Research Project with Quandela - Quarmen EMJM

David Jesús Árbol Guerrero

May 2024

1 Introduction

In this project we treat the problem of optimal clustering of a dataset for further training of a machine learning model. Here, we focus in the first step, the problem of clusterisation, which can be regarded as a QUBO problem. The objective of this work is to replicate the QUBO problem defined in [1] in D-Wave, to solve it using Simulated Quantum Annealing and Quantum Annealing, possibly adding new constraints; and finally, to discuss the dependence of the problem's complexity with the size of the dataset. In particular, we will use the Iris Dataset.

Apart from the model given in [1], a new one was introduced for our training data to be more balanced and to improve its quality [2]. A comparison between them was given to show if the second model actually improves the first.

2 Theoretical Background

2.1 The Iris Dataset

The Iris Dataset is a classically used dataset which contains information about sepals length and width, and petals length and width in cm, of 150 flower samples within 3 different species of flowers: 'setosa', 'versicolor' and 'virginica'.

For simplicity we will enumerate these species as 0, 1 and 2 respectively.

Therefore, as explained in [1] we must separate the dataset X in different pure-class subsets: $X^0, X^1, ..., X^v \subset X$; and in this case $v = 2$ to have 3 subsets.

2.2 Clustering problem

The Clustering Problem is a very relevant matter in the field of Machine Learning because it consists of finding the best training dataset to train a model. If we find the best dataset, we will get the most reliable Machine Learning (ML) model that can be achieved with this data.

The dataset consists of 2 variables: X, a set of N vectors in $\mathbb{R}^d$ containing the d features of the samples that we want to classify; and $y \in \{0, 1, ..., v-1\}^N$, that divide the vectors in the v different *classes* or *subsets* $X^0, ..., X^r, ..., X^{v-1}$. We want to train a ML model to be able to classify any input data in $\mathbb{R}^d$ within the different classes.

The ML model will take the dataset that is already classified and will perform an optimisation of parameters in a perceptron to get the optimal classification. However, many studies [1, 2] claim that it is essential to do a previous treatment of the data to get better results. Thus, we must clusterise each subset's data in different clusters with minimum distance maintaining the *balance* of the clusters within the same *pure-class* subset. Moreover, each point must belong to exactly one cluster.

2.3 Choice of the number of clusters: Silhouette Average

The Silhouette Average is a metric used to evaluate the cluster quality. It measures how well points are clustered within their clusters and how separated are from other clusters. It is defined for each point and, averaging over all the points, we obtain the clustering performance.

We will use this package to obtain the best number of clusters K^r for each subset X^r. It can also be used to evaluate the quality of the clustering, going the score from -1, the worst to 1, the best.

3 Defining the QUBO problem

We can define a QUBO problem for each subset X^r to generate the clustering. Let's consider the subset X^r, $r \in \{0, ..., v-1\}$. To achieve a good cluster, an appropiate one for training a further machine learning model, we want it to have 3 properties:

- **Minimum distance** [1, 2]: Given a cluster $\Phi_j \subset X^r$, each point in it must be as near as possible to all the other

points in the same cluster. For this reason, we define our objective function, function whose minimisation will give us the optimal assigment of a cluster to each point. It can be expressed as:

$$\min \sum_{i=1}^{K} \sum_{x,y \in \Phi_i} ||x-y||^2 \tag{1}$$

with x, y, points in the considered subset; $\Phi_j \subset X^r$, cluster number j; and K, the optimal amount of clusters in this subset.

This can be regarded as a QUBO problem defining:

$$\hat{w} = [\underbrace{w_{0,0}, ..., w_{N-1,0}}_{\text{cluster 1}}, \underbrace{w_{1,2}, ..., w_{N-1,2}}_{\text{cluster 2}}, ..., \underbrace{w_{1,K-1}, ..., w_{N-1,K-1}}_{\text{cluster K-1}}] = [w_{ij}].$$

with $i \in \{0, ..., N-1\}$ running in all possible points and $j \in \{0, ..., K-1\}$ running in all possible clusters.

$\hat{w}$ is defined as:

$$w_{ij} = \begin{cases} 1 & \text{if } x_i \in \Phi_j \\ 0 & \text{if } x_i \notin \Phi_j \end{cases}$$

Then, the objective function is:

$$f(\hat{w}) = \hat{w}^T (I_k \otimes D) \hat{w} \tag{2}$$

with I_K, the identity matrix of order K and D the matrix of squared distances among points in this subset.

- **Penalty 1** [1, 2]: We want to add the constraint that each piece of data must belong to exactly one cluster. Otherwise, the minimisation would be trivial: $\hat{w} = [0, ..., 0]$, which would make no sense.

  To achieve this, we add a term in the objective function, that is minimum when each point is assigned to exactly one cluster and multiplied by a factor β. This factor must be large enough to obtain solutions that fulfil the constraint but not too much, so we do not loose the main objective: minimising distances within each cluster. From [2] we know that the best option is

$$\beta = max(D) \tag{3}$$

  The penalty is defined as:

$$P_1(\hat{w}) = \beta \sum_{i=0}^{N-1} (\hat{w}_i^T \hat{w}_i - 1)^2 \tag{4}$$

  with $\hat{w}_i = [w_{i,0}, ..., w_{i,K-1}]$.

  We must find its expression in the formalism we are considering. Matrix calculus can require many resources if the problem size increases, so instead of the complex way that P_1 was defined in [2], I calculate its shape differently:

$$\beta(\hat{w}_i^T \hat{w}_i - 1)^2 = \beta(\sum_j w_{ij}^2 - 1)^2 = \\ = -\beta \sum_j w_{ij}^2 + \beta \sum_{j \neq j'} w_{ij}\, w_{ij'} + \beta$$

  We can neglect the last term because it is a constant, which has no use in an optimisation problem. Moreover, we get that the diagonal terms are $-\beta$ and the non-diagonal terms are 0 if $i \neq i'$ and $+\beta$ if $i = i'$, $j \neq j'$.

- **Penalty 2** [1, 2]: Secondly, we want to add a constraint for the clusters to be balanced. This means that clusters within a same subset have a similar size. **This is something that was not considered or explained in** [1] but I found in [2] a way to define this.

  The penalty is defined as:

$$P_2(\hat{w}) = \alpha \sum_{j=0}^{K-1} (\hat{w}_j'^T \hat{w}_j' - N/K)^2 \tag{5}$$

  with $\hat{w}_j' = [w_{0,j}, ..., w_{N-1,j}]$.

  Similarly as with P_1, matrix calculus can require many resources, so I calculate its shape in a different way from [2]:

$$\alpha(\hat{w}_j'^T \hat{w}_j' - \frac{N}{K})^2 = \alpha(1 - \frac{2N}{K}) \sum_i w_{ij}^2 + \alpha \sum_{i \neq i'} w_{ij}\, w_{i'j} + C. \tag{6}$$

  The diagonal terms are $\alpha(1 - \frac{2N}{K})$ while the non-diagonal terms are 0 if $j \neq j'$ and $+\alpha$ if $j = j'$, $i \neq i'$.

  From [2], we know that the best option for maintaining the main objective of minimising the distances within a cluster and the second objective of assigning only one cluster to each point, is to set

$$\alpha = \frac{\max D}{2N/k - 1} \tag{7}$$

4 Functional implementation

4.1 QUBO matrices

We observe a good implementation of the QUBO matrices in the heatmaps of figure 1 for a dataset containing the 90% of the data.

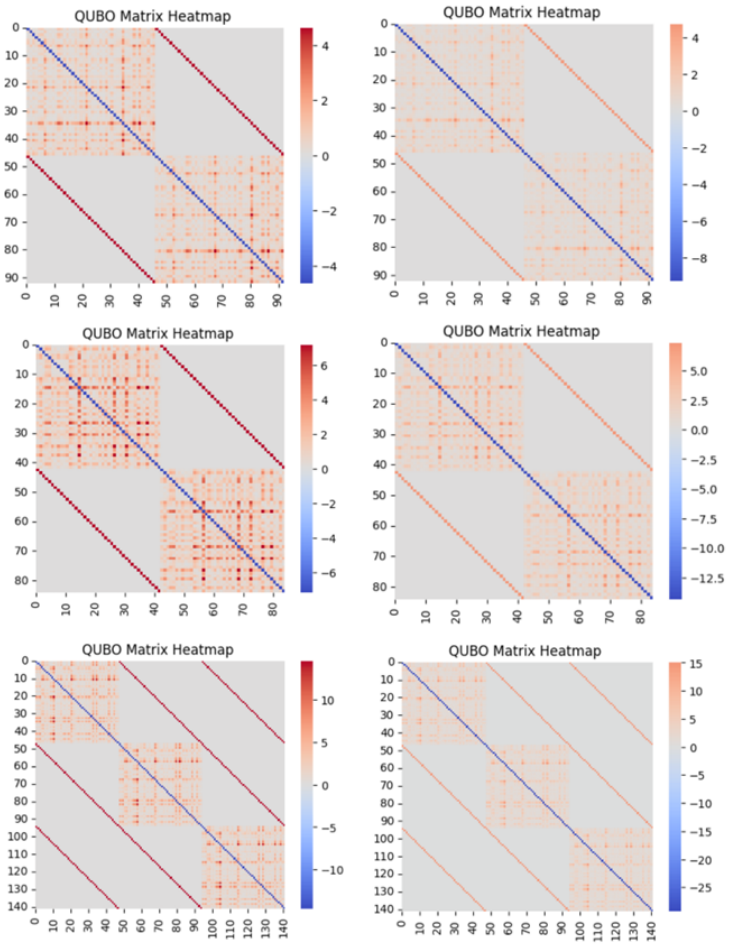

Figure 1: Heatmap of the QUBO matrices for the penalty 1 (left column) and penalties 1 and 2 (right column) for each of the three pure-class subset (rows 0, 1 and 2). We observe that for subset 2 we have $K = 3$, while $K = 2$ for subsets 0 and 1.

$\mathbf{P_1}$	X^0	X^1	X^2
M.A.	0	0	0
Un.	0.125	0.111	0.154
C_0	0.5	0.556	0.385
C_1	0.375	0.333	0.462
Good.	0.875	0.889	0.846

Table 1: Ratio of data in each cluster when only penalty 1 (P_1) is used. M.A. means multiply assigned, Un., unassigned, and Good., goodness. The 20% of the data in Iris Dataset was used.

$\mathbf{P_1}$, $\mathbf{P_2}$	X^0	X^1	X^2
M.A.	0	0	0
Un.	0.125	0.222	0.154
C_0	0.5	0.444	0.462
C_1	0.375	0.333	0.385
Good.	0.875	0.889	0.846

Table 2: Ratio of data in each cluster when both penalties 1 and 2 (P_1, P_2) are used. M.A. means multiply assigned Un., Unassigned and Good., goodness. The 20% of the data in Iris Dataset was used.

4.2 Comparison of models with Penalty 1 vs Penalties 1 and 2

We took 3 different-sized datasets, with 20%, 50% and 80% of the Iris Dataset. Then, we compared the *goodness*, percentage of points that are assigned to exactly 1 cluster, and *balance*, similarity in sizes of clusters within the same subset. We took different sizes to make sure that this does not bias the result.

To get the results, we used the `SimulatedAnnealingSampler`. Using this, we can apply a perfect simulator of the quantum computer to solve the problem. It uses Monte Carlo and Metropolis algorithms. The optimisation, according to Metropolis, performs updates at a sequence of decreasing temperatures which gives a local minimum. Using Monte Carlo, we can get multiple local minimum points from which we can choose the "absolute" minimum.

4.2.1 20% of the dataset:

From tables 1, 2 we see that the only change happens in X^1, where the points in C_0 and C_1 are changed in a way that they are more balanced in the case of 2 penalties, but this affects to the amount of assigned data. We observe that data from C_0, that is the biggest cluster, become unassigned.

We would need more studies to clarify if it is better to have

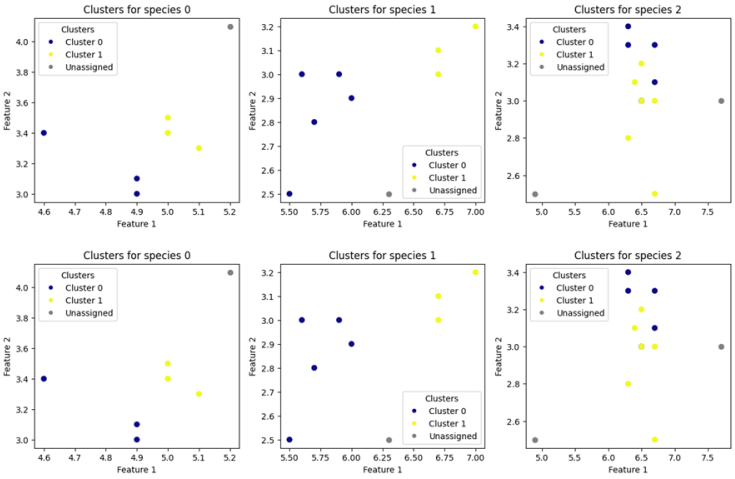

Figure 2: Plot showing the features 1 and 2 of the point of the dataset in each subset. The colours express the clusters classification. The 90% of the data in Iris Dataset was used.

$\mathbf{P_1}$	X^0	X^1	X^2
M.A.	0	0	0
Un.	0.333	0.379	0.227
C_0	0.333	0.310	0.409
C_1	0.310	0.333	0.364
Good.	0.667	0.621	0.772

Table 3: Ratio of data in each cluster when only penalty 1 (P_1) is used. M.A. means multiply assigned, Un., unassigned, and Good., goodness. The 50% of the data in Iris Dataset was used.

good balanced clusters or that the fact that we have less points for the training dataset worsens the model.

In figure 2 we see the classification and in both cases, with and without penalty 2 we have an appropiate behaviour. Notice that the dataset is in $\mathbb{R}^4$ so there are additional dimensions that we cannot appreciate.

4.2.2 50% of the dataset:

We observe in tables 3 and 4 that with the addition of P_2 we lose unassigned points in subsets X^0 and X^1. Moreover, for these subsets the clusters are perfectly balanced and for X^2 it stays the same.

We can conclude that in this case the addition of P_2 is advantageous.

In figure 3 as in the prior case we see the classification and in both cases, with and without penalty 2 we have an appropiate behaviour.

$\mathbf{P_1}$, $\mathbf{P_2}$	X^0	X^1	X^2
M.A.	0	0	0
Un.	0.25	0.310	0.227
C_0	0.375	0.345	0.363
C_1	0.375	0.345	0.409
Good.	0.750	0.690	0.773

Table 4: Ratio of data in each cluster when both penalties 1 and 2 (P_1, P_2) are used. M.A. means multiply assigned Un., Unassigned and Good., goodness. The 50% of the data in Iris Dataset was used.

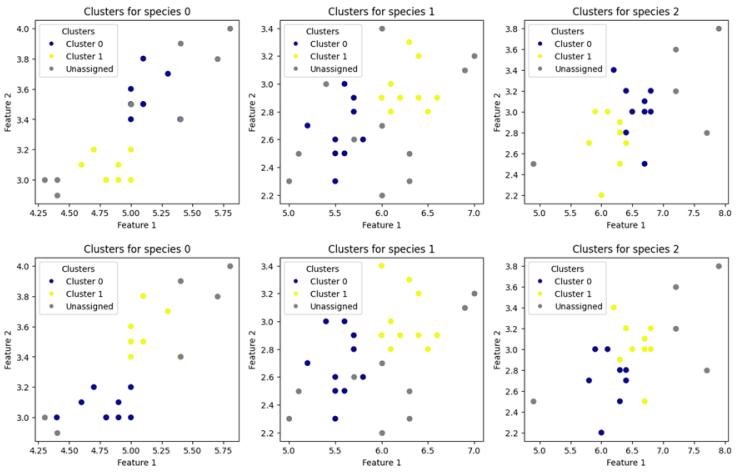

Figure 3: Plot showing the features 1 and 2 of the point of the dataset in each subset. The colours express the clusters classification. The 50% of the data in Iris Dataset was used.

$\mathbf{P_1}$	X^0	X^1	X^2
M.A.	0	0	0
Un.	0.362	0.465	0.378
C_0	0.298	0.256	0.378
C_1	0.340	0.279	0.244
Good.	0.638	0.535	0.622

Table 5: Ratio of data in each cluster when only penalty 1 (P_1) is used. M.A. means multiply assigned, Un., unassigned, and Good., goodness. The 90% of the data in Iris Dataset was used.

$\mathbf{P_1}$, $\mathbf{P_2}$	X^0	X^1	X^2
M.A.	0	0	0
Un.	0.277	0.395	0.311
C_0	0.340	0.326	0.311
C_1	0.383	0.279	0.377
Good.	0.723	0.605	0.689

Table 6: Ratio of data in each cluster when both penalties 1 and 2 (P_1, P_2) are used. M.A. means multiply assigned Un., Unassigned and Good., goodness. The 90% of the data in Iris Dataset was used.

4.2.3 90% of the dataset:

Comparing tables 5, 6 we can see a slight improvement of the goodness when both penalties were used as well as for the balances.

In figure 4 as in the prior case we see the classification and in both cases, with and without penalty 2 we have an appropiate behaviour.

4.2.4 Comparison of using or not using the second penalty

In general it was observed that including the second penalty improves the goodness and the balance of the clusters. However, this is because the unassigned points for the case of 1 penalty are used to fill the smaller clusters to balance them and this can affect to the main problem of distance minimisation of the points within each cluster.

Moreover, due to the fact that we are studying this property for an only dataset and that we took only 3 measurements, we can conclude that further research is needed to discern if the penalty supposes an advantage in general. Actually, there are several of them as we can see in [2], but this report is a modest study. In this case, it seems that it is preferable to use the penalty 2.

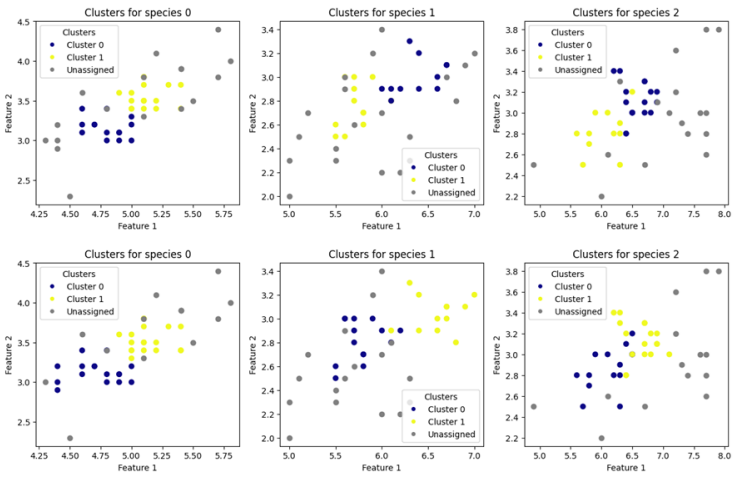

Figure 4: Plot showing the features 1 and 2 of the point of the dataset in each subset. The colours express the clusters classification. The 90% of the data in Iris Dataset was used.

4.3 Comparison of different solving methods: classical, simulated quantum annealing and quantum annealing

In this section we took the 15%, 50% and 90% of the data and solved the clustering problem using both penalties with different methods. We used **2 classical solvers**. The first consisted of sampling all the possible values of $\hat{w}$ and pick the ones that minimise the objective function. We saw that it could work only for around $\lesssim 17\%$ of the data, which makes 25 points in total. This is beacause the time complexity of the problem increases as $\mathcal{O}(Nk^2)$. The second, of random sampling $\hat{w}$.

Later, we used the `SimulatedAnnealingSampler`, a **classically simulated quantum computing process** that we already discussed in the section 4.2. We will see that, the energies obtained with the Metropolis Algorithm are much lower than the ones using a random sampler. But if we want to get better results, for the case in which we have a large dataset, we need to use the D-Wave's Leap Hybrid Solver. We can use Quantum Advantage to solve this problem, because it should scales slower by means of this method. The complexity is treated later in the section 6.

Finally, we use the **Quantum Annealing Sampler of D-Wave** to solve the QUBO problem (and as it is real, it will be noisy). Concretely, it is called `Advantage_system4.1`. As we are using the free trial we only have 1 minute in total to compute and we cannot compute longer than 0.6 s each time we access to D-Wave servers. For this reason we do not observe any advantage with this method. In the case of small-sized datasets we get the same results as with `SimulatedAnnealingSampler` while for large-sized ones, this method is worse. In this context, worse means that the computed objective function minimum

Solver	X^0	X^1	X^2
All	-29.64	-14.92	-37.59
Random	-17.66	9.14	-4.01
S.A.	-29.64	-14.92	-37.59
Quantum	-29.64	-14.92	-37.59

Table 7: Objective function minimum comparison for each solver. The 15% of the data was used.

	X^0	X^1	X^2
M.A.	0	0	0
Un.	0.167	0.143	0.222
C_0	0.333	0.286	0.333
C_1	0.5	0.286	0.444
Good.	0.833	0.857	0.778

Table 8: Simulated Annealing and Quantum Annealing: Both presented the same results. Ratio of data in each cluster with both penalties. M.A. means multiply assigned Un., Unassigned and Good., goodness. The 15% of the data in Iris Dataset was used.

value is higher.

In conclusion, we obtained better results with the classically simulated quantum sampler (`SimulatedAnnealingSampler`) because we do not have enough power and time to compute in D-Wave servers. Still, according to [2], even using all their resources (5,627 qubits and 40,279 couplers) there is not advantage because D-Wave has not reached quantum advantage yet in this field.

Then, the data from which these conclusions were took are presented here.

4.3.1 Solvers: 15% of the dataset

Given the low amount of data, for "All", "S.A." and "Quantum" we obtained the same objective function minimum value. Moreover, the analysis for the Quantum and Simulated Annealing is the same. There is very good balance and goodness for the solutions obtained. We can see the clusters obtained using Simulated Annealing in the figure 5.

4.3.2 Solvers: 50% of the dataset

In the tables 9 y 12 we have that the Quantum Annealing solver obtained a slightly better minimum. We got a small advantage only in this case and we can conclude that this is due to a statistic deviation and the small size of the dataset. This claim comes from the fact that using the 90% of the dataset the advantage of the Simulated Annealing solver is very evident in all

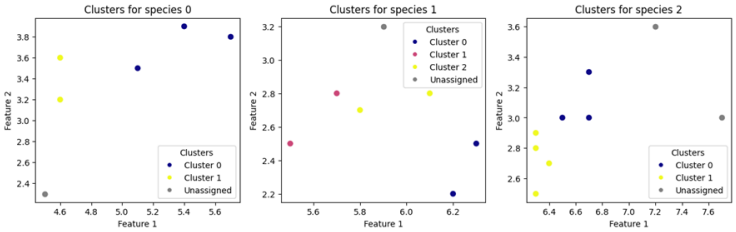

Figure 5: Simulated Annealing: Plot showing the features 1 and 2 of the point of the dataset in each subset. The colours express the clusters classification. The 15% of the data in Iris Dataset was used.

	X^0	X^1	X^2
M.A.	0	0	0
Un.	0.154	0.25	0.16
C_0	0.423	0.375	0.440
C_1	0.423	0.375	0.400
Good.	0.846	0.792	0.840
F.O. min.	-130.16	-173.42	-220.46

Table 9: Simulated Annealing: Ratio of data in each cluster with both penalties. M.A. means multiply assigned Un., Unassigned and Good., goodness. The 50% of the data in Iris Dataset was used.

of the 3 of the subsets.

4.3.3 Solvers: 90% of the dataset

As it was said in the last subsection, we have that when the size of the dataset is big, the Simulated Annealing (classically simualted) results in better minimum values. This is because the quantum computer does not have enough resources yet to reach quantum advantage. Actually, we are using very few resources from the computer because we have access only to the free trial.

	X^0	X^1	X^2
M.A.	0	0	0
Un.	0.154	0.25	0.16
C_0	0.423	0.375	0.400
C_1	0.423	0.375	0.440
Good.	0.846	0.750	0.840
F.O. min.	-130.16	-173.50	-220.46

Table 10: Quantum Annealing: Ratio of data in each cluster with both penalties. M.A. means multiply assigned Un., Unassigned and Good., goodness. The 50% of the data in Iris Dataset was used.

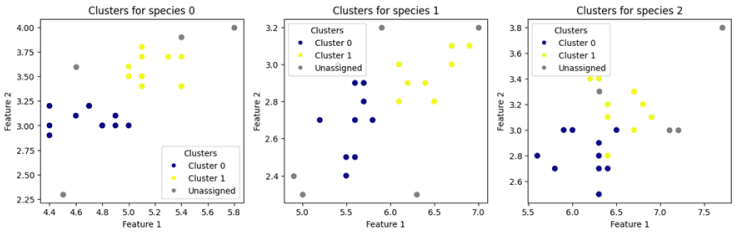

Figure 6: Simulated Annealing: Plot showing the features 1 and 2 of the point of the dataset in each subset. The colours express the clusters classification. The 50% of the data in Iris Dataset was used.

	X^0	X^1	X^2
M.A.	0	0	0
Un.	0.349	0.381	0.170
C_0	0.326	0.286	0.340
C_1	0.326	0.333	0.277
C_2	–	–	0.213
Good.	0.652	0.619	0.830
F.O. min.	−184.95	−237.83	−729.80

Table 11: Simulated Annealing: Ratio of data in each cluster with both penalties. M.A. means multiply assigned Un., Unassigned and Good., goodness. The 90% of the data in Iris Dataset was used.

	X^0	X^1	X^2
M.A.	0	0	0
Un.	0.348	0.381	0.191
C_0	0.348	0.310	0.298
C_1	0.304	0.320	0.298
C_2	–	–	0.212
Good.	0.652	0.619	0.809
F.O. min.	−178.32	−230.16	−679.26

Table 12: Quantum Annealing: Ratio of data in each cluster with both penalties. M.A. means multiply assigned Un., Unassigned and Good., goodness. The 90% of the data in Iris Dataset was used.

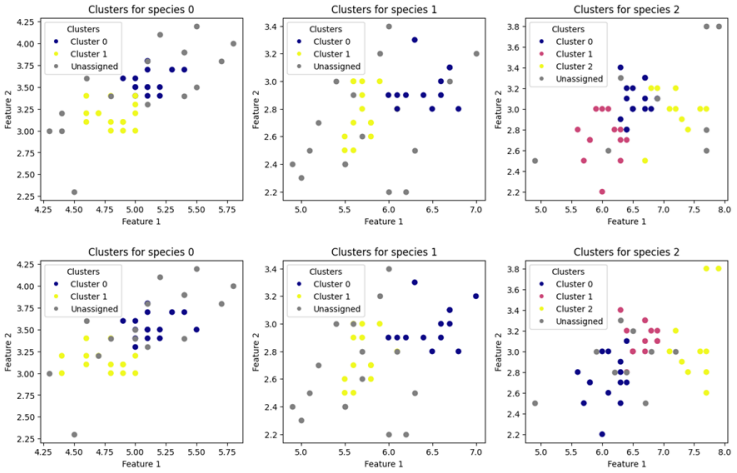

Figure 7: Simulated and Quantum Annealing (first and second row respectively): Plot showing the features 1 and 2 of the point of the dataset in each subset. The colours express the clusters classification. The 90% of the data in Iris Dataset was used.

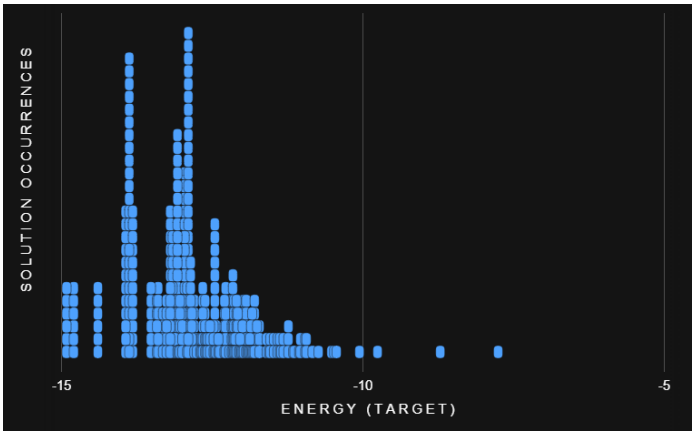

Figure 8: Quantum Annealing data obtained with the free trial access to D-Wave servers. The 15% of the data in Iris Dataset was used.

4.3.4 Examples of quantum data from subset 1

In figures 8 (15%), 9 (50%), 10 (90%), we observe the quantum data that we obtained using D-Wave servers. These are the computed local minimums for subset 1. As we see, for 8 (15%) and 9 (50%), there are enough resources to accomplish many iterations of the algorithm so as to obtain statistically meaningful results. On the other hand, for 10 (90%) the results are not statistically meaningful because the frequencies are very small and we cannot qualitatively distinguish any statistical behaviour. This means that we cannot trust the minimums obtained. Therefore, we should increase the number of iterations or qubits to get reliable results in this last case but the trial access did not allow to do it.

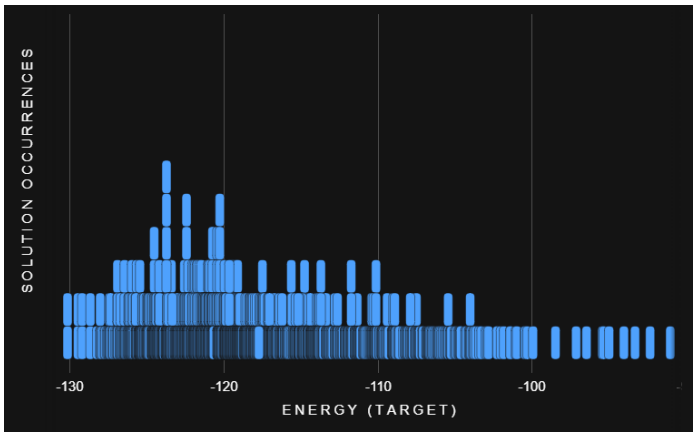

Figure 9: Quantum Annealing data obtained with the free trial access to D-Wave servers. The 50% of the data in Iris Dataset was used.

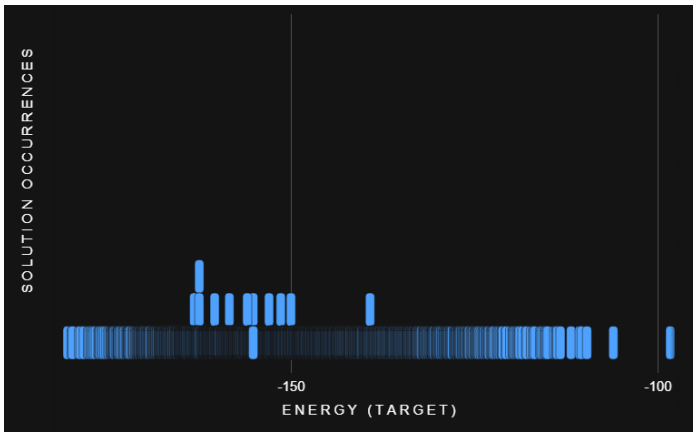

Figure 10: Quantum Annealing data obtained with the free trial access to D-Wave servers. The 90% of the data in Iris Dataset was used.

5 Originality

This project has followed the article [1] as it was explained in the instructions given. However, it did not explained how to obtain balanced clusters and after some research, a way was found in [2]. However, it was concluded while programming that the mathematical formalisation described in [2] was computationally inefficient with the size of the matrices (number of points in the dataset) so this report and the notebook shows a better way to implement it.

As [1] only considered the first penalty (to assign an only cluster to each point) and [2] considered two (another one to balance the cluster), it seemed interesting to compare both methods. The comparison is in section 4.2.

Also, the instructions did not consider the use of a real evaluation of quantum annealing but this project compared a real (and therefore noisy) one with the perfect simulated annealing solver.

6 Complexity and resources [2]

QUBO problems are NP-hard type. Its time complexity for classical non-quantum algorithms is $\mathcal{O}(Nk^2)$. For Lloyd's (non-quantum) we need $\mathcal{O}(Nkd)$ to get a locally optimal solution but $\mathcal{O}(N^{kd+1})$ to exactly solve it. The number of variables are $\mathcal{O}(Nk)$ while the quadratic qubit footprint is $\mathcal{O}(N^2k^2)$. In the worst case, classically we have $\mathcal{O}(N^2kd)$. However, it can be solved through quantum annealing in constant time $\mathcal{O}(1)$ which is a huge advantage. The problem is the embedding, that scales quadratically with the number of variables. **Note:** N is the number of variables, k the number of clusters and d the number of features.

At the moment, the quantum processor we used has 5,627 qubits 40,279 couplers, which is not enough. We need to improve the embedding to get quantum advantage in this problem so we would not need many more qubits to achieve this, but the architecture of the processor must be deeply revised.

7 Conclusions

Firstly, it was showed a comparison between two methods of clustering: one considered a penalty to assign exactly one cluster to each point of the dataset, the other one was to get balanced clusters, with similar size. We could conclude that we got a clustering with less unassigned points (goodness) and they are very well balance using both penalties instead of using only the first. However, we must take this claim carefully because we have an small dataset and few samples were taken (25%, 50% and 90%).

Secondly, we compared different methods to solve the QUBO problem for 15%, 50% and 90% of the data. The two classical ones consisted of sampling every possible solution and check the value of the objective function while the other, of sampling random possible solutions. They were obviously inefficient and the first could not be computed when using the 50% and 90% of the data. Then, we used the (classically) Simulated Annealing and the Quantum Annealing with D-Wave. We concluded that the resources we could use with the trial access to D-Wave servers were insufficient. Then, the classically simulated annealing got better results of the minimum for big datasets (90%) while it got the same results for small ones (15%, 50%). Moreover, according to [2], the D-Wave quantum computer cannot get quantum advantage against the best known Lloyd's Algorithm for clustering, which is classical. However, we did not use this one as this is a modest report with educational purposes.

References

[1] Lenny Putri Yulianti, Agung Trisetyarso, Judhi Santoso, Kridanto Surendro.
A hybrid quantum annealing method for generating ensemble classifiers.
Journal of King Saud University - Computer and Information Sciences 35 (2023), 101831.

[2] Arthur, D., Date, P., 2021.
Balanced k-means clustering on an adiabatic quantum computer.
Quantum Inf Process 20, 294. https://doi.org/10.1007/s11128-021-03240-8.