1=1}^{b1=B} \sum_{b2=1}^{b2=B} \frac{fr_{b2} * h\,rep_{b2} * IE_{b1} * C}{\sum ENS_Z} \cdot x_{b1} = \sum_{b1=1}^{b1=B} Q_{b1} \cdot x_{b1}$$

Eq. 8. QUBO Formulation of the Reliability Metric

As can be seen, there is an interaction between faulted zones and batteries, but not between batteries. Consequently, there are no quadratic terms affecting the optimisation variable x, meaning the problem is linear and, therefore, the QUBO matrix diagonal. This way, it would be hard to show quantum advantage as it can be solved using classical computation. Therefore, the introduction of new terms into the problem should be studied to increase the complexity and make it more suitable for quantum computing. This could be studying the interaction between batteries and allowing for more than one battery of the same size to be placed in the same zone.

B. Heuristic Approach:

Even if using the previously discussed formulation it could be difficult to show a quantum advantage, other heuristic methods could be used to solve the problem with quantum computing and could bring good results. This means solving the problem without computing power flows, reducing the number of calculations and, essentially, using common sense. Some relevant strategies that could be followed are presented next:

- **Energy Community Detection:** This strategy, inspired in [17], could allow for simplifying the network even more to work with large power systems. As creating these logical partitions of the networks is known to be an NP-complete problem, quantum computing would be extremely useful to solve it. This way, the previously calculated zones would be divided into energy communities using complex network and graph theory to facilitate the clustering process by using modularity optimisation. The optimisation problem proposed will try to maximise a metric for modularity, which will show how connected the elements of one partition are compared to others.

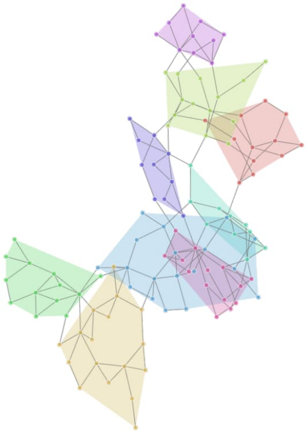


Fig 7. Example of Energy Community Detection using Quantum Computing for Simplifying a Meshed Network [17]

Literature shows great results when using quantum annealers with this purpose. Depending on the size of the resulting network after the energy community detection, the allocation problem could be solved using quantum or classical computing.

- **Eigencentrality:** The end of the graph branches or the pendant nodes are expected to be the most critical because they are connected to the least number of nodes and, therefore, are more likely to suffer an outage in case of fault. In this sense, eigencentrality would allow for locating this kind of node. This is the opposite of what is done in [18], where the t-most central nodes are identified. Nevertheless, as the importance of nodes is ranked, the t-least central nodes can also be determined by looking at the t-least important nodes in the rank. This ranking is done by attending to the scores from eigenvectors. This way, the problem of eigencentrality consists of finding the eigenvector that corresponds to the leading eigenvalue of the adjacency matrix of the network.

It is important to remember the network object of study has distributed generation which may be installed in pendant nodes. This must be taken into consideration as the eigencentrality algorithm may choose them as a candidate for placing the batteries when, in reality, in some cases, they shouldn't be. Therefore, a penalty must be included in this sense.

Moreover, depending on the chosen communities and the resulting network topology there might be interesting nodes which are not considered when calculating the eigencentrality. As an example, the following network from [17] is presented:

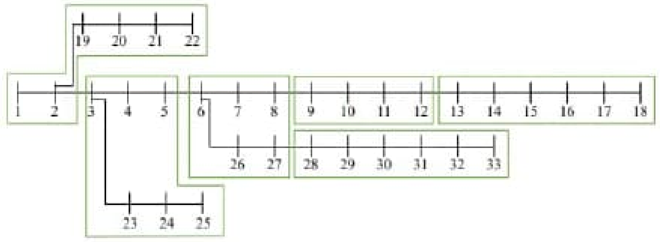
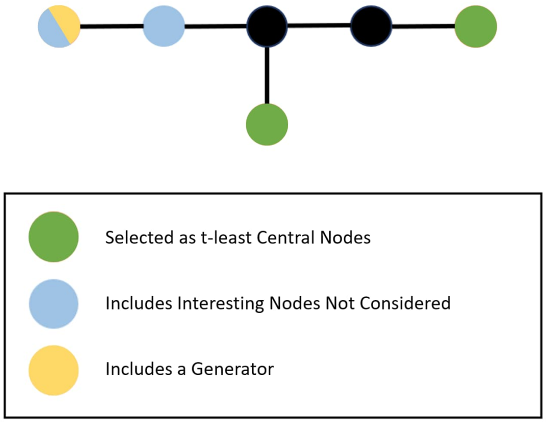


Fig. 8. Example of Energy Community Partitioning for a Radial Network [17]

Attending to the resulting communities which are represented in the diagram, the following graph can be obtained:



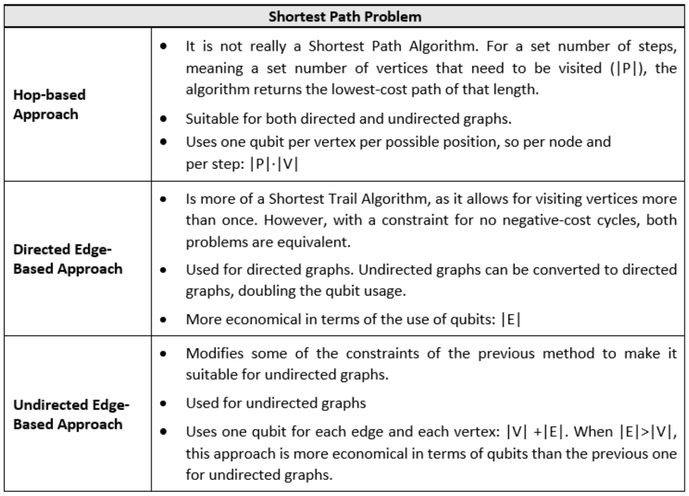
Selected as t-least Central Nodes
Includes Interesting Nodes Not Considered
Includes a Generator

Fig. 9 Example of the Graph Obtained for the Energy Communities in Fig. 8

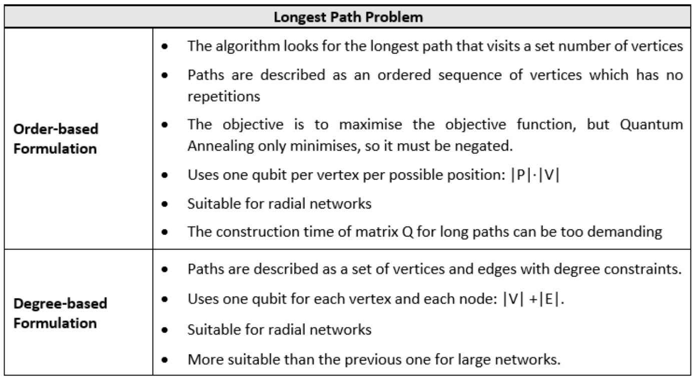
As can be seen, some of the communities (represented in blue) contain pendant nodes which may be interesting, such as node 25. Nevertheless, when translated into the graph this is not reflected and, therefore, it is not considered in the eigencentrality calculations. Also, node 22 may be excluded because of the penalty for the generation. To avoid these situations, some metrics for each community can be pre-computed before using eigencentrality in the entire network.

- **Distance Between Nodes**: Several hypotheses can be made to solve the allocation problem, such as that the batteries should be placed as far as possible from the generators, as far as possible from each other, and as near as possible to important nodes. Thus, the problem is transformed into a matter of maximising and minimising distances between nodes in a graph. To calculate distance between nodes several strategies can be followed. One of them is Dijkstra's classical algorithm. However, other quantum alternatives using QUBO formulation allow for solving the longest [20] and shortest path problems [19], which are presented in the following table:

Table 1. Methods for Solving Shortest and Longest Path Problems Using Quantum Computing



Shortest Path Problem	
Hop-based Approach	• It is not really a Shortest Path Algorithm. For a set number of steps, meaning a set number of vertices that need to be visited (|P|), the algorithm returns the lowest-cost path of that length. • Suitable for both directed and undirected graphs. • Uses one qubit per vertex per possible position, so per node and per step: |P|·|V|
Directed Edge-Based Approach	• Is more of a Shortest Trail Algorithm, as it allows for visiting vertices more than once. However, with a constraint for no negative-cost cycles, both problems are equivalent. • Used for directed graphs. Undirected graphs can be converted to directed graphs, doubling the qubit usage. • More economical in terms of the use of qubits: |E|
Undirected Edge-Based Approach	• Modifies some of the constraints of the previous method to make it suitable for undirected graphs. • Used for undirected graphs • Uses one qubit for each edge and each vertex: |V| +|E|. When |E|>|V|, this approach is more economical in terms of qubits than the previous one for undirected graphs.



Longest Path Problem	
Order-based Formulation	• The algorithm looks for the longest path that visits a set number of vertices • Paths are described as an ordered sequence of vertices which has no repetitions • The objective is to maximise the objective function, but Quantum Annealing only minimises, so it must be negated. • Uses one qubit per vertex per possible position: |P|·|V| • Suitable for radial networks • The construction time of matrix Q for long paths can be too demanding
Degree-based Formulation	• Paths are described as a set of vertices and edges with degree constraints. • Uses one qubit for each vertex and each node: |V| +|E|. • Suitable for radial networks • More suitable than the previous one for large networks.

One of the main limitations of working with large networks is the limited number of qubits of quantum computers. As the network size is reduced, this is not expected to be an issue in this case. Nevertheless, alternatives like Hybrid Computing could be used to reduce the number of qubits needed. In this sense, the quadratic parts of the QUBO problem would be solved using a quantum computer, while the linear parts could be solved using a classical computer.

The problem will be attempted to be solved by using the quantitative approach first, and the heuristic approach will be left as a backup in case it is not possible to find a quantum advantage using the quantitative approach. However, the problem is yet to be implemented into quantum computing, as it is part of a larger project carried out by i-DE in collaboration with Multiverse Computing. In this sense, this is just a first approximation for solving the problem using quantum computing and it may vary as the project develops.

VIII. CONCLUSIONS

For the set problem the most computationally complex parts of the algorithm have been identified by analysing the total computation time and its scalability. It has been determined that the optimisation is the most time-consuming part of the algorithm, with 56% of the total time, followed by the pre-processing with 43% of the total time. Looking at the specific functions, the power flow and the optimisation have been identified as the most time-consuming. As there are not any quantum technologies that allow for solving power flows for large networks, this paper has focused on the optimisation.

It is concluded this problem is suitable to be solved with a quantum computer, in particular a quantum annealer, as it could be possible to find a quantum advantage. As the problem has at least a cubic computation time growth, it could benefit from the quantum speedup, especially when working with larger networks, boosting at the same time the introduction of quantum technologies into the power sector.

For its implementation into quantum computing, two different strategies can be followed as previously explained: a quantitative approach and a heuristic approach. Regarding the quantitative approach, the reliability function, which is the most computationally complex shows a linear behaviour when translating it into QUBO. This may make it difficult to find a quantum advantage using this approach. Nevertheless, new hypotheses could be introduced to increase the complexity and transform it into a quadratic problem, such as allowing for placing more than one battery of the same type in the same node and taking into account their interaction. The heuristic approach shows a completely different way of solving the problem, using different hypotheses to the original problem but reaching the same solution: an optimal battery network design. With this approach, it may be easier to find a quantum advantage, but it would be harder to find a benchmark to analyse the actual improvement obtained by using quantum computing.

IX. FUTURE WORK

As previously mentioned, this paper is part of a larger project that will finish with the implementation of the problem into quantum computing. This will allow for comparing the results obtained with both computing paradigms to determine whether quantum computing has provided any benefits and, therefore, is suitable for solving this kind of problem. The full strategy for the implementation is yet to be determined. Some useful tools to do so have been presented in this paper. Nevertheless, other strategies must be explored to find the best approach for solving the problem.

Regarding the power flow, it is one of the most computationally complex functions in the problem. However, it is not possible to translate it into quantum computing yet. In this case, the power flow is run several times in order to obtain the hourly voltage profile in a year for each node. This way, as every iteration is completely independent, this could be parallelised to reduce the total computation time.

Two main simplifications were made for project Flexener which could possibly be let go using quantum computing. The first is carrying out the network partitioning attending to the placement of tele-controlled switches. This may not be the best approach as one of the objectives of the algorithm is to minimise the investment and installing new tele-controlled switches is much cheaper than installing batteries. Therefore, other ways of partitioning the network should be studied. An interesting approach would be using quantum computing, particularly the previously explained energy communities strategy.

Another one of the simplifications made is regarding the formulation of the objective function related to the investment. Many aspects have not been taken into account, such as the cost of other elements such as inverters or the installation and maintenance cost, which may be lower if batteries are connected in the same place or if mobile batteries are used. This way, the formulation of this objective function should be revised.

REFERENCES

[1] Sigrit. L, "Islanded Operations and Islanding of Microgrids", Operation and Planning of Future Distribution Networks, Universidad Pontificia Comillas ICAI, February 2022.

[2] Postigo Marcos. F, and Mateo Domingo. C, "T.3.1.2. Algoritmo que determina el tamaño y la ubicación de baterías", Instituto de Investigación Tecnológica, Universidad Pontificia Comillas ICAI, February 2022.

[3] "Quantum Computing. Progress and Prospects", National Academies of Sciences, Engineering, and Medicine. (2019). Quantum Computing: Progress and Prospects - National Academies of Sciences, Engineering, and Medicine, Division on Engineering and Physical Sciences, Intelligence Community Studies Board, Computer Science and Telecommunications Board, Committee on Technical Assessment of the Feasibility and Implications of Quantum Computing - Google Libros

[4] Microsoft. "What is quantum computing?". What is Quantum Computing | Microsoft Azure

[5] IBM, "What is Quantum Computing?", What is Quantum Computing? | IBM

[6] AWS, Amazon, "¿Qué es la Computación Cuántica?", ¿En qué consiste la computación cuántica? - Explicación sobre la computación cuántica - AWS (amazon.com)

[7] Quantum Zeitgeist, "Differences between quantum annealers and gate-based quantum computing", Differences Between Quantum Annealers And Gate-based Quantum Computing — Quantum Computing News And Features (quantumzeitgeist.com)

[8] Giani. A and Eldredge. Z, "Quantum Computing Opportunities in Renewable Energy". SN Computer Science, 2, no. 393, 2021. Quantum Computing Opportunities in Renewable Energy | SpringerLink

[9] D-Wave, "What is Quantum Annealing?" What is Quantum Annealing? — D-Wave System Documentation documentation (dwavesys.com)

[10] Microsoft. "Hybrid Quantum Computing", 2023. Overview of hybrid quantum computing - Azure Quantum | Microsoft Learn

[11] Herman. A, "The Quantum Revolution Is Here, Its Name Is Hybrid", Forbes, 2022. The Quantum Revolution Is Here, Its Name Is Hybrid (forbes.com)

[12] L e S Lopes. P, "Quantum Algorithms vs. Quantum-Inspired Algorithms", 2023. Quantum Algorithms vs. Quantum-Inspired Algorithms - The New Stack

[13] Kahane. T, "Quantum-Inspired Computing – What is it and why does it matter?" Fujitsu, 2020. Quantum-Inspired Computing – What is it and why does it matter? : FUJITSU BLOG - Global

[14] Sreedhar. R, "QIOA: Quantum Inspired Optimisation Algorithm", Department of Microtechnology and Nanoscience, Chalmers University of Technology, 2020. QIOA: Quantum Inspired Optimisation Algorithm (chalmers.se)

[15] Farhi. E, Goldstone. J and Gutmann. S, "A Quantum Approximate Optimization Algorithm", arXiv:1411.4028, 2014. 1411.4028.pdf (arxiv.org)

[16] Glover. F, Kochenberger. G, and Du. Y, "Quantum Bridge Analytics I: A tutorial on Formulating and Using QUBO Models", 2019. arXiv:1811.11538

[17] Ferbández-Campoamor. M, O'Meara. C, Cortiana. G, Peric. V and Bernabé-Moreno. J, "Community Detection in Electrical Grids Using Quantum Annealing", 2021 arXiv:2112.08300

[18] Akrobotu. P. D, James. T. E, Negre. C. F. A and Mniszewski. S. M, "A QUBO formulation for top-t eigencentrality nodes", 2022. arXiv:2105.00172

[19] McCollum. J and Krauss. T, "Solving the Network Shortest Path Problem on a Quantum Annealer", IEEE Transactions on Quantum Engineering, vol. 1, pp. 1-12, 2020. Solving the Network Shortest Path Problem on a Quantum Annealer | IEEE Journals & Magazine | IEEE Xplore

[20] McCollum. J and Krauss. T, "QUBO formulations of the Longest Path Problem", Theoretical Computer Science, Vol 863, pp 86-101, 2021. QUBO formulations of the longest path problem - ScienceDirect