Overview of QuantumCLEF 2024: The Quantum Computing Challenge for Information Retrieval and Recommender Systems at CLEF

Andrea Pasin[1], Maurizio Ferrari Dacrema[2], Paolo Cremonesi[2], and Nicola Ferro[1]

[1] University of Padua, Italy
andrea.pasin.1@phd.unipd.it, nicola.ferro@unipd.it
[2] Politecnico di Milano, Italy
{maurizio.ferrari, paolo.cremonesi}@polimi.it

Abstract. *Quantum Computing (QC)* is an innovative research field that has gathered the interest of many researchers in the last few years. In fact, it is believed that QC could potentially revolutionize the way we solve very complex problems by dramatically decreasing the time required to solve them. Even though QC is still in its early stages of development, it is already possible to tackle some problems by means of quantum computers and to start catching a glimpse of its potential. Therefore, the aim of the QuantumCLEF lab is to raise awareness about QC and to develop and evaluate new QC algorithms to solve challenges that can be encountered when implementing *Information Retrieval (IR)* and *Recommender Systems (RS)* systems. Furthermore, this lab represents a good opportunity to engage with QC technologies which are typically not easily accessible.
In this work, we present an overview of the first edition of QuantumCLEF, a lab that focuses on the application of *Quantum Annealing (QA)*, a specific QC paradigm, to solve two tasks: Feature Selection for IR and RS systems, and Clustering for IR systems. There have been a total of 26 teams who registered for this lab and eventually 7 teams managed to successfully submit their runs following the lab guidelines. Due to the novelty of the topics, participants have been provided with many examples and comprehensive materials that allowed them to understand how QA works and how to program quantum annealers.

1 Introduction

Information Retrieval (IR) and *Recommender Systems (RS)* systems have been studied and improved for several years. Nowadays, these systems need to face very complex challenges such as applying computationally expensive methods to huge amounts of data that are constantly being produced.

To solve this issue, researchers are now investigating *Quantum Computing (QC)*, an emerging computing paradigm that has the potential to revolutionize the way we currently solve problems. QC is not only about a new technology that can be used in place of traditional hardware, but it also represents a

paradigm shift that allows to view and solve problems from a new perspective exploiting quantum physics principles. Thanks to principles such as superposition and entanglement, quantum computers can theoretically explore exponentially larger problem spaces with respect to traditional computers considering devices with the same number of quantum bits (qubits) and traditional bits respectively.

In recent years, quantum computers have started to become more robust, powerful, and accessible. This has allowed researchers and practitioners to start exploring the application of QC to practical problems. However, QC is still in its infancy and there are several limitations yet to overcome, most of which concerning the hardware. In fact, qubits are very delicate and must be completely isolated from the environment since any interferences or noises (e.g., electromagnetic interferences, thermal fluctuations) could impact their state, thus breaking the computation. On the other hand, traditional systems have been developed for decades and they represent more robust alternatives.

In this exciting and innovative context, it is natural to wonder whether it is possible to apply QC to solve some of the complex tasks that are faced by IR and RS systems. For this reason, we decided to start a new CLEF lab called QuantumCLEF [20, 21] which focuses on the study, development, and evaluation of QC algorithms for IR and RS. This lab has 4 main objectives:

- develop new QC algorithms for IR and RS and evaluate them, comparing the results (efficiency and effectiveness) with traditional approaches;
- gather all resources and data for future researchers to compare their results with the ones achieved during the lab;
- allow participants to learn more about QC through comprehensive materials and to use real quantum computers, which are still not easily accessible to the public;
- raise the awareness of the potential of QC and form a new research community around this new field.

In this paper, we present the overview of the first edition of QuantumCLEF held in 2024. This edition has focused on the usage of *Quantum Annealing (QA)*, a specific QC paradigm that can be used to tackle optimization problems. We have granted participants access to the state-of-the-art QA devices (quantum annealers) produced by D-Wave, one of the leading companies in this sector. The QA paradigm is easier to understand with respect to the Universal Gate-Based paradigm. Furthermore, D-Wave provides several tools and libraries to program quantum annealers without requiring a very deep knowledge of the quantum physics governing these devices.

This QuantumCLEF edition was composed of two main tasks:

- **Task 1**: Feature Selection for IR and RS;
- **Task 2**: Clustering for IR.

Participants were asked to develop their own algorithms to solve the tasks using both QA and *Simulated Annealing (SA)*, a well-known optimization approach similar to QA but without any quantum effects and therefore can be run

on classical devices. Due to the novelty of the topics, comprehensive materials (i.e., videos, slides, and examples) were provided to the participants to lower the entry barrier and to allow them to understand how QA works and how to program quantum annealers. An ad-hoc infrastructure has been created to grant participants access to real quantum annealers while also easing the workflow and enhancing reproducibility. In total 26 teams participated in our tasks, 7 of which actively participated and submitted their runs. More specifically, 6 teams managed to successfully submit their runs for Task 1 while 1 team managed to submit for Task 2. The results show that approaches that use QA or *Hybrid (H)* methods are as effective as SA and traditional approaches while being generally more efficient.

The paper is organized as follows: Section 2 discusses related works; Section 3 presents the tasks of the QuantumCLEF 2024 lab while Section 4.1 introduces the lab's setup and the design and implementation of our ad-hoc infrastructure; Section 5 shows and discusses the results achieved by the participants; finally, Section 6 draws some conclusions and outlooks some future work.

2 Related Works

2.1 Background on Quantum and Simulated Annealing

We provide here a brief introduction to QA and to *Simulated Annealing (SA)*, a traditional optimization algorithm that does not take advantage of quantum technologies.

Quantum Annealing. QA is a QC paradigm that is based on special-purpose devices (quantum annealers) able to tackle optimization problems with a certain structure. The basic idea of a quantum annealer is to represent a problem as the energy of a physical system and then leverage quantum-mechanical phenomena, e.g., superposition and entanglement, to let the system find a state of minimal energy, which corresponds to the solution of the original problem.

To use quantum annealers, one needs to formulate the optimization problem as a minimization one using the *Quadratic Unconstrained Binary Optimization (QUBO)* formulation [14], a well-known optimization technique. QUBO is defined as:

$$\min \quad y = x^T Q x \tag{1}$$

where x is a vector of binary decision variables, and Q is a matrix of constant values representing the problem we wish to solve. Then, a further step called *minor embedding* is required to map the general mathematical formulation into the physical quantum annealer hardware, accounting for the limited number of qubits and the physical connections between them. Each quantum annealer or *Quantum Processing Unit (QPU)* has, in fact, its own architecture, which can be seen as a graph: each vertex represents a qubit, and each edge represents an interaction between two qubits. Therefore, minor embedding involves choosing which physical qubits represent the decision variables. If the QUBO problem

does not fit directly in the QPU, for example because a decision variable is connected to more variables than the available physical connections between qubits, multiple connected qubits will be used to represent one decision variable and the connections to the other variables will be split between them. Due to this the number of qubits required to solve a problem on a quantum annealer may be much higher than the number of its decision variables. Minor embedding is a complex task in itself and a NP-hard problem, which can be solved relying on some heuristic methods [8]. If the problem does not fit on the QPU, D-Wave provides *Hybrid (H)* approaches that are able to automatically handle large problems using intelligent techniques to split them and solve them using both traditional methods and QA methods. By splitting problems into sub-problems it will be possible to make them fit inside the QPU of quantum annealers.

Occasionally, it might be necessary to add constraints to the problems. This can be done by means of penalties $\mathrm{P}(x)$ [28], which penalize solutions that do not meet the specified constraints. These penalties are then added to the original cost function y to achieve the final formulation as follows:

$$\min \quad \mathrm{C}(x) = y + \mathrm{P}(x) \ . \tag{2}$$

Penalties can be controlled through hyperparameters to manage their influence with respect to the given formulation.

To sum up, using a quantum annealer requires several stages [28]:

1. **Formulation**: find a way to express the desired algorithm as an optimization problem by leveraging the QUBO framework and compute the actual QUBO matrix Q;
2. **Embedding**: generate the minor embedding of the QUBO for the quantum annealer hardware;
3. **Data Transfer**: transfer the problem and the embedding on the global network to the data center that hosts the quantum annealer;
4. **Annealing**: run the quantum annealer itself. This phase is composed by several stages such as programming the QPU, sampling a solution, and then reading the solution. This is an inherently stochastic process. Therefore, it is usually run a large number of times (hundreds) in which several samples are returned, each one resembling a possible solution to the considered problem. The solutions must then be checked for their feasibility, and then the best one among them (i.e., the optimal one according to the objective function) is usually considered the final solution to the submitted problem.

Generally, once a QUBO problem has been embedded and sent to the quantum annealer, it can be solved in a few milliseconds.

Simulated Annealing. SA is a consolidated meta-heuristic that can be run on traditional hardware [6, 26]. It is a probabilistic algorithm that can be used to find the global minimum of a given cost function, even in the presence of many local minima. It is based on an iterative process that starts from an initial

solution and tries to improve it by randomly perturbing it. The cost function is represented by the QUBO problem formulation, similar to what would be used for QA. In SA, there is no minor embedding phase since the problem is directly solved on a traditional machine.

We underline that SA is an optimization algorithm different from QA, it is not a simulation of QA on a traditional machine, and, therefore these two algorithms are not equivalent. However, SA can be used for benchmarking purposes to show how well QA performs with respect to a traditional hardware counterpart.

The access to quantum annealers in QuantumCLEF is limited to ensure a fair distribution of resources. Therefore, SA can also be used to perform initial experiments to assess a QUBO formulation feasibility without affecting the available quota in the quantum environment.

2.2 Related Challenges

In the context of CLEF, there have not been other challenges involving the application and evaluation of QC. However, since QC technologies are starting to become more available and robust, it is necessary to raise awareness about their potential and to learn how these technologies can be used to possibly improve the current state-of-the-art IR and RS systems.

Outside CLEF, we are not aware of other challenges or shared tasks that have been done in the past involving the use of QC. There are some other challenges starting off this year offered by big-tech companies such as IBM[3] and Google[4]. These challenges involve the development of QC algorithms which will be executed on quantum computers to solve some practical real-world challenges. There has also been a Quantum Computing challenge in 2016 organized by Microsoft[5], which however used simulators for Language-Integrated Quantum Operations and not real quantum computers.

3 Tasks

QuantumCLEF 2024, which was initially presented in a paper at CLEF 2023 [20], addresses two different tasks involving computationally intensive problems that are closely related to the Information Access field: Feature Selection and Clustering. The main goals for each task are:

- finding one or more possible QUBO formulations of the problem;
- evaluating the QA approach compared to a corresponding traditional approach to assess both its efficiency and its effectiveness.

[3] https://challenges.quantum.ibm.com/2024
[4] https://www.xprize.org/prizes/qc-apps
[5] https://www.microsoft.com/en-us/research/academic-program/microsoft-quantum-challenge/challenge/

For each task, we have provided Jupyter Notebooks that served as starting points for the participants to learn how to program quantum annealers and to successfully carry out the tasks following the submission guidelines. Moreover, we provided the slides that were presented during the ECIR Tutorial [10] covering the fundamental concepts of QC and QA. We also streamed and recorded a video tutorial[6] about the usage of our infrastructure and the notebooks available to the participants.

For both tasks, participants are asked to submit their runs using both QA and SA. In this way, it will be possible to compare the efficiency and effectiveness of these two similar optimization techniques that employ quantum annealers and traditional hardware respectively.

3.1 Task 1 - Quantum Feature Selection

This task focuses on formulating the well-known *NP-Hard* feature selection problem in such a way that it can be solved with a quantum annealer, similarly to what has already been done in previous works [9, 18].

Objectives. Feature Selection is a widespread problem for both IR and RS which requires the identification of a subset of the available features (e.g., the most informative, less noisy, etc.) to train a learning model. This problem is very impacting since many of IR and RS systems involve the optimization of learning models, and reducing the dimensionality of the input data can improve their performance. Therefore, in this task, we aim to understand if QA can be applied to solve this problem more efficiently and effectively, exploiting its capability of exploring a larger problem space in a short amount of time.

Sub-tasks. Task 1 is divided into two sub-tasks:

- **Task 1A**: Feature Selection for IR. This task involves selecting the optimal subset of features using QA and SA that will be used to train a LambdaMART [7] model according to a Learning-To-Rank framework;
- **Task 1B**: Feature Selection for RS. This task involves selecting the optimal subset of features using QA and SA that will be used to train a kNN recommendation system model. The item-item similarity is computed with cosine on the feature vectors, a shrinkage of 5 is added to the denominator and the number of selected neighbors for each item is 100.

Datasets. For Task 1A, we decided to employ the famous MQ2007 [23] and the Istella S-LETOR [16] datasets. MQ2007 represents an easier challenge since it has 46 features, allowing direct embedding of the problem formulations inside the QPU of quantum annealers. Istella instead has 220 features and it is impossible

[6] https://www.youtube.com/watch?v=fKrnaJn40Kk/

to embed problem formulations directly, thus requiring some further processing steps for the participants to fit the problem into the physical QPU hardware.

For Task 1B instead, we decided to employ a custom dataset of music recommendations containing 1.9 thousand users and 18 thousand items. The dataset contains both collaborative data, with 92 thousand implicit user-item interactions, as well as two different sets of item features that are derived from item descriptions and user-provided tags, called Item Content Matrix (ICM). The small set, ICM_150, includes 150 features and can be embedded directly on the QPU with small adjustments, the large set, ICM_500, has 500 features and requires significant pruning to fit in the QPU or the use of Hybrid methods. Both sets of features contain noisy and redundant features.

Evaluation Measures. The official evaluation measure for both Task 1A and Task 1B is nDCG@10.

Baseline. For sub-task 1A the baseline is a Feature Selection model that uses a Recursive Feature Elimination approach paired with a Linear Regression model to select the most relevant subset of features.

For sub-task 1B the baseline is a kNN recommendation system model that uses all the available features. The hyperparameters are the same used for the model computed on the selected features, i.e., the item-item similarity is computed with cosine adding a shrink term of 5 to the denominator, and the number of neighbors is 100.

Runs Format. Participants in both tasks 1A and 1B can submit a maximum of 5 runs per dataset using QA or Hybrid methods and a maximum of 5 runs using SA. Each run that uses QA or Hybrid methods should correspond to a run that employs SA. In this way, it is possible to make a fair comparison between them.

The results of the run must be a text file which lists the features that were selected, one per line. The discarded features are not reported in the run file. Furthermore, the last line must report the list of IDs associated with the problems solved using QA, SA, or Hybrid to obtain the final subset of features by the considered approach.

Each run file must be left in each team's workspace in a specific directory called */config/workspace/submissions*, which is already available.

The submission file name should comply with the format *[Task]_[Dataset]_[Method]_[Groupname]_[SubmissionID].txt*, where:

- [**Task**]: it should be either *1A* or *1B* based on the task the submission refers to;
- [**Dataset**]: it should be either *MQ2007*, *Istella*, *150_ICM* or *500_ICM* based on the dataset used;
- [**Method**]: it should be either *QA* or *SA* based on the method used;
- [**Groupname**]: the team name;

– [**SubmissionID**]: a custom submission ID that must be the same for the submissions using the same algorithm but performed with different methods (e.g., QA or SA).

3.2 Task 2 - Quantum Clustering

This task focuses on the formulation of the Clustering problem in such a way that it can be solved with a quantum annealer. It involves grouping the items according to their characteristics. Thus, "similar" items fall into the same group while different items belong to distinct groups.

Objectives. Clustering is a relevant problem for IR and RS since it can be helpful for organizing large collections, helping users explore a collection, and providing similar search results to a given query. Furthermore, it can be beneficial to split users according to their interests or build user models with the cluster centroids [27] speeding up the runtime of the system or its effectiveness for users with limited data.

This task is more focused on the IR field and is applied in a document retrieval scenario where documents have been transformed into their corresponding embeddings by a Transformer model. Each document can be seen as a vector in the space and it is possible to cluster points based on their distances, which can be interpreted as a dissimilarity function: the more distant two vectors are, the more different the corresponding documents are likely to be. In this task, participants should apply QA and SA to cluster documents into 10, 25, and 50 clusters. Participants must report the found centroids and the corresponding associated documents.

By clustering documents, it is possible to reduce the searching time by considering the most similar centroid to the input query and then retrieving only the documents belonging to that centroid's cluster instead of looking at the whole collection of documents.

Clustering fits very well with a QUBO formulation and various methods have already been proposed [3, 4, 25]. Most of these methods involve the usage of one variable per document, thus making it very hard to consider large datasets due to the limited number of physical qubits and interconnections between them. There are ways to overcome this issue, such as by applying a coarsening or a hierarchical approach.

Datasets. For this task, we considered a custom split of the ANTIQUE [15] dataset containing 6486 documents, 200 queries, and manual relevance judgments. Each document and each query have been transformed into a corresponding embedding with the pre-trained **all-mpnet-base-v2** model[7]. The queries are divided into 50 for the Training Dataset and 150 for the Test Dataset.

[7] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

Evaluation Measures. The official evaluation measures for Task 2 are:

- the Davies-Bouldin Index to measure the overall cluster quality without considering the document retrieval phase;
- nDCG@10 to measure the retrieval effectiveness based on the clusters found.

Baseline. For this task, the baseline is a traditional k-Medoids approach using the cosine distance as a distance function.

Runs Format. Participants in task 2 can submit a maximum of 5 runs for each number of clusters (i.e., 10, 25, 50) using QA or Hybrid methods and a maximum of 5 runs using SA. Each run that uses QA or Hybrid methods should correspond to a run that employs SA. In this way, it is possible to make a fair comparison between them.

The run file must be a text file (JSON formatted) with a list of 10, 25, and 50 vectors that represent the final centroids achieved through their clustering algorithm. Each centroid should also be followed by the list of documents that belong to the given cluster. Furthermore, the last line must report the list of IDs associated with the problems solved using QA, SA, or Hybrid to obtain the final clusters by the considered approach.

Each run file must be left in each team's workspace in a specific directory called */config/workspace/submissions*, which is already available.

The submission file name should comply with the format *[Centroids]_[Method]_[Groupname]_[SubmissionID].txt*, where:

- [**Centroids**]: it should be either 10, 25, or 50 based on the number of centroids;
- [**Method**]: it should be either *QA* or *SA* based on the method used;
- [**Groupname**]: the team name;
- [**SubmissionID**]: a custom submission ID that must be the same for the submissions using the same algorithm but performed with different methods (e.g., QA or SA).

4 Lab Setup

In this section, we detail the infrastructure that was specifically created to carry out this lab and we present the guidelines the participants had to comply with to submit their runs.

4.1 Infrastructure

Having access to quantum annealers is not straightforward. In fact, D-Wave enforces some policies on the usage of these devices by setting some monthly timing quotas to submit and solve problems on their devices. There are API

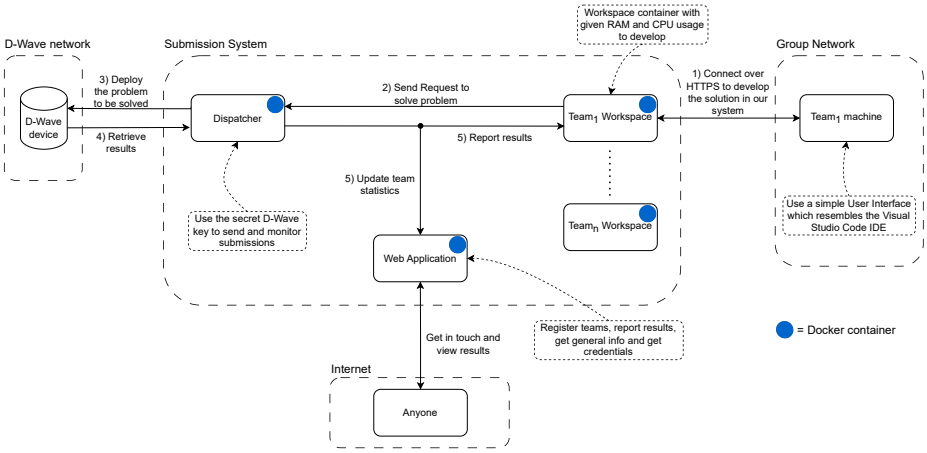


Fig. 1: High-level representation of the infrastructure.

keys that are given to people who use quantum annealers so that it will be possible to monitor the access and usage.

Since it is not possible to disclose our API key to the participants, we decided to build our own infrastructure that allows participants to use quantum annealers without knowing our API key and without needing to stipulate any agreements with D-Wave to obtain their own API keys.

Furthermore, to measure efficiency participants must use the same computing hardware. To this end, our infrastructure provides all the participants with corresponding workspaces located in an AWS server. All workspaces have the same computational resources in terms of CPU and RAM, thus ensuring also easy reproducibility.

Finally, we wanted to create a workflow that was as easy as possible. To this end, participants can access our infrastructure directly from the Web through a simple interface. This interface lets them monitor their quotas but also allows them to develop and execute their code directly from their browsers, without having to worry about installing anything on their machines or dealing with command-line tools.

This infrastructure has been implemented using Docker images orchestrated through Kubernetes. It is made up of several components that are interconnected together to provide both organizers and participants easy access to the needed resources, see Figure 1. All problems submitted by the participants were saved in a database to monitor their quotas and to gather data to draw statistics about the lab.

The final infrastructure was deployed on a *m6a.8xlarge* AWS EC2 instance equipped with an AMD EPYC 7R13 processor. Table 1 reports the specifications of the hardware resources corresponding to that instance and to each team's workspace. All participants were given the same monthly quota to use quantum resources. Table 2 reports the monthly quotas according to the two tasks.

Table 1: The hardware resources corresponding to the AWS EC2 instance and to the participants' workspaces.

Hardware resources.			
-	**CPU**	**RAM**	**Hard Drive**
Infrastructure	32 cores	128 GB RAM	1 TB HDD
Workspace	1200 millicores	10 GB RAM	20 GB HDD

Table 2: The monthly quotas to use quantum resources according to the tasks.

Monthly quotas for the tasks.			
Task	**March**	**April**	**May**
Task 1: Feature Selection	30 seconds	30 seconds	50 seconds
Task 2: Clustering	50 seconds	50 seconds	150 seconds

4.2 General guidelines

Each team has access to its personal area inside our infrastructure with the credentials that have been provided to them. All runs must be executed by using the workspaces that have been created for each one of the participating teams, thus ensuring a fair comparison and easy reproducibility.

All participants cannot exceed their given quotas (see Table 2) to execute problems on quantum devices. The quotas can be monitored by each participating team through a dashboard that is constantly being automatically updated, reporting usages of the different methods (i.e., QA, H, and SA) and some general statistics.

All participants' runs must follow the file formats that are already described in Section 3.1 and 3.2 to allow us running our evaluation tools smoothly.

Participants have also been asked to upload their files on their own Bitbucket git repositories to enhance reproducibility. Each repository has been created by us inside a Bitbucket project[8]. Their repositories have been kept private through the challenge but are now public.

5 Results

In this Section, we present the results achieved by the participants and we discuss their approaches. Out of the 26 registered teams, 7 teams managed to upload some final runs. In total, the number of runs is 65 considering both SA, QA, and H(H was introduced in Section 2.1). Table 3 reports the 7 teams that correctly participated and submitted some final runs.

In total, throughout the entire lab participants have submitted 976 problems. Specifically, 758 of them were solved with SA, while 199 were solved using QA

[8] https://bitbucket.org/eval-labs/workspace/projects/QCLEF24

Table 3: The teams who participated and submitted at QuantumCLEF 2024.

Team	**Affiliation**	**Country**
BIT.UA	IEETA/DETI, LASI, University of Aveiro	Portugal
CRUISE	RMIT University	Australia
NICA	Iran University of Science and Technology, Departement of Computer Engineering	Iran, Islamic Republic Of
OWS	Friedrich Schiller Universität Jena	Germany
qIIMAS	Universidad Nacional Autonoma de Mexico	Mexico
QTB	Universidad Tecnologica de Bolivar	Colombia
shm2024	Madras Christian College, Chennai	India

and 18 with the H method. The total execution time of SA has been almost 12 hours while the total QA and H execution time has been roughly 4 minutes.

The QA execution time in this whole Section refers to the *Annealing* phase as described in Section 2.1, therefore it includes the time required to program the QPU, sampling, and reading the result. The embedding time and network latencies are not taken into account and are left to be considered for possible future editions of the QuantumCLEF lab.

5.1 Task 1A

Here we present the results achieved by the teams participating in task 1A.

MQ2007 dataset. As it is possible to see in Table 4, teams considered different numbers of features in their submissions. In general, we can observe that most of the submissions achieve similar nDCG@10 values when considering a number of features that lies between 10 and 25. In fact, Figure 2 shows that for these runs the Tukey HSD test performed after the Two-Way ANOVA hypothesis test shows no significant differences. Instead, runs that consider only 5 features achieve nDCG@10 values that are significantly different (lower) with respect to the others. This is reasonable since by considering too few features, then there is a high information loss.

Figure 3 shows the nDCG@10 values and Annealing timings of the runs that used QA and SA. From this figure we can see that, in terms of efficiency (i.e., Annealing time), runs using QA required a shorter amount of time with respect to SA. On average, QA required ≈ 9.89 times less compared to SA, thus representing a more efficient alternative. Considering effectiveness, SA seems to be performing more consistently. However, on average it performs only ≈ 1.03 times better compared to QA.
Teams adopted different approaches to address this task:

- team **BIT.UA** [1] tried different QUBO formulations that involved the usage of different correlation-based measures such as Spearman coefficient, Pearson coefficient, and Mutual Information [9]. Furthermore, their approach also

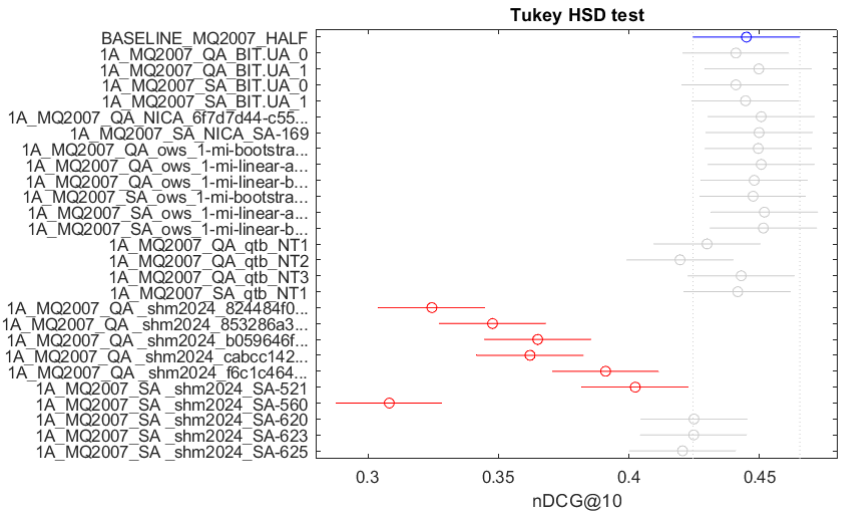


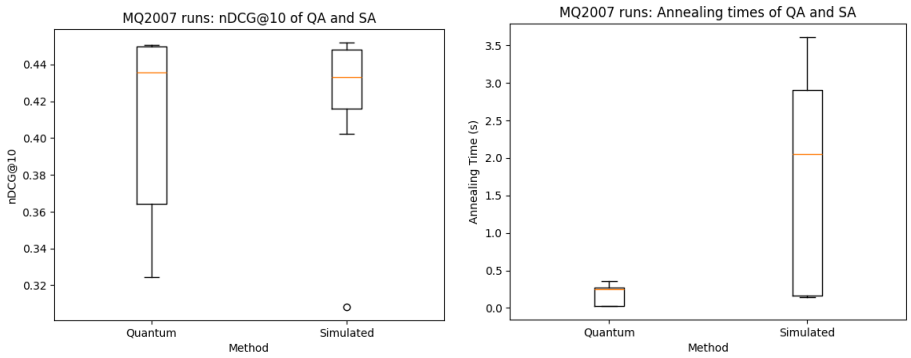
Fig. 2: The Tukey HSD test considering the nDCG@10 values associated with different runs and queries for the MQ2007 dataset.

involved the usage of a scaling factor to automatically balance the importance of the diagonal terms in the matrix Q with respect to the off-diagonal terms. Additionally, they also tried investigating some non-linear functions that adjusted the weights of the values returned by the correlation-based measures. The number of features chosen was decided by using a validation dataset approach with a custom LambdaMART model.

– team **NICA** [17] and team **shm2024** [13] used a QUBO formulation which involved the Mutual Information [9] as a correlation-based measure.

– team **QTB** [22] investigated different QUBO formulations involving different correlation-based measures (e.g., Mutual Information [9]). The team employed all methods (i.e., QA, H and SA), and the H approach allowed them to achieve a high score with only a few features thanks to its pre-processing and post-processing capabilities.

– team **OWS** [12] employed a QUBO matrix that was formulated using Mutual Information [9], in which some of its components were recalculated using the results achieved by a bootstrapping approach. In this way, the team recalculated the values associated with the diagonal components, the off-diagonal components, or both. The team focused on choosing only 25 features and the optimization of the number of considered features is left for future works.



(a) nDCG@10 of SA and QA runs (b) Annealing time of SA and QA runs

Fig. 3: The box plots of the nDCG@10 values and Annealing timings associated with the runs using QA and SA on the MQ2007 dataset.

Istella dataset. As it is possible to see in Table 5 and in Figure 4, also in this case teams considered different numbers of features in their submissions. However, for the Istella dataset, most of the runs are statistically different from each other because the number of features used varies a lot. It is interesting to see that the baseline method employing Recursive Feature Elimination considering 110 features performed much worse with respect to all participants' runs. Furthermore, running Recursive Feature Elimination to keep the top 110 features required a considerable amount of time (almost 2 hours of computation) and a considerable amount of RAM (24 GB), which is much higher than the teams' workspace specifications.

The teams adopted similar approaches to the ones described for the MQ2007 dataset to solve the Feature Selection task on the Istella dataset. However, since the dataset could not fit entirely in the QPU due to the high number of features, two teams decided to adopt the following pre-processing techniques:

- team **BIT.UA** [1] employed different approaches such as using a first stage SA approach to select only a subset of features or the manual elimination of features with high correlation values between them before solving the problem with QA.
- team **NICA** [17] kept only the 50 features that had the highest Mutual Information value towards the target variable, thus reducing the feature set.

Figure 5 shows the nDCG@10 values and Annealing timings of the runs that used QA and SA. From this figure we can see that, in terms of efficiency (i.e., Annealing time), also in this case runs using QA required a shorter amount of time with respect to SA. On average, QA required ≈ 10.45 times less compared to SA, thus representing a more efficient alternative. Similar considerations apply also for effectiveness. In fact, SA seems to be performing more consistently however, on average it performs only ≈ 1.03 times better compared to QA.

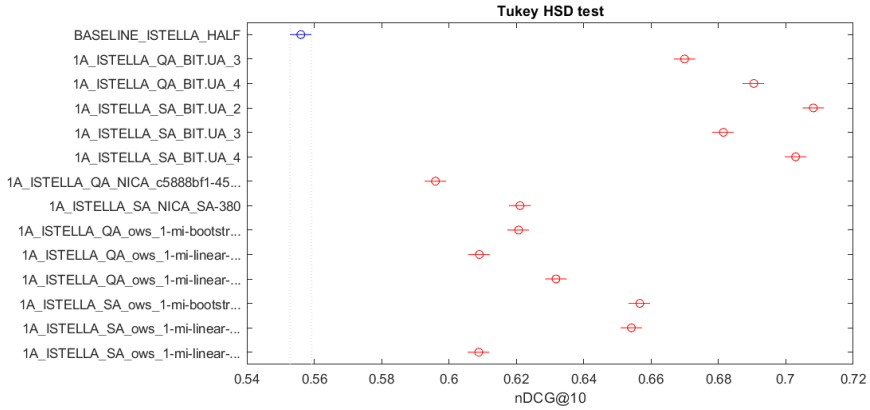
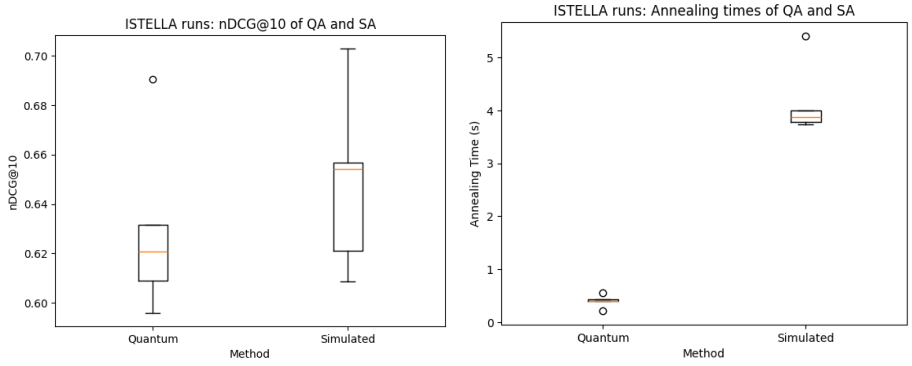


Fig. 4: The Tukey HSD test considering the nDCG@10 values associated with different runs and queries for the Istella dataset.

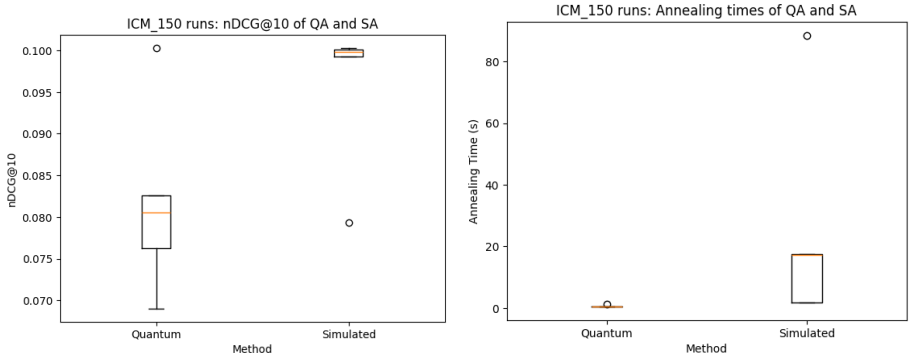


(a) nDCG@10 of SA and QA runs (b) Annealing time of SA and QA runs

Fig. 5: The box plots of the nDCG@10 values and Annealing timings associated with the runs using QA and SA on the Istella dataset.

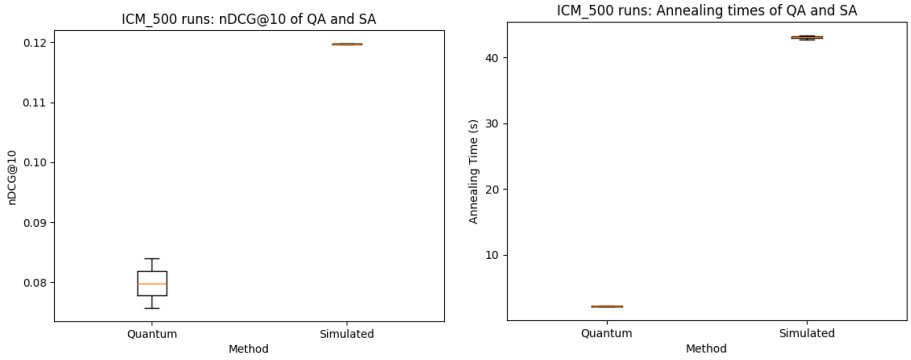
5.2 Task 1B

Here we present the results achieved by the two teams participating in task 1B. Results are divided according to the two feature sets. For both the small ICM (see Table 6) and the large one (see Table 7) the teams were able to improve the effectiveness of the baseline RS by a large margin, around 23% on the small set and 44% on the large one. Team **CRUISE** [19] especially achieved a large improvement by developing a counterfactual version of nDCG to enhance a feature selection method based on Mutual Information. The idea considers that Mutual Information does not account for the final goal of making recommendations.



(a) nDCG@10 of QA and SA runs (b) Annealing time of QA and SA runs

Fig. 6: The box plots of the nDCG@10 values and Annealing timings associated with the runs using QA and SA on the ICM_150 dataset.



(a) nDCG@10 of QA and SA runs (b) Annealing time of QA and SA runs

Fig. 7: The box plots of the nDCG@10 values and Annealing timings associated with the runs using QA and SA on the ICM_500 dataset.

The proposed approach is based on MIQUBO [9] and introduces a term in the diagonal of Q which represents the change in nDCG@10 obtained by removing each of the features individually, weighted by a scaling factor. In this way, the diagonal of Q includes both the Mutual Information between the feature values and the target label, as well as the weighted change in nDCG@10. For the small ICM, with 150 features, QA is 35.88 times faster than SA but it is 1.17 times worse in terms of nDCG@10 (see Figure 6).

For the large ICM, with 500 features, that could not fit on the QPU, team **CRUISE** [19] split the features into subsets small enough to be tackled by the

QPU. Then, the features selected in each subset have been merged into a final set of features. For this large ICM QA is 19.53 times faster than SA but it is 1.5 times worse in terms of nDCG@10 (see Figure 7). Note that the number of selected features is very different so this could play a role.

5.3 Task 2

Here we present the results achieved by the teams participating in task 2. Table 8 reports the results achieved in this task.

In this task, we can see that team **qIIMAS** managed to achieve higher results with respect to the baseline for each number of clusters considered. The approach adopted by team **qIIMAS** [2] consisted of employing the QUBO formulation proposed in a previous work [5]. Due to the high dimensionality of the dataset, they decided to first apply a traditional approach to reduce the number of points n to some representatives m where $m < n$. Then they performed the clustering approach on the m representatives in a hierarchical fashion, returning the final set of centroids and their associated n points. They investigated the usage of both QA, H, and SA.

In Figure 8 we can observe that there are no statistical differences among runs using H and runs using SA considering the nDCG@10 values achieved.

Figure 9 shows the Annealing time of the runs that used H and SA. From this figure we can see that, in terms of efficiency (i.e., Annealing time), runs using H required a shorter amount of time with respect to SA. On average, H required ≈ 21.75 times less compared to SA, thus representing a more efficient alternative. In addition, the H methods achieved slightly better results in terms of effectiveness, being ≈ 1.02 times better than SA on average.

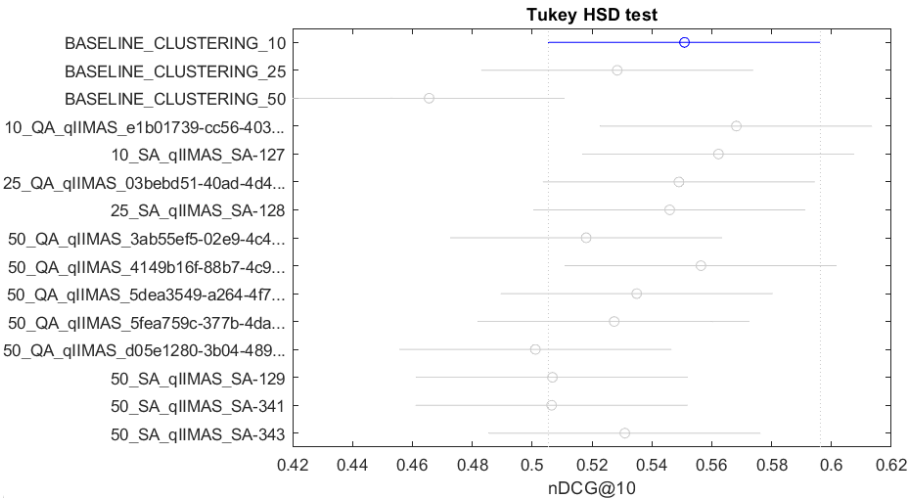
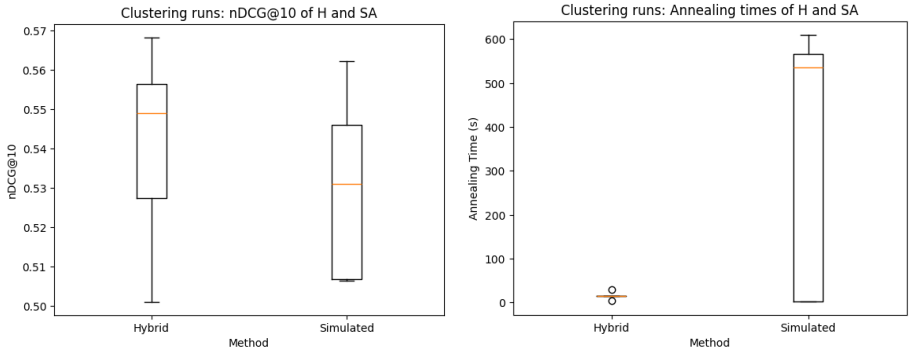


Fig. 8: The Tukey HSD test considering the nDCG@10 values associated with different runs and queries for the Clustering dataset.



(a) nDCG@10 of H and SA runs (b) Annealing time of H and SA runs

Fig. 9: The box plots of the nDCG@10 values and Annealing timings associated with the runs using H and SA on the Clustering dataset.

6 Conclusions and Future Work

In this paper, we have presented the overview of the first edition of the QuantumCLEF 2024 lab, the first lab at CLEF focusing on the study, development, and evaluation of QC algorithms.

This lab was composed of two tasks concerning the problems of Feature Selection and Clustering, specifically focused on IR and RS systems. An ad-hoc infrastructure was created to ease the participants' workflow and to grant them access to computational resources and the cutting-edge quantum annealers provided by D-Wave.

A total of 26 teams registered for the lab and 7 of them successfully managed to submit their runs. The results have shown that QA and H managed to achieve comparable results in terms of effectiveness with respect to SA while achieving a higher level of efficiency in terms of Annealing time. This shows that QC is starting to become a powerful technology that could help in the resolution of complex problems, especially in the future once it has matured enough.

This lab represented a great opportunity not only to develop and evaluate QC algorithms on **real quantum computers** (quantum technologies are still not easily accessible to the general public) but also to raise awareness of the potential of QC, which is likely to become a powerful technology in the future. The data obtained throughout the challenge has also been useful in preparing a new QC tutorial presented to the community at the international SIGIR conference 2024 [11]. Furthermore, participants were provided with comprehensive materials such as videos, slides, and examples that allowed them to learn how QC and QA work. Finally, we opted for maximum transparency, allowing participants to work with the actual D-Wave libraries without constraining them to use custom functions. In this way, participants familiarized themselves with the official D-

Wave libraries and, thus, are now able to program quantum annealers even outside our infrastructure to solve other problems in their research field.

In the future, we plan to organize a second edition of QuantumCLEF with different tasks and more challenges. We also plan to further improve the infrastructure according to the comments received by the participants through the lab to ensure a smoother experience for participants of a possible future edition of QuantumCLEF. Moreover, we would like to invest in a more powerful infrastructure that will grant access to more participants and that will provide more resources (in terms of CPU and RAM) to each workspace. In this way, it will be possible to consider even a more fair comparison between SA and QA. If possible, we would also like to extend the infrastructure to include a gate-based quantum computer [24], in addition to the already available quantum annealer.

Acknowledgments

We acknowledge the financial support from ICSC - "National Research Centre in High Performance Computing, Big Data and Quantum Computing", funded by the European Union – NextGenerationEU.

We acknowledge the CINECA award under the ISCRA initiative, for the availability of high-performance computing resources and support.

References

1. Almeida, T., Matos, S.: Towards a hyperparameter-free qubo formulation for feature selection in ir. In: Faggioli, G., Ferro, N., Galuščáková, P., García Seco de Herrera, A. (eds.) Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum (2024)
2. Alvarez Giron, W., Tellez, J., Tovar Cortes, J., Gómez Adorno, H.: Team qiimas on task 2 - clustering. In: Faggioli, G., Ferro, N., Galuščáková, P., García Seco de Herrera, A. (eds.) Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum (2024)
3. Arthur, D., Date, P.: Balanced k-means clustering on an adiabatic quantum computer. Quantum Inf. Process. **20**(9), 294 (2021), https://doi.org/10.1007/s11128-021-03240-8
4. Bauckhage, C., Piatkowski, N., Sifa, R., Hecker, D., Wrobel, S.: A QUBO formulation of the k-medoids problem. In: "Lernen, Wissen, Daten, Analysen", Berlin, Germany, CEUR Workshop Proceedings, vol. 2454, pp. 54–63, CEUR-WS.org (2019), URL https://ceur-ws.org/Vol-2454/paper_39.pdf
5. Bauckhage, C., Piatkowski, N., Sifa, R., Hecker, D., Wrobel, S.: A qubo formulation of the k-medoids problem. In: LWDA, pp. 54–63 (2019)
6. Bertsimas, D., Tsitsiklis, J.: Simulated annealing. Statistical science **8**(1), 10–15 (1993)
7. Burges, C.J.C.: From RankNet to LambdaRank to LambdaMART: An Overview. Tech. rep., Microsoft Research, MSR-TR-2010-82 (June 2010)

8. Cai, J., Macready, W.G., Roy, A.: A practical heuristic for finding graph minors. arXiv preprint arXiv:1406.2741 (2014)
9. Ferrari Dacrema, M., Moroni, F., Nembrini, R., Ferro, N., Faggioli, G., Cremonesi, P.: Towards Feature Selection for Ranking and Classification Exploiting Quantum Annealers. In: Proc. 45th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), pp. 2814–2824, ACM Press, New York, USA (2022)
10. Ferrari Dacrema, M., Pasin, A., Cremonesi, P., Ferro, N.: Quantum Computing for Information Retrieval and Recommender Systems. In: European Conference on Information Retrieval, pp. 358–362, Springer (2024)
11. Ferrari Dacrema, M., Pasin, A., Cremonesi, P., Ferro, N.: Using and Evaluating Quantum Computing for Information Retrieval and Recommender Systems (2024)
12. Fröbe, M., Alexander, D., Hendriksen, G., Schlatt, F., Hagen, M., Potthast, M.: Team openwebsearch at CLEF 2024: QuantumCLEF. In: Faggioli, G., Ferro, N., Galuščáková, P., García Seco de Herrera, A. (eds.) Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum (2024)
13. Gersome, S., Mahibha, J., Thenmozhi, D.: Team shm2024 on quantum feature selection. In: Faggioli, G., Ferro, N., Galuščáková, P., García Seco de Herrera, A. (eds.) Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum (2024)
14. Glover, F., Kochenberger, G., Hennig, R., Du, Y.: Quantum bridge analytics I: a tutorial on formulating and using QUBO models. Annals of Operations Research **314**, 141–183 (July 2022)
15. Hashemi, H., Aliannejadi, M., Zamani, H., Croft, W.B.: Antique: A non-factoid question answering benchmark. In: Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part II 42, pp. 166–173, Springer (2020)
16. Lucchese, C., Nardini, F.M., Orlando, S., Perego, R., Silvestri, F., Trani, S.: Post-learning optimization of tree ensembles for efficient ranking. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval, pp. 949–952 (2016)
17. Naebzadeh, A., Eetemadi, S.: Nica at quantum computing CLEF tasks 2024. In: Faggioli, G., Ferro, N., Galuščáková, P., García Seco de Herrera, A. (eds.) Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum (2024)
18. Nembrini, R., Ferrari Dacrema, M., Cremonesi, P.: Feature selection for recommender systems with quantum computing. Entropy **23**(8), 970 (2021)
19. Niu, J., Li, J., Deng, K., Ren, Y.: Cruise on quantum computing for feature selection in recommender systems. In: Faggioli, G., Ferro, N., Galuščáková, P., García Seco de Herrera, A. (eds.) Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum (2024)
20. Pasin, A., Ferrari Dacrema, M., Cremonesi, P., Ferro, N.: qCLEF: A Proposal to Evaluate Quantum Annealing for Information Retrieval and Recom-

mender Systems. In: International Conference of the Cross-Language Evaluation Forum for European Languages, pp. 97–108, Springer (2023)
21. Pasin, A., Ferrari Dacrema, M., Cremonesi, P., Ferro, N.: QuantumCLEF-Quantum Computing at CLEF. In: European Conference on Information Retrieval, pp. 482–489, Springer (2024)
22. Payares, E., Puertas, E., Martinez Santos, J.C.: Team qtb on feature selection via quantum annealing and hybrid models. In: Faggioli, G., Ferro, N., Galuščáková, P., García Seco de Herrera, A. (eds.) Working Notes of CLEF 2024 - Conference and Labs of the Evaluation Forum (2024)
23. Qin, T., Liu, T.Y.: Introducing letor 4.0 datasets. arXiv preprint arXiv:1306.2597 (2013)
24. Rieffel, E., Polak, W.: An introduction to quantum computing for non-physicists. ACM Computing Surveys (CSUR) **32**(3), 300–335 (2000)
25. Ushijima-Mwesigwa, H., Negre, C.F.A., Mniszewski, S.M.: Graph partitioning using quantum annealing on the d-wave system. CoRR **abs/1705.03082** (2017), URL http://arxiv.org/abs/1705.03082
26. Van Laarhoven, P.J., Aarts, E.H., van Laarhoven, P.J., Aarts, E.H.: Simulated annealing. Springer (1987)
27. Wu, Y., Cao, Q., Shen, H., Tao, S., Cheng, X.: INMO: A model-agnostic and scalable module for inductive collaborative filtering. In: SIGIR '22: The 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, Madrid, Spain, pp. 91–101, ACM (2022), https://doi.org/10.1145/3477495.3532000
28. Yarkoni, S., Raponi, E., Bäck, T., Schmitt, S.: Quantum annealing for industry applications: Introduction and review. Reports on Progress in Physics (2022)

A Task 1A - Team Results

Table 4: The results for Task 1A on the MQ2007 dataset. An adjacent couple of rows (marked with the same color) represents the results achieved with QA/H and SA using the same problem formulation. Results marked in yellow(●) refer to the baselines' results.

Team	Submission id	nDCG@10	Annealing time (ms)	Type	N° features
BIT.UA	1A_MQ2007_QA_BIT.UA_0	0.441	274	QA	18
BIT.UA	1A_MQ2007_SA_BIT.UA_0	0.441	1351	SA	16
BIT.UA	1A_MQ2007_QA_BIT.UA_1	0.4497	270	QA	20
BIT.UA	1A_MQ2007_SA_BIT.UA_1	0.4446	3607	SA	18
NICA	1A_MQ2007_QA_NICA_6f7d7d44-c559-4e36-9b10-b7e51e521036	0.4506	274	QA	17
NICA	1A_MQ2007_SA_NICA_SA-169	0.4498	3510	SA	15
OWS	1A_MQ2007_QA_ows_1-mi-bootstrap-mixture	0.4495	279	QA	25
OWS	1A_MQ2007_SA_ows_1-mi-bootstrap-mixture	0.4475	2818	SA	25
OWS	1A_MQ2007_QA_ows_1-mi-linear-and-quadratic-bootstrapped-boost-3	0.4506	270	QA	25
OWS	1A_MQ2007_SA_ows_1-mi-linear-and-quadratic-bootstrapped-boost-3	0.4519	2752	SA	25
OWS	1A_MQ2007_QA_ows_1-mi-linear-bootstrapped-boost-3	0.448	241	QA	25
OWS	1A_MQ2007_SA_ows_1-mi-linear-bootstrapped-boost-3	0.4515	2759	SA	25
QTB	1A_MQ2007_QA_qtb_NT1	0.4299	356	QA	13
QTB	1A_MQ2007_SA_qtb_NT1	0.4024	3174	SA	10
QTB	1A_MQ2007_QA_qtb_NT2	0.4195	5000	H	10
QTB	-	-	-	SA	-
QTB	1A_MQ2007_QA_qtb_NT3	0.443	4309	H	10
QTB	-	-	-	SA	-
shm2024	1A_MQ2007_QA _shm2024_b059646f-a9fd-4fd6-9589-c6e117400a9e	0.365	30	QA	5
shm2024	1A_MQ2007_SA _shm2024_SA-521	0.4024	284	SA	5
shm2024	1A_MQ2007_QA _shm2024_cabcc142-3fc5-4b22-8a6b-c7a45857fbc2	0.3621	27	QA	5
shm2024	1A_MQ2007_SA _shm2024_SA-560	0.3082	164	SA	5
shm2024	1A_MQ2007_QA _shm2024_f6c1c464-6dba-4a44-93b8-92ad6c4f60f9	0.391	29	QA	5
shm2024	1A_MQ2007_SA _shm2024_SA-620	0.4249	143	SA	5
shm2024	1A_MQ2007_QA _shm2024_853286a3-7f47-4de8-b0a0-247a65e6f6b6	0.3477	28	QA	5
shm2024	1A_MQ2007_SA _shm2024_SA-623	0.4248	147	SA	5
shm2024	1A_MQ2007_QA _shm2024_824484f0-b6fa-44b6-9bc7-0cb073db84e7	0.3245	29	QA	5
shm2024	1A_MQ2007_SA _shm2024_SA-625	0.4205	144	SA	5
BASELINE	ALL_FEATURES	0.4473	-	-	46
BASELINE	RFE HALF_FEATURES	0.4450	-	-	23

Table 5: The results for Task 1A on the Istella dataset. An adjacent couple of rows (marked with the same color) represents the results achieved with QA/H and SA using the same problem formulation. Results marked in yellow(●) refer to the baselines' results.

Team	Submission id	nDCG@10	Annealing time (ms)	Type	N° features
BIT.UA	1A_Istella_QA_BIT.UA_3	0.6699	16325	SA+QA	92
BIT.UA	1A_Istella_SA_BIT.UA_3	0.6814	19071	SA	90
BIT.UA	1A_Istella_QA_BIT.UA_4	0.6905	551	QA	82
BIT.UA	1A_Istella_SA_BIT.UA_4	0.7029	5404	SA	72
BIT.UA	-	-	-	SA	-
BIT.UA	1A_Istella_SA_BIT.UA_2	0.7081	13827	SA	161
NICA	1A_Istella_QA_NICA_c5888bf1-4549-418c-92b8-b7175c9185e4	0.596	427	QA	15
NICA	1A_Istella_SA_NICA_SA-380	0.6211	3998	SA	15
OWS	1A_Istella_QA_ows_1-mi-bootstrap-mixture	0.6207	215	QA	25
OWS	1A_Istella_SA_ows_1-mi-bootstrap-mixture	0.6566	3875	SA	25
OWS	1A_Istella_QA_ows_1-mi-linear-and-quadratic-bootstrapped-boost-3	0.609	394	QA	25
OWS	1A_Istella_SA_ows_1-mi-linear-and-quadratic-bootstrapped-boost-3	0.6541	3728	SA	25
OWS	1A_Istella_QA_ows_1-mi-linear-bootstrapped-boost-3	0.6317	402	QA	25
OWS	1A_Istella_SA_ows_1-mi-linear-bootstrapped-boost-3	0.6088	3785	SA	25
BASELINE	ALL_FEATURES	0.7146	-	-	220
BASELINE	RFE HALF_FEATURES	0.5560	-	-	110

B Task 1B - Team Results

Table 6: Task 1B results on the 150_ICM dataset. Adjacent row pairs (same color) show the results achieved with QA/H and SA for the same problem formulation. Results highlighted in yellow(●) refer to the baselines' results.

Team	Submission id	nDCG@10	Annealing time (ms)	Type	N° features
CRUISE	1B_150_ICM_QA_CRUISE_1	0.0805	536	QA	138
CRUISE	1B_150_ICM_SA_CRUISE_1	0.0998	1745	SA	140
CRUISE	1B_150_ICM_QA_CRUISE_2	0.0826	529	QA	136
CRUISE	1B_150_ICM_SA_CRUISE_2	0.0993	17358	SA	140
CRUISE	1B_150_ICM_QA_CRUISE_3	0.0690	531	QA	132
CRUISE	1B_150_ICM_SA_CRUISE_3	0.1001	1760	SA	140
CRUISE	1B_150_ICM_QA_CRUISE_4	0.0763	558	QA	133
CRUISE	1B_150_ICM_SA_CRUISE_4	0.0793	17387	SA	140
CRUISE	1B_150_ICM_QA_CRUISE_5	0.1003	1375	QA	144
CRUISE	1B_150_ICM_SA_CRUISE_5	0.1003	88395	SA	144
NICA	-	-	-	QA	-
NICA	1B_150_ICM_SA_NICA_SA-457	0.0895	12247	SA	145
BASELINE	ALL_FEATURES	0.0810	-	-	150

Table 7: Task 1B results on the 500_ICM dataset. Adjacent row pairs (same color) show the results achieved with QA/H and SA for the same problem formulation. Results highlighted in yellow(●) refer to the baselines' results.

Team	Submission id	nDCG@10	Annealing time (ms)	Type	N° features
CRUISE	1B_500_ICM_QA_CRUISE_1	0.0757	2287	QA	407
CRUISE	1B_500_ICM_SA_CRUISE_1	0.1196	43339	SA	450
CRUISE	1B_500_ICM_QA_CRUISE_2	0.0839	2123	QA	397
CRUISE	1B_500_ICM_SA_CRUISE_2	0.1198	42777	SA	450
BASELINE	ALL_FEATURES	0.0827	-	-	500

C Task 2 - Team Results

Table 8: Task 2 results. Adjacent row pairs (same color) show the results achieved with QA/H and SA for the same problem formulation. Results highlighted in yellow(●) refer to the baselines' results.

Team	Submission id	nDCG@10	DBI	Annealing time (ms)	Type	N° centroids
qIIMAS	10_QA_qIIMAS_e1b01739-cc56-4034-baa2-558a44e0bd65	0.5682	6.3121	14993	H	10
qIIMAS	10_SA_qIIMAS_SA-127	0.5622	6.6847	535516	SA	10
BASELINE	BASELINE_10	0.5509	7.9892	-	-	10
qIIMAS	25_QA_qIIMAS_03bebd51-40ad-4d45-b2a1-8d58b829afc6	0.549	5.3510	14995	H	25
qIIMAS	25_SA_qIIMAS_SA-128	0.546	5.3369	565985	SA	25
BASELINE	BASELINE_25	0.5284	6.1201	-	-	25
qIIMAS	50_QA_qIIMAS_4149b16f-88b7-4c90-b9cd-7bec94929898	0.5564	4.8032	14993	H	50
qIIMAS	50_SA_qIIMAS_SA-129	0.5068	4.7868	610068	SA	50
qIIMAS	50_QA_qIIMAS_5fea759c-377b-4dab-80d8-ca70ba205f02	0.5274	4.9537	29995	H	50
qIIMAS	50_SA_qIIMAS_SA-343	0.531	4.8112	2871	SA	50
qIIMAS	50_QA_qIIMAS_d05e1280-3b04-4897-aa0f-5d508d72d8e0	0.5011	5.0868	3994	H	50
qIIMAS	50_SA_qIIMAS_SA-341	0.5065	5.4160	2793	SA	50
qIIMAS	50_QA_qIIMAS_3ab55ef5-02e9-4c4d-9c7c-51cb56be3d9a	0.518	5.1842	18	QA	50
qIIMAS	-	-	-	-	SA	50
qIIMAS	50_QA_qIIMAS_5dea3549-a264-4f70-812a-ac52b2108663	0.5349	4.6978	67	QA	50
qIIMAS	-	-	-	-	SA	50
BASELINE	BASELINE_50	0.4656	5.3679	-	-	50