Optimizing the Selection of Recommendation Carousels with Quantum Computing

Maurizio Ferrari Dacrema
Politecnico di Milano
Italy
maurizio.ferrari@polimi.it

Nicolò Felicioni
Politecnico di Milano
Italy
nicolo.felicioni@polimi.it

Paolo Cremonesi
Politecnico di Milano
Italy
paolo.cremonesi@polimi.it

ABSTRACT

It has been long known that quantum computing has the potential to revolutionize the way we find solutions of problems that are difficult to solve on classical computers. It was only recently that small but functional quantum computers have become available on the cloud, allowing to test their potential. In this paper we propose to leverage their capabilities to address an important task for recommender systems providers, the optimal selection of recommendation carousels. In many video-on-demand and music streaming services the user is provided with a homepage containing several recommendation lists, i.e., carousels, each built with a certain criteria (e.g., artist, mood, Action movies etc.). Choosing which set of carousels to display is a difficult problem because it needs to account for how the different recommendation lists interact, e.g., avoiding duplicate recommendations, and how they help the user explore the catalogue. We focus in particular on the *adiabatic computing* paradigm and use the D-Wave quantum annealer, which is able to solve NP-hard optimization problems, can be programmed by classical operations research tools and is freely available on the cloud. We propose a formulation of the carousel selection problem for black box recommenders, that can be solved effectively on a quantum annealer and has the advantage of being simple. We discuss its effectiveness, limitations and possible future directions of development.

CCS CONCEPTS

• **Information systems** → **Collaborative filtering**; **Recommender systems**; • **General and reference** → **Evaluation**.

KEYWORDS

Recommender Systems; User Interface; Quantum Computing

ACM Reference Format:
Maurizio Ferrari Dacrema, Nicolò Felicioni, and Paolo Cremonesi. 2021. Optimizing the Selection of Recommendation Carousels with Quantum Computing. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27-October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3460231.3478853

1 INTRODUCTION

The idea to exploit quantum phenomena to perform computational tasks was formulated decades ago and widely studied theoretically. Broadly speaking, there are two main paradigms to build quantum computers (QC): the *gate model* in which a quantum state is evolved using a sequence of operations performed with quantum gates, and the *adiabatic model* which leverages the way quantum systems naturally tend to evolve towards a state of minimal energy. In recent years QCs have started to become available on the cloud to the research community. The D-Wave Advantage QC belongs to the adiabatic model and is able to solve NP-hard optimization problems via quantum annealing. The speedup offered by quantum annealing is still an open question, but a significant constant speedup was shown in practice for some cases [10]. Currently, quantum annealers are the only QCs powerful enough to solve realistic problems, e.g., feature selection for recommender systems [21], this is the reason we will use them in our experiments.

One of the advantages of quantum annealers is that they solve combinatorial optimization tasks that can be formulated with traditional operations research tools, making them simpler to use than other technologies. The selection of which recommendation lists to display in a carousel user interface, choosing from a pool of black box recommendation models, is among the problems that can be represented in that formulation. It is common for movie-on-demand and music streaming services to provide the users with a page containing several recommendation lists, each generated with a different strategy, e.g., TV Series, Latest Releases. These lists are usually referred to as *widgets*, *shelves* or as *carousels*. Selecting the optimal recommendation lists to display is a complex problem of industrial interest [3, 11, 27]. In this scenario the overall recommendation quality of the page does not depend only on each individual recommendation list but has to take into account how the lists complement each other [12, 13]. Clearly, it is not beneficial to show the user very similar recommendation lists, even though each of them individually may have high accuracy. Furthermore, since the purpose of a recommender system is to help the user exploration of the catalogue, focusing on accuracy alone is not sufficient and it is known that a set of diverse recommendations can improve user satisfaction [6, 28]. Few papers have been published proposing strategies to optimize the layout of a page with multiple carousels, mostly referring to industrial scenarios: Netflix [27], Spotify [16], Deezer [3] and Amazon Video [11].

In this paper we propose a formulation to select the optimal set of recommendation lists to display in a carousel interface, without assumptions on the models generating them, which can be solved effectively on the D-Wave quantum annealer, showing a potential application of this new technology. The formulation simplicity allows its extension and improvement along several directions. We release our source code on Github.

The rest of the paper is organized as follows. In Section 2 we describe the quantum annealing technology, in Section 3 we present the carousel setting, in Section 4 we propose our optimization model and in Section 5 we show the results. Finally, in Section 6 we draw the conclusions and discuss future works.

2 QUANTUM ANNEALING

Quantum annealing is a meta-heuristic that can be used to minimize a given objective function by leveraging quantum fluctuations and tunneling [2]. Compared with Simulated Annealing [1] it is not limited to thermal fluctuations and is better able to escape local optima. While it is possible to simulate quantum annealing on a classical computer, a more interesting option is to use a real physical device which naturally displays the needed quantum behavior and acts as a special-purpose solver. This is the idea behind the development of quantum annealers. In order to solve a problem one has to craft the corresponding energy landscape, known as the *Hamiltonian* [10] (i.e., its objective function), which will then be minimized by the quantum effects. An important aspect of this technology is that the Hamiltonian is equivalent to the *quadratic unconstrained binary optimization* (QUBO) formulation of optimization problems, which is as follows:

$$\begin{aligned} \min \quad & y = x^T Q x \\ & x \text{ binary} \end{aligned} \tag{1}$$

where $x \in \{0,1\}^n$ is a vector of n binary variables and Q is an $|n| \times |n|$ matrix that contains the parameters of the model. It is possible to easily describe in QUBO formulation many NP-complete and NP-hard optimization problems [15, 19].

Within the Quantum Processing Unit (QPU) of a quantum annealer qubits are connected to each other according to a certain graph topology. In this paper we will use the D-Wave Advantage quantum annealer, which has more than 5000 qubits in a graph topology called *Pegasus* where each qubit is connected to at most 15 others [5]. The Pegasus topology is rather complex but it is based on a simpler one called *Chimera*, in which qubits are grouped in bipartite graphs of 8 qubits called *unit cells*, each of them connected to the adjacent ones, as shown in Figure 1. In order to solve a problem represented in QUBO format it is necessary to embed it into the QPU graph. This *minor embedding* process requires two steps:

Embedding. The Q matrix can be interpreted as the adjacency matrix of the undirected problem graph G. Thus, given G and the quantum annealer topology P, the embedding step finds another graph G_{emb}, which is a subgraph of G such that G can be obtained from G_{emb} by contracting edges. A visual example is shown in Figure 2, where, in order to fit the triangular problem graph in the desired square topology, the embedding algorithm duplicates a node using two qubits to represent the same logical variable.

Parameter setting. This step adds additional constraints to the loss function to take into account the nodes and edges created during the embedding phase. For example, it will add equality constraints between qubits representing the same logical variable.

Although the D-Wave Advantage QPU has more than 5000 qubits, the maximum instance size of the problems that can solved on the

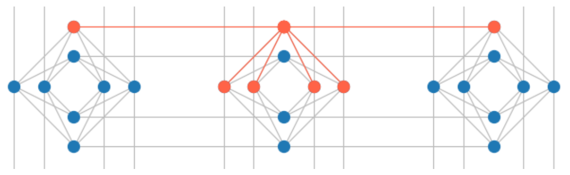
Figure 1: A portion of the Chimera graph which displays 3 unit cells. Each node is a qubit and each edge a connection. Each qubit has 6 connections, 4 within the same cell and 2 to different cells, here highlighted in red.

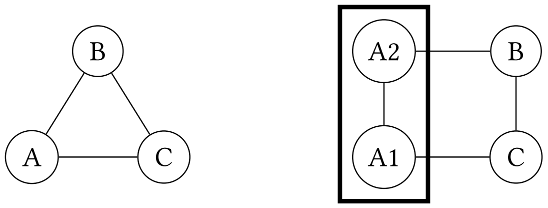

Figure 2: An example of how a problem graph can be embedded in a QPU topology. In this case, the triangular problem graph cannot fit directly into the square topology, therefore the problem variable *A* becomes a logical variable represented by two different qubits *A1* and *A2*, here highlighted.

QPU depends on the problem topology and its density. If the problem is very dense several physical qubits will be needed to connect the desired logical variables, reducing the problem size that can be embedded. In such cases, traditional operations research tools can be used to partition the QUBO problem in smaller instances that can fit on the QPU with a hybrid quantum-classical strategy. Given a problem represented in QUBO format and its embedding, it is possible to sample optimal solutions in tens of milliseconds.

3 CAROUSEL LAYOUT

The scenario where a carousel user interface is employed has certain distinguishing characteristics that must be taken into account [12]. First of all, the user interface is two-dimensional and will have a certain number carousels, see Figure 3. The carousels will be generated by different algorithms or by different providers. In general, there will be no global post-processing step to alter the provided recommendations based on the entire carousel layout. This means that a certain item may appear in more than one carousel.[1]

It is possible to evaluate the recommendation accuracy, e.g., using Precision, Recall, of a set of carousels by simply concatenating all recommendation lists and then applying a traditional single-list evaluation protocol. It is however important that each correct recommendation be counted only once when the item appears in multiple carousels [12]. An immediate consequence is that selecting the optimal set of carousels is not a trivial task and it becomes more complex as the number of carousels increases. The improvement in recommendation quality obtained by adding a certain recommendation list to a set of carousels will be given not by how many

[1]For instance, in the example shown in Figure 3, the TV series *Space Force* is present both in the *TV Comedies* and *New Releases* carousels.

correct recommendations it contains, but rather by the *new* correct recommendations that were not already contained in the other lists.

4 QUBO CAROUSEL SELECTION

In this section we describe the proposed *QUBO carousel selection* (QCS) strategy. The goal is to select the set of carousels which will optimize the accuracy of the overall recommendations while ensuring adequate diversity. The problem variables are defined as $x_m \in \{0,1\}, m \in M$, where M is the set of all available recommendation models, of cardinality $|M|$. The model will be selected if the corresponding x_m has value 1. We also define function $rec(i), i \in I$ which represents the number of times item i has been recommended. This information can be gathered easily because computing each model individual recommendation quality already requires to generate the recommended item lists.

We have now to define the coefficients of the QUBO problem. Given two models x and y, the diagonal components of matrix Q, q_{xx} and q_{yy}, which we call *bias*, refer to the impact of selecting those models individually (e.g., their individual recommendation quality) while the off-diagonal component q_{xy} represents the interaction between the two (i.e., how the recommendation lists complement each other). Intuitively, we want to select accurate models but we want to discourage the selection of models that provide recommendations that are *too similar*, to avoid showing the users the same items multiple times, with adverse effects on user satisfaction. In order to model this phenomena we propose the following accuracy and diversity heuristics:

Accuracy. This heuristic is based on the recommendation quality of each individual model measured with a metric of choice, in our experiments we use Precision computed on a validation set. The bias of each model is computed by scaling linearly its accuracy in the range $q_{xx} \in [0,1]$ where 1 is the recommendation quality of the best performing model. The interaction between two models is computed as the product of their biases $q_{xy} = q_{xx} \cdot q_{yy}$. This interaction will combine with the following Diversity heuristic.

Diversity. In order to account for how similar the recommendations generated by two models are, we rely on the total number of times each item has been recommended. We explore two types of diversity heuristics. The first type considers the *correlation* between the $rec(i)$ values associated to the two models, we use different correlations: Pearson, Spearman and Kendall τ. In those cases, highly correlated recommendation distributions should be discouraged, hence the corresponding q_{xy} will be the positive correlation value. The bias will be set to 1. A second approach considers the dispersion of the provided recommendations and uses a known metric to measure the diversity of recommendations, the Gini Index [29]. Since our aim is to encourage the selection of models which have different recommendation distributions, we compute the Gini index of the difference between the $rec(i)$ values of both models. Highly different recommendation distributions will result in both positive and negative numbers, hence a higher Gini Index. In this case the q_{xy} value will be the negative of the index and the bias will be 1.

Selection. The QUBO formulation, see Equation 1, does not allow hard constraints and requires them to be represented as penalties in the loss function. In order to achieve the selection of a certain number of models c, we introduce a penalty that will be zero only when exactly c models are selected:

$$k = \left(\sum_{m \in M} x_m - c \right)^2$$

The final formulation of QCS is built according to the loss function in Equation 1 including accuracy A, diversity D and selection penalty k, weighted with coefficients α and s:

$$\min \quad y = x^T \left(\alpha A + (1-\alpha) D \right) x + sk \qquad x \text{ binary}$$

5 EVALUATION

In order to evaluate the proposed carousel selection approach we report the results for two known movie recommendation datasets, a domain that tends to use the carousel user interface: Netflix and MovieLens 10M. *Netflix* [4] is a well known dataset from the *Netflix Prize.*[2] In order to reduce the computational time we randomly sampled 20% of the users, the resulting dataset contains 95k users, 17k items and 20M ratings. *MovieLens 10M* [17] is a popular dataset which contains 69k users, 10k items and 10M ratings, as well as item feature data such as tags and genre. Both datasets are split by randomly selecting 80% of interactions for the training set and 10% for both validation and test set. We follow the evaluation procedure described in [12] such that a correct recommendation is only counted once and in the *first* carousels it appears in. To allow the replicability of our results we release the source code for our experiments online[3] both datasets are publicly available and the D-Wave quantum Annealer is freely available on the cloud.

5.1 Selection Baselines

As previously observed, there are only few works that deal with the problem of model selection for a carousel interface. Usually, those works make assumptions on the recommendation models, leverage specific types of data such as session and context or require an online setting. In other cases, the selection of carousels is part of the recommendation model itself, which means it is not applicable in the scenario where the recommendation models are black boxes. In this work therefore we target a scenario where there are fewer degrees of freedom. In order to compare the proposed model selection approach, we report the following selection baselines:

Exhaustive. This baseline evaluates all possible model selections and chooses the one with the best Precision on the validation data. An exhaustive search is very computationally expensive as it corresponds to selecting the combinations without repetitions of c carousels within a pool of M models, which are $M!/c!(M-c)!$.

Individual Greedy. The models are selected according to their Precision on the validation data, taken with decreasing Precision values. This approach cannot account for duplicate recommendations and may select a set of models producing similar lists, but has a low computational cost as it requires to evaluate the M models only once.

[2]The data is publicly available here: https://www.kaggle.com/netflix-inc/netflix-prize-data

[3] https://github.com/qcpolimi/RecSys21_CarouselSelectionQuantumComputing

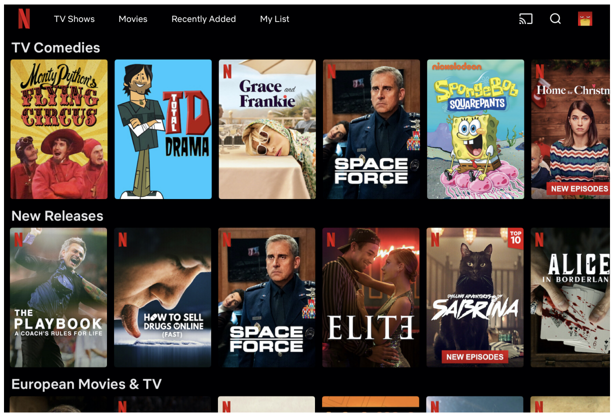

Figure 3: The homepage of Netflix, a popular video-on-demand streaming service, with multiple carousels.

Incremental Greedy. This baseline does not select the models based on a fixed accuracy value, but rather iteratively evaluates all candidate models accounting for those that have already been selected. Due to this, if a model has individually high accuracy but provides recommendations similar to those of the models already selected, it will provide few *new* correct recommendations and exhibit lower recommendation quality. This baseline is better suited to account for the characteristics of a carousel interface, but it is much more computationally expensive requiring to run $\sum_{i=1}^{c} M - i + 1$ model evaluations.

5.2 Available Recommendation Algorithms

The selection strategy proposed in this paper is designed to identify the optimal subset of recommendation models. In order to represent a realistic scenario where a multitude of options are available, we include several simple but well-known competitive algorithms [14].

Non-Personalized: TopPopular, GlobalEffects and Random.
Item-Based: ItemKNN and UserKNN [25], which compute the item or user similarity based on user interactions, with cosine similarity and shrinkage, and machine learning models SLIM [22], SLIM BPR and EASE^R [26].
Graph-based: P$^3\alpha$ [8] and RP$^3\beta$ [23], simple methods that simulate a random walk on a bipartite user-item graph.
Matrix Factorization: We included several known models, PureSVD [9], FunkSVD [14], Non-negative matrix factorization [7], MF BPR [24] and IALS [18].
Content-based and Hybrid: ItemKNN CBF, that builds the item similarities using the item features, and ItemKNN CF CBF in which the item ratings are concatenated with the item features [20]. Both use cosine with shrinkage.

We optimized all recommendation models following the best practices and hyperparameter ranges reported in [14], using a Bayesian search with 50 cases. Overall, the Movielens 10M dataset has 21 available recommender models: 15 collaborative, 3 content-based using genre, tags and both, and 3 hybrid content-collaborative KNNs. Netflix instead has 15, due to the absence of content features.

5.3 Results

The results are shown in Table 1. We can see that the number of possible carousel selections grows quickly and evaluating them all easily requires several weeks.[4] Comparing the greedy strategies we can see that the Incremental Greedy has a very high recommendation quality, very close to that of the Exhaustive search. The much lower recommendation quality of the Individual Greedy indicates how important it is to account for how the recommendations generated by the selected models complement or repeat each other. The Incremental Greedy strategy however has to perform a round of evaluations for each carousel to select, making it rather computationally expensive. To select all 9 carousels for the Movielens 10M dataset, the Incremental Greedy needs to evaluate 153 selections, each requiring 3 minutes. Even though the correct recommendations for each user and model can be precomputed, it is not a viable strategy in scenarios where we want to rapidly respond to user actions, for example by personalizing the layout in real time, as done in [27]. Furthermore, a greedy approach has limited flexibility in modeling more complex scenarios.

QCS is always competitive with the Individual Greedy strategy on Netflix, while on Movielens it is when combining both accuracy and diversity computed with the Gini Index. The optimal value for the α hyperparameter, optimized on the validation data, varied according to the number of carousels from 0.01 to 0.9. Using correlations on the number of recommendations per item, either the total number of recommendations or only the correct ones, did not prove effective. QCS is overall not able to outperform the Incremental Greedy strategy, although it comes within 1-3%, especially on Netflix and for small number of carousels on Movielens. One difference between the two strategies which may play a role is that the Incremental Greedy accounts for all user's recommendation lists while QCS relies only on the global item distribution. This suggests that future works should explore other scalable heuristic strategies to better model user wise effects. QCS is also significantly more scalable requiring fractions of a second on both Simulated Annealing and the D-Wave QC, compared to an hour for the Incremental Greedy, making it a viable strategy to rapidly respond

[4]We ran the experiments on an 8 core i7 3.5Ghz CPU, with 16GB RAM.

Table 1: Precision at 10 results for selection strategy and number of carousels. QCS uses the D-Wave Advantage quantum annealer. QCS results outperforming the Individual Greedy are highlighted in bold. Exhaustive search results on Movielens10M are missing for pages of 7 carousels or more as they required more than four weeks of computation.

Netflix Prize

		4	5	6	7	8	9
Total Selections (thousands)		1.3	3.0	5.0	6.0	6.0	5.0
Selection Method	Exhaustive	0.0907	0.0763	0.0681	0.0611	0.0552	0.0504
	Incremental Greedy	0.0878	0.0768	0.0681	0.0611	0.0552	0.0504
	Individual Greedy	0.0718	0.0630	0.0551	0.0499	0.0466	0.0440
	QCS Accuracy	**0.0803**	**0.0641**	**0.0580**	**0.0549**	**0.0519**	**0.0485**
	QCS Gini Index	**0.0852**	**0.0765**	**0.0676**	**0.0604**	**0.0534**	**0.0504**
	QCS Gini Index + Accuracy	**0.0845**	**0.0728**	**0.0667**	**0.0575**	**0.0506**	**0.0483**

Movielens 10M

		4	5	6	7	8	9
Total Selections (thousands)		5.9	20.3	54.2	116.2	203.4	293.9
Selection Method	Exhaustive	0.0829	0.0709	0.0619	-	-	-
	Incremental Greedy	0.0820	0.0698	0.0615	0.0547	0.0491	0.0445
	Individual Greedy	0.0774	0.0632	0.0543	0.0483	0.0422	0.0405
	QCS Accuracy	0.0697	**0.0649**	**0.0555**	0.0468	**0.0436**	0.0399
	QCS Gini Index	**0.0814**	0.0631	**0.0546**	**0.0508**	**0.0444**	**0.0428**
	QCS Gini Index + Accuracy	**0.0821**	**0.0680**	**0.0581**	**0.0522**	**0.0455**	**0.0422**

to user actions. Overall, this constitutes a promising result for a possible application of this emerging technology. As more variables are added to the model, e.g., personalized carousel layouts for different clusters of users, and the available QC hardware improves, the advantage of quantum annealing over other classical solvers will likely become more apparent.

6 CONCLUSIONS

In this paper we have proposed a strategy to select the optimal set of carousels to be displayed on a user interface of a movie-on-demand or music streaming service. The proposed formulation can be solved effectively on today's quantum annealers providing an example of possible application for this new technology. Overall, the experiments conducted show that quantum computers are becoming viable for applied research and can be used to solve small but realistic problems. Significant future works are the optimization of the relative ordering of the selected carousels, which requires the definition of a suitable heuristic, as well as including a form of personalization of the carousel layout.

REFERENCES

[1] Tameem Albash and Daniel A Lidar. 2018. Adiabatic quantum computation. *Reviews of Modern Physics* 90, 1 (2018), 015002.

[2] B Apolloni, N Cesa-Bianchi, and D De Falco. 1988. *A numerical implementation of "quantum annealing"*. Technical Report BIBOS-324. Bielefeld TU. Bielefeld-Bochum-Stochastik, Bielefeld. https://cds.cern.ch/record/192546

[3] Walid Bendada, Guillaume Salha, and Théo Bontempelli. 2020. Carousel Personalization in Music Streaming Apps with Contextual Bandits. In *RecSys 2020: Fourteenth ACM Conference on Recommender Systems, Virtual Event, Brazil, September 22-26, 2020*, Rodrygo L. T. Santos, Leandro Balby Marinho, Elizabeth M. Daly, Li Chen, Kim Falk, Noam Koenigstein, and Edleno Silva de Moura (Eds.). ACM, 420–425. https://doi.org/10.1145/3383313.3412217

[4] James Bennett, Stan Lanning, et al. 2007. The netflix prize. In *Proceedings of KDD cup and workshop*, Vol. 2007. New York, 35.

[5] Kelly Boothby, Paul Bunyk, Jack Raymond, and Aidan Roy. 2020. Next-generation topology of d-wave quantum processors. *arXiv preprint arXiv:2003.00133* (2020).

[6] Laming Chen, Guoxin Zhang, and Eric Zhou. 2018. Fast greedy map inference for determinantal point process to improve recommendation diversity. In *Advances in Neural Information Processing Systems*. 5622–5633.

[7] Andrzej Cichocki and Anh Huy Phan. 2009. Fast Local Algorithms for Large Scale Nonnegative Matrix and Tensor Factorizations. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* 92-A, 3 (2009), 708–721. https://doi.org/10.1587/transfun.E92.A.708

[8] Colin Cooper, Sang-Hyuk Lee, Tomasz Radzik, and Yiannis Siantos. 2014. Random walks in recommender systems: exact computation and simulations. In *23rd International World Wide Web Conference, WWW '14, Seoul, Republic of Korea, April 7-11, 2014, Companion Volume*, Chin-Wan Chung, Andrei Z. Broder, Kyuseok Shim, and Torsten Suel (Eds.). ACM, 811–816. https://doi.org/10.1145/2567948.2579244

[9] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of the 2010 ACM Conference on Recommender Systems, RecSys 2010, Barcelona, Spain, September 26-30, 2010*, Xavier Amatriain, Marc Torrens, Paul Resnick, and Markus Zanker (Eds.). ACM, 39–46. https://doi.org/10.1145/1864708.1864721

[10] Vasil S. Denchev, Sergio Boixo, Sergei V. Isakov, Nan Ding, Ryan Babbush, Vadim Smelyanskiy, John Martinis, and Hartmut Neven. 2016. What is the Computational Value of Finite-Range Tunneling? *Phys. Rev. X* 6 (Aug 2016), 031015. Issue 3. https://doi.org/10.1103/PhysRevX.6.031015

[11] Weicong Ding, Dinesh Govindaraj, and S. V. N. Vishwanathan. 2019. Whole Page Optimization with Global Constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 3153–3161. https://doi.org/10.1145/3292500.3330675

[12] Nicolò Felicioni, Maurizio Ferrari Dacrema, and Paolo Cremonesi. 2021. A Methodology for the Offline Evaluation of Recommender Systems in a User Interface with Multiple Carousels. In *Adjunct Proceedings of the 29th ACM Conference on User Modeling, Adaptation and Personalization*. 10–15.

[13] Nicolò Felicioni, Maurizio Ferrari Dacrema, Fernando Benjamín Pérez Maurera, and Paolo Cremonesi. 2021. Measuring the Ranking Quality of Recommendations in a Two-Dimensional Carousel Setting. In *Proceedings of IIR2021 - 11th Italian Information Retrieval Workshop*.

[14] Maurizio Ferrari Dacrema, Simone Boglio, Paolo Cremonesi, and Dietmar Jannach. 2021. A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research. *ACM Trans. Inf. Syst.* 39, 2, Article 20 (Jan. 2021), 49 pages. https://doi.org/10.1145/3434185

[15] Fred W. Glover, Gary A. Kochenberger, and Yu Du. 2019. Quantum Bridge Analytics I: a tutorial on formulating and using QUBO models. *4OR* 17, 4 (2019), 335–371. https://doi.org/10.1007/s10288-019-00424-y

[16] Alois Gruson, Praveen Chandar, Christophe Charbuillet, James McInerney, Samantha Hansen, Damien Tardieu, and Ben Carterette. 2019. Offline Evaluation to Make Decisions About PlaylistRecommendation Algorithms. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining, WSDM 2019, Melbourne, VIC, Australia, February 11-15, 2019*, J. Shane Culpepper, Alistair Moffat, Paul N. Bennett, and Kristina Lerman (Eds.). ACM, 420–428. https://doi.org/10.1145/3289600.3291027

[17] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.* 5, 4 (2016), 19:1–19:19. https://doi.org/10.1145/2827872

[18] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM 2008), December 15-19, 2008, Pisa, Italy*. IEEE Computer Society, 263–272. https://doi.org/10.1109/ICDM.2008.22

[19] Andrew Lucas. 2014. Ising formulations of many NP problems. *Frontiers in Physics* 2 (2014), 5. https://doi.org/10.3389/fphy.2014.00005

[20] Bamshad Mobasher, Xin Jin, and Yanzan Zhou. 2003. Semantically Enhanced Collaborative Filtering on the Web. In *Web Mining: From Web to Semantic Web, First European Web Mining Forum, EMWF 2003, Cavtat-Dubrovnik, Croatia, September 22, 2003, Revised Selected and Invited Papers (Lecture Notes in Computer Science)*, Bettina Berendt, Andreas Hotho, Dunja Mladenic, Maarten van Someren, Myra Spiliopoulou, and Gerd Stumme (Eds.), Vol. 3209. Springer, 57–76. https://doi.org/10.1007/978-3-540-30123-3_4

[21] Riccardo Nembrini, Maurizio Ferrari Dacrema, and Paolo Cremonesi. 2021. Feature Selection for Recommender Systems with Quantum Computing. *Entropy* 23, 8 (2021). https://doi.org/10.3390/e23080970

[22] Xia Ning and George Karypis. 2011. SLIM: Sparse linear methods for top-n recommender systems. In *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM '11)*. 497–506.

[23] Bibek Paudel, Fabian Christoffel, Chris Newell, and Abraham Bernstein. 2017. Updatable, Accurate, Diverse, and Scalable Recommendations for Interactive Applications. *ACM Trans. Interact. Intell. Syst.* 7, 1 (2017), 1:1–1:34. https://doi.org/10.1145/2955101

[24] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, Montreal, QC, Canada, June 18-21, 2009*, Jeff A. Bilmes and Andrew Y. Ng (Eds.). AUAI Press, 452–461. https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1630&proceeding_id=25

[25] Badrul Munir Sarwar, George Karypis, Joseph A. Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the Tenth International World Wide Web Conference, WWW 10, Hong Kong, China, May 1-5, 2001*, Vincent Y. Shen, Nobuo Saito, Michael R. Lyu, and Mary Ellen Zurko (Eds.). ACM, 285–295. https://doi.org/10.1145/371920.372071

[26] Harald Steck. 2019. Embarrassingly Shallow Autoencoders for Sparse Data. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 3251–3257. https://doi.org/10.1145/3308558.3313710

[27] Chao-Yuan Wu, Christopher V. Alvino, Alexander J. Smola, and Justin Basilico. 2016. Using Navigation to Improve Recommendations in Real-Time. In *Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016*, Shilad Sen, Werner Geyer, Jill Freyne, and Pablo Castells (Eds.). ACM, 341–348. https://doi.org/10.1145/2959100.2959174

[28] Mi Zhang and Neil Hurley. 2008. Avoiding monotony: improving the diversity of recommendation lists. In *Proceedings of the 2008 ACM Conference on Recommender Systems, RecSys 2008, Lausanne, Switzerland, October 23-25, 2008*, Pearl Pu, Derek G. Bridge, Bamshad Mobasher, and Francesco Ricci (Eds.). ACM, 123–130. https://doi.org/10.1145/1454008.1454030

[29] Tao Zhou, Zoltán Kuscsik, Jian-Guo Liu, Matúš Medo, Joseph Rushton Wakeling, and Yi-Cheng Zhang. 2010. Solving the apparent diversity-accuracy dilemma of recommender systems. *National Academy of Sciences* 107, 10 (2010), 4511–4515.