SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

TECHNICAL UNIVERSITY OF CRETE

DIPLOMA THESIS

AVOIDING CONTENT BUBBLES BY NETWORK-FRIENDLY RECOMMENDATION ALGORITHMS

AUTHOR : EVANGELIA TZIMPIMPAKI

THESIS COMMITTEE : PROF. THRASYVOULOS SPYROPOULOS

PROF. ATHANASIOS LIAVAS

PROF. GEORGIOS KARYSTINOS

CHANIA, FEBRUARY, 2024

"Your screen is increasingly a kind of one-way mirror
that reflects your own interests,
while the algorithm analysts observe everything you click on"

Pariser, 2017, p.13

ABSTRACT

Almost all online services encourage users to establish a profile, granting access to personalized content. Having more and more detailed data from the user, allows for the platforms to detect his interests and to create the content that has the greatest chance for success. However, there are instances when recommendations become excessively personalized, especially in (cache-friendly) systems also guiding suggestions towards content with low access cost. This can lead the user in a state where they are consistently presented with content of a singular nature, which may or may not sustain his interest in the long run. This thesis aims to improve recommendation systems, by increasing the diversity of recommended content, thus preventing the creation of content bubbles. First, an overview is provided, initiating with the exposition of Baseline Recommendation Systems (BS-RS), their evolution into Network-Friendly Recommendation Systems (NF-RS), and the representation of the content bubble phenomenon in NF-RS. The setup of BS-RS and NF-RS as optimization problems is detailed, and the introduced Diverse NF-RS is presented, addressing the content bubble phenomenon. The optimization problem for Diverse NF-RS is formulated, demonstrated to be convex, and linearized before being solved. No previously established implementation adequately addresses the diversity issue with comparable cost-diversity trade-offs. The proposed solution incorporates additional fairness metrics from other works, establishing that our proposed Recommendation System can accommodate them without compromising the favourable trade-offs achieved.

ΠΕΡΙΛΗΨΗ

Σχεδόν όλες οι διαδικτυακές υπηρεσίες ενθαρρύνουν τους χρήστες να δημιουργήσουν ένα προφίλ, παρέχοντάς τους έτσι πρόσβαση σε εξατομικευμένο περιεχόμενο. Αντλώντας συνεχώς λεπτομερή δεδομένα από το χρήστη, οι πλατφόρμες εντοπίζουν τα ενδιαφέροντά του, και συστήνουν στο χρήστη όλο και πιο πετυχημένο περιεχόμενο - δηλαδή, σχετικό με τις προτιμήσεις του. Ωστόσο, υπάρχουν περιπτώσεις όπου οι συστάσεις γίνονται υπερβολικά προσωποποιημένες, ειδικά στα συστήματα γνωστά ως 'φιλικά προς το δίκτυο'. Τέτοιου είδους συστήματα προσπαθούν να συστήσουν πετυχημένο περιεχόμενο, αλλά παράλληλα ωθούν και τους χρήστες προς περιεχόμενα με χαμηλό κόστος πρόσβασης (πχ. που βρίσκονται στην cache). Αυτό μπορεί να οδηγήσει το χρήστη σε μία κατάσταση όπου του παρουσιάζονται μόνιμα συστάσεις ενός συγκεκριμένου χαρακτήρα, διατηρώντας ή όχι το ενδιαφέρον του μακροπρόθεσμα. Η παρούσα διπλωματική εργασία στοχεύει στη βελτίωση των συστημάτων συστάσεων, μέσω της αύξησης της ποικιλομορφίας του προτεινόμενου περιεχομένου, αποτρέποντας έτσι τη δημιουργία του φαινομένου γνωστού ως 'φυσαλίδες περιεχομένου'. Ξεκινάμε με την παρουσίαση των τυπικών συστημάτων συστάσεων, την εξέλιξή τους σε 'φιλικά προς το δίκτυο' συστήματα συστάσεων, και την αναπαράσταση του φαινομένου των 'φυσαλιδών περιεχομένου' στα δεύτερα. Εισάγουμε τα 'ποικίλα, φιλικά προς το δίκτυο συστήματα συστάσεων', τα οποία στοχεύουν στην παράλληλη επίτευξη ικανοποιητικών συστάσεων, χαμηλού κόστους και υψηλής ποικιλομορφίας. Αφού διατυπώσουμε τη λειτουργία αυτών των συστημάτων ως πρόβλημα βελτιστοποίησης, αποδεικνύουμε ότι το πρόβλημα αυτό είναι κυρτό, και το γραμμικοποιούμε πριν το επιλύσουμε. Από όσο γνωρίζουμε, δεν υπάρχει αντίστοιχη υλοποίηση στη σχετική βιβλιογραφία η οποία να αντιμετωπίζει το ίδιο φαινόμενο επαρκώς, ενώ μάλιστα το σύστημα που δημιουργήσαμε αποδεδειγμένα επιφέρει πολύ καλή αντιστάθμιση κόστους-ποικιλομορφίας. Τέλος, το σύστημά μας επιτρέπει την ενσωμάτωση επιπλέον παραμέτρων (άλλων ερευνών), χωρίς να διακυβεύονται τα ευνοϊκά αποτελέσματα που επιτυγχάνει.

ACKNOWLEDGEMENTS

I would like to express my gratitude to my thesis supervisor **Prof. Thrasyvoulos Spyropoulos** for his guidance, encouragement and understanding. His mentorship played a pivotal role in steering me in the right direction during challenging times. Moreover, I am indebted to him for inspiring me to delve into a subject that has now become my passion. Furthermore, I would like to express my appreciation to my family for their unwavering support throughout all these years. Lastly, and of utmost importance, I would like to thank all my friends for being present throughout this journey in every possible way. You transformed this challenging experience into a more positive and enjoyable one. Special thanks are extended to Andreas and my study partners - Rania and Nikos - for providing me with courage when I had none. This thesis would not have been possible without the contribution of all those mentioned above.

Contents

1 Introduction

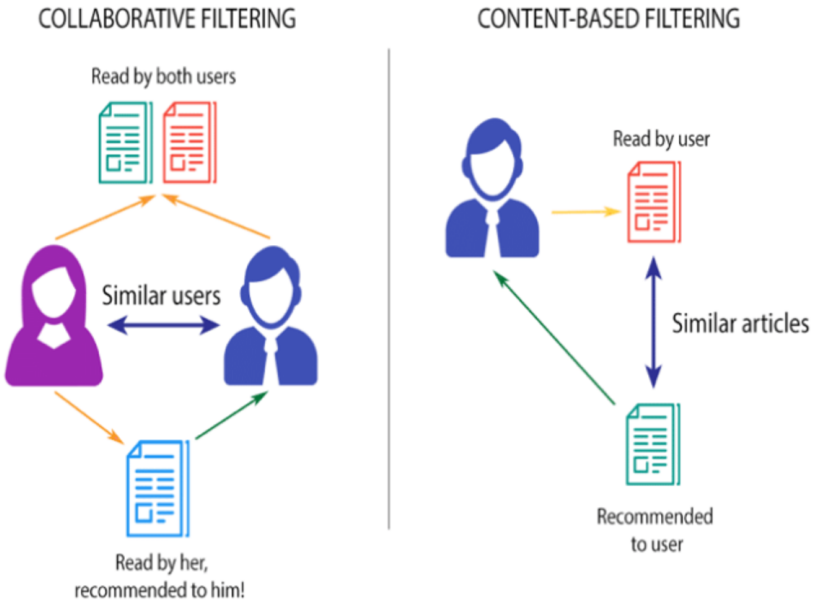
In the contemporary digital landscape, dominated by platforms like Netflix, TikTok, and YouTube, users find themselves immersed in an ever-expanding sea of content. Despite the sheer volume, the content recommendations provided by these platforms remain remarkably captivating, even addictive. The secret behind this allure lies in the sophisticated recommendation systems employed by these platforms. Uniquely tailored to individual preferences, these systems analyze user behavior and interactions to curate personalized content suggestions. The ongoing challenge is to continually enhance these recommendation algorithms, captivating users and maintaining their engagement. Our thesis aims to contribute to this evolution by delving into the intricacies of recommendation algorithms. Through this exploration, we seek to uncover innovative approaches that can elevate the user experience by offering even more compelling content recommendations.

1.1 Problem definition

A **recommendation system** (or recommender system - RS) is an artificial intelligence algorithm designed to predict consumer interests and suggest additional content accordingly. Typically, the suggestions refer to various decision-making processes, ranging from selecting products to purchase (eBay, Amazon etc), deciding on music to listen to (Spotify, YouTube etc), choosing online news to read, or finding interesting content on social media platforms (Instagram, Facebook, TikTok etc). There are also specialized recommendation systems tailored for specific topics like restaurants, online dating, research articles, advertisements and financial services.

Most Recommendation Systems follow the same steps during the recommendation process [47]. Firstly, user information is analyzed to create a comprehensive user profile. Next, based on the available information, the system selects the most suitable item(s) to present

to the user. The final step incorporates a feedback mechanism, allowing the Recommendation System (RS) to track user satisfaction and adjust the user model accordingly. There are two different approaches for the recommended item selection process: i) collaborative filtering and ii) content-based filtering. The **collaborative filtering** method relies on the assumption that individuals who shared preferences in the past are likely to continue doing so in the future, enjoying similar types of items as they have previously. The algorithm calculates similarity scores between users, and uses these scores to predict what items a target user is likely to be interested in. It identifies peer users (*user-based collaborative filtering*) or items (*item-based collaborative filtering*) with rating histories similar to the current user or item, creating recommendations based on this neighbourhood similarity. For example, if similar users wanted low cost items, the algorithm will suggest showing similar low-cost items to the current user. The collaborative filtering approach does not depend on machine-analysable content, allowing it to effectively recommend items without necessitating a deep understanding of the items themselves. Various algorithms are employed to measure user similarity or item similarity in recommendation systems (e.g. k-nearest neighbour, Pearson Correlation). Collaborative filtering approaches often suffer from three problems: cold start (for a new user/item, there is not enough data), scalability (there are millions of users and products, thus, a large amount of computation power is necessary to calculate recommendations), and sparsity (the large number of items means very few ratings for each item).



The **content-based filtering** method is a technique used to make predictions about a user's preferences based on the characteristics (genre, director, actor etc) of the items that the user has liked in the past. It can be applied to any type of item that has explicit

or implicit attributes, including books, movies, music, and products. The idea behind content-based filtering is that if a user has liked items with certain attributes in the past, they are likely to like items with similar attributes in the future. These methods mainly use techniques like Bayesian Classifiers, cluster analysis, decision trees, and artificial neural networks. A challenge with content-based filtering is whether the system can learn user preferences from one content source and apply them to other types. While suitable for traditional recommendation systems where the items are well-described by their attributes (e.g., books in Amazon, movies in Netflix), in different platforms there can be a lack of attribute descriptions for items. This limitation is addressed by most systems (like Netflix) using some form of a hybrid approach : a combination of collaborative filtering, content-based filtering, and other techniques to provide more accurate recommendations.

The quality of a platform's recommendation system significantly impacts its overall success. Users often seek recommendations when confronted with a vast array of items within a service. In such scenarios, Recommendation Systems play a crucial role by assisting them in discovering material they might not have found independently. This material aims to be highly appealing for the specific user, since effective recommended content not only satisfies the user, but also prolongs their stay on the platform, contributing to its success.

However, the pursuit of high-quality recommendations poses significant network costs. In addressing this issue, a recent approach known as network friendly recommendations has emerged. **Network Friendly Recommendation Systems (NF-RS)** extend their influence beyond user satisfaction to optimize network-level performance. The main focus of this shift is to reduce the network cost, while still maintaining appealing recommendations.

Numerous networking mechanisms can be explored to drastically reduce costs in Recommendation Systems, with **caching** standing out as a primary example. In our work, we leverage the concept known as **"cache-friendly recommendations"** ([2]-[20]). Unlike conventional recommendations that focus solely on intriguing content, our approach strategically guides suggestions towards content with both high interest and low access cost (i.e.

cached content). This methodology aims to keep the quality of recommendations above a certain threshold, while minimizing expenses in the recommendation process. Additionally, recommending a content that is almost as interesting to the user, but locally cached, might not just be "acceptable to the user, better for the network", but even beneficial to both the user and the network, if that content can be streamed for example at better quality [4]. So, the proposed NF-RS would act as following : *Instead of simply recommending interesting content, suggestions could be nudged towards interesting content which is also cached.* For instance, upon peak hours, the recommendation system would put higher preference on recommending content pre-cached in the vicinity of a user. Then, during an "off" period, the system would **update the cached content depending on users' interactions** of the day. Content Delivery Networks (CDNs) are mainly used for this type of caching; Amazon, Facebook, Netflix and many more platforms use them. Notably, Netflix has even designed its own CDN, known as OpenConnect, to maximize offload efficiency, minimize upstream demand on the network, and achieve higher speed in content delivery.

However, including the network cost as a determinant in the selection process for recommended items may **reduce diversity** within the recommendations provided by the system. In particular, highly popular contents (those frequently requested by users) will be mainly cached due to their higher engagement, ratings, and overall consumer interaction. Thus, the introduction of cache-friendly recommenders is likely to further concentrate recommendations around a small subset of items, namely those cached near the users. This specific kind of bias ultimately results in a decrease in diversity within the recommendations.

The above scenario becomes particularly evident when cached contents are also relative. For example, consider a system with three users: **A**, **B**, and **C**. Suppose there are three contents, and only one is recommended to each user at a time.

Interest of user **A** in the three contents: **1**, 0.8, 0.2
Interest of user **B** in the three contents: 0.8, **1**, 0.1
Interest of user **C** in the three contents: 0.3, 0.9, **1**

The BS-RS recommends the most interesting item to each user; item 1 for user **A**, item 2 for user **B**, and item 3 for user **C**. If only item 2 is cached close to all three users, then the NF-RS would recommend item 2 to all of them. By suffering only a minor reduction in users' **A** and **C** recommendation utility, the NF-RS increases the cache hit rate by 3 times. However, the diversity of recommended content to the pool of users is also reduced by 3 times - all 3 items were shown to someone, originally, but now only item 2 is shown.

This can be a concern both for the users (who see less content variety, in the long run), as well as the content creators of items 1 and 3. The main question we are trying to address therefore can be framed as follows: *Can we have the whole pie (large network cost reduction) and eat it too (maintain a satisfactory content diversity)?*

We define the phenomenon of reduced diversity (described above) as a **content bubble (or filter bubble)**. A content bubble represents a state of intellectual isolation wherein desired information might be occasionally omitted, progressively narrowing the users' exposure to diverse content. Within this confined space, consumers may experience feelings of boredom and **reduced satisfaction**, potentially leading them to leave the platform they were using. Beyond this dissatisfaction, content bubbles give rise to ethical concerns, since limiting exposure solely to some content might be perceived as **biasing opinions**. This issue is particularly relevant in platforms providing news content.

For instance, consider contents 1, 2, and 3 from the previous example to be political articles. Assume that content 2 refers to the current government, while contents 1 and 3 are related to the opposition. If all users are presented with the same content - namely the one related to the current government - there are two potential outcomes: (i) **negative user experience** for those interested in politics holding views contrary to those presented in content 2, and (ii) **Perceived Bias** or Political Direction: users may question the neutrality and objectivity of the recommendation system. The content bubble phenomenon has raised significant ethical concerns to many, including activist Eli Pariser, who expressed fear regarding the influence of media on directing people through recommendations:

Your identity shapes your media. There is just one flaw in this logic : Media also shape identity. And as a result, these services may end up creating a good fit between you and your media by changing... you.

It is important to note that we have performed preliminary measurements to confirm this initial suspicion about content bubbles being created in NF-RSs. However, this is a topic of another thesis, where this phenomenon is examined in detail.

1.2 Proposed solution

The objective of this thesis is to introduce an algorithm that ensures both network friendliness and content diversity. Specifically, our aim is to redefine the Network-Friendly Optimization Problem with the following objectives: i) maintain the core principle of minimizing network-related costs, without compromising the quality of recommendations (as in the original problem), and ii) reduce the formation of content bubbles.

The significance of this work arises from the limited exploration conducted by previous studies regarding the extent to which NF-R schemes introduce content similarity and whether it reaches a level that may be excessively unappealing for the user.

Conversely, content providers may impose explicit diversity requirements, such as restricting content similarity to no more than 50%. Currently, this option is not available in existing NF-R schemes, as diversity requirements have not been considered as a design aspect in NF-RS.

The contributions of this thesis include :

- the Diverse Network-Friendly Recommendation-System formulation
- proving it is a convex problem and hence, can be optimally solved
- performing a Linear Equivalent Transformation allowing fast(er) solution of the problem, without compromising optimality
- extensive simulations, using real datasets, that demonstrate that significantly better diversity-cost trade-offs can be achieved compared to the standard NF-RS. For instance, a 0.2% increase in network cost corresponded to a 9% increase in content diversity for the case of **5.3.9**.
- sensitivity analysis when introducing additional fairness metrics on top of diversity

Notably, we are not building a new recommendation algorithm, thus we do not compete with already existing platforms like Google, Netflix etc. Instead, we extract the items' relevance scores from these systems and try to address the content bubble problem.

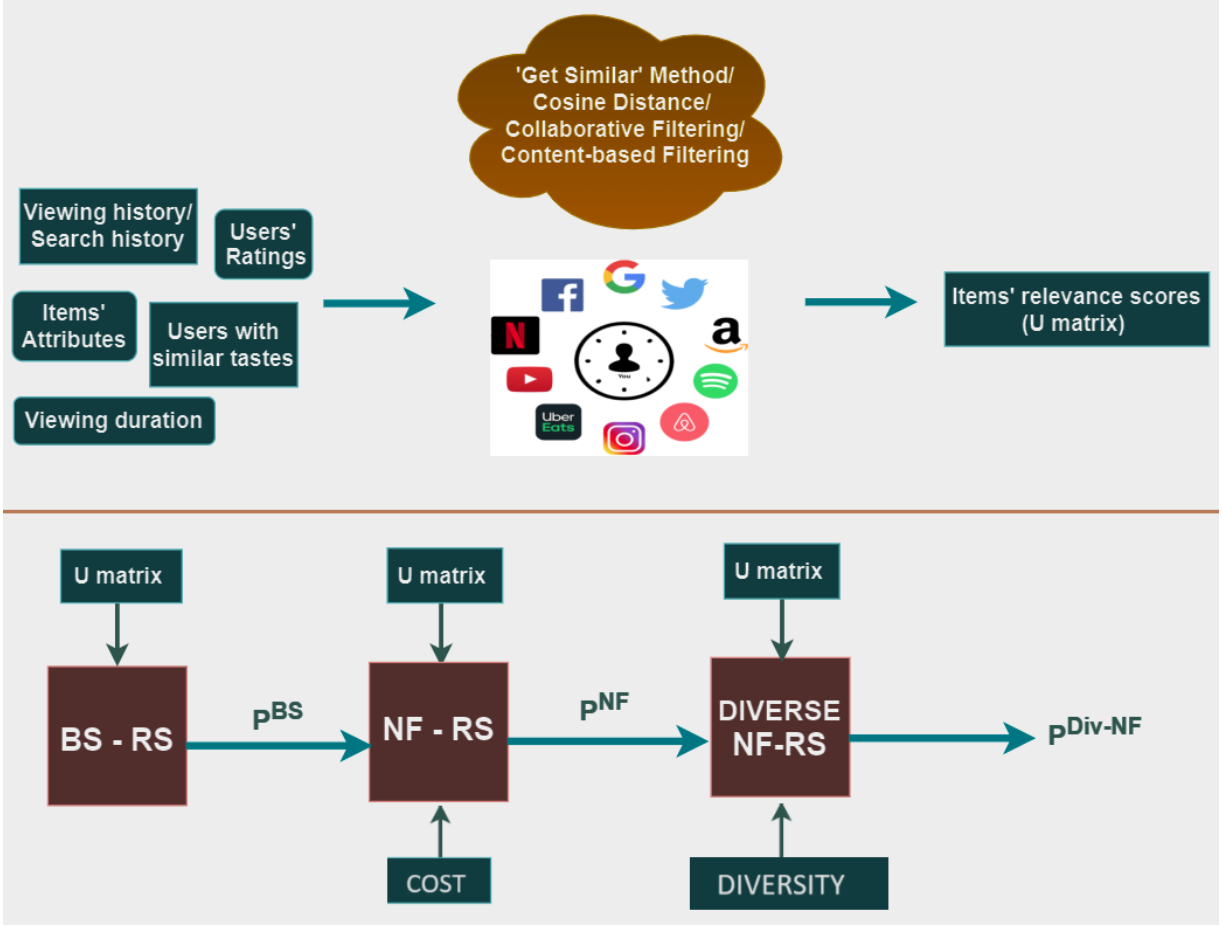


Figure 1. Evolution of Recommendation Systems. From BS, to NF, to Diverse NF.

2 Related work

Quality criteria of Recommendations. Recommendation Systems operate with the primary goal of maximizing customer satisfaction through the delivery of high-quality recommendations. The core strategy involves questioning how **relevant** the recommended content is to item(s) the user has previously viewed (and liked - if we have information regarding their ratings) [6],[42]. Various state-of-the-art data-driven methods, such as neighborhood-based methods [44], latent factorization methods [45], and cosine similarity [46], are employed to estimate relevance scores for all pairs of items. In our research, the relevance scores between items were obtained from the datasets of two real platforms: Last.fm and MovieLens, as defined precisely in [6]. Specifically, the Last.fm platform implements the 'getSimilar' method to derive the contents' relevance scores, while in the MovieLens platform [24] item-to-item collaborative filtering and cosine distance are applied to calculate the relevance between each pair of contents.

Network Friendly Recommendation Systems. The paradigm of NF-RS has been recently proposed and studied under different network set-ups and content services ([1]–[21]). The proposed NF-RS schemes aim to minimize the network cost by selecting "less costly" recommendations ([4]-[6],[20]) or by jointly designing the recommendation and network policy ([2],[3],[7]–[17]). To attain "less-costly" recommendations, NF-RSs employ efficient cache utilization, implementing various caching techniques.

Caching for Network-Friendly Recommendation Systems. Caching is the process of storing data in a cache - a temporary storage area that facilitates faster access to data - with the goal of improving system performance and minimizing the network cost. Although caching is extremely helpful for building less costly systems, studies have come to the result that ***caching policy alone is limited*** in the amount of performance gain it can bring [4], as the size of content catalogues is typically much larger than the storage capacity of a small cache. Consequently, a significant volume of requests is inevitably directed through the backhaul, regardless of the specific caching policy in place. Increasing cache capacity or backhaul capacity seem like the only way to cope with this problem, but these are hardware solutions involving significant costs. The following question then arises: *Are there any practical software-based solutions that can improve caching efficiency, at a low cost?* The answer to this question follows from two key observations: (i) the performance of a caching algorithm is dependent on user requests; (ii) user requests are increasingly driven by recommendation algorithms (e.g. 80% of Netflix's video views are through recommendations, while the corresponding percentage for YouTube is 50%). The proposal in [4] is therefore to not try to further improve what is stored at each cache, but rather to ***better exploit the already cached content*** by taking advantage of the recommendation algorithms integrated in the content services. For instance, upon peak hours (e.g. during the day) a recommendation system could put higher preference on recommending content **pre-cached** in the vicinity of a user (to a nearby cache) ([2]-[20]). Then, during an "off" period (e.g. at night), the system updates the cached content depending on users'

interactions of the day. This model has been assumed in plenty of related work, and is for example how Netflix operates its caches. Dynamic caching is quite more complicated and will be examined in future work. For now, we will follow the example of [4].

Fairness in Recommendation Systems. Various articles in scientific literature address the topic of fairness in Recommendation Systems. Specifically, [38] explores the kinds of fairness, as : **item fairness** (treating items fairly instead of prioritizing some of them), **user fairness** (treating users fairly instead of recommending based on demographic information) and **joint fairness** (satisfies both). Whether the allocation is fair can affect the user's and the provider's experience. If, for example, the recommendation cannot be fair to users, then the platform may lose users with specific interests. If the recommendation treats different items unfairly, then the providers of these discriminated items may leave the platform. In [39] users' age was proved to affect the recommendations in a music platform, while in [40], the gender bias was investigated in recommending career-related items.

Despite the many approaches on fairness made by the RS community ([26]–[36], [38]-[41]), our primary focus has been on a distinct form of fairness : forging a Network-Friendly RS that maintains **fairness towards the Baseline RS** (i.e., any standard RS). This approach aims to strike a balance between cost efficiency and a fair transition to a more network-friendly RS. The design of the specific Fair NF-RS has been implemented in [36]; a framework we have adopted in our work. The desired level of fairness in [36] is determined by an input variable, denoted as *fairness weight*. Fairness captured the deviation between $\mathbf{p^{BS}}$ (Baseline content demand) and $\mathbf{p^{NF}}$ (Network-Friendly content demand); thus, a generic measure, denoted as F, was employed, where $F = f(\mathbf{p^{BS}}, \mathbf{p^{NF}})$. The function f is defined according to the requirements of the content provider. In [36], three fairness measures were considered (which were then linearised, as shown on **Table 1**). We include the formulas here for completeness, as we will use them later in our work:

F-max : $F_{max} = max_{i \in K} |p_i^{NF} - p_i^{BS}|$ relates to the individual fairness and accounts for the "worst case", i.e., no content has a demand difference larger than F_{max}.

F-TV : $F_{TV} = \frac{1}{2} \cdot \sum_{i \in K} |p_i^{NF} - p_i^{BS}|$ is the total variation distance between the two distributions, i.e., the average change in content demand. It allows more flexibility than F_{max} in shaping the demand, since it does not impose a constraint for every single content.

F-KL : $F_{KL} = \sum_{i \in K} p_i^{BS} \log(\frac{p_i^{BS}}{p_i^{NF}})$ is the Kullback–Leibler (KL) divergence, a widely used measure for the difference between distributions. F_{KL} is more sensitive to changes in contents with lower demand p_i^{BS} .

The previously discussed fairness metrics, while occasionally mitigating content bubbles **as a side-effect**, do not consistently address the diversity problem independently. This limitation arises because these fairness constraints are designed to maintain recommenda-

tions closely aligned with the Baseline Recommendation System (BS-RS). However, our current objective diverges from this by placing emphasis on diversity.

For instance, consider a RS with a content catalogue of 9 items, out of which only one is being recommended each time a user engages with content. Suppose that the user is currently consuming item 1 and that the BS-RS sequentially recommends the items {7, 4, 9, 8} in the next 4 recommendations. Now, assume that our proposed diverse solution is {2, 6, 5, 3}. Despite offering the same amount of diversity as the BS-RS (i.e., 4 different items recommended in the total 4 recommendations, resulting in 100% diversity for both systems), not a single recommended item in our diverse system matches those in the BS recommendations. Thus, according to the discussed fairness metrics, the system may be deemed unfair, but for us, it is fair in terms of diversity.

Certainly, establishing a system with recommendations closely resembling those of the BS-RS may exhibit similar diversity in its recommendations. However, the primary distinction in our proposed solution lies in the possibility of finding an alternative solution with lower network costs and good QoR, featuring slightly different recommended items from the BS-RS, while maintaining equally high diversity with that of the BS-RS.

Table 1: Set of linear fairness constraints $\mathbf{S}\left(\mathbf{z}, \mathbf{p}^{\mathrm{NF}}\right)$.

$$\mathbf{F}_{\max}: \begin{array}{l} p_i^{BS} - p_i^{NF} \leq c_f \\ p_i^{NF} - p_i^{BS} \leq c_f \end{array} \quad \forall i \in \mathcal{K}$$

$$\mathbf{F_{TV}}: \sum_{i \in K} z_i \leq c_f$$
$$p_i^{BS} - p_i^{NF} \leq z_i \quad \forall i \in \mathcal{K}$$
$$p_i^{NF} - p_i^{BS} \leq z_i \quad \forall i \in \mathcal{K}$$

$$\mathbf{F_{KL}}: \sum_{i \in \mathcal{K}} p_i^{BS} \cdot z_i \geq -\left(c_f - \sum_{i \in \mathcal{K}} p_i^{BS} \log\left(p_i^{BS}\right)\right)$$
$$z_i \leq e^{(m-1)\cdot s} \cdot p_i^{NF} - (m-1)s - 1, \quad \forall i \in \mathcal{K}, m \in \{1, \ldots, M\}$$

Table 2: Important Notation.

$\mathcal{K}$	content catalog size ($\lvert\mathcal{K}\rvert = K$)
c_f	fairness weight $c_f \in [0, 1]$
$\mathbf{p}$	content demand ; $\mathrm{p} = [p_1, \ldots, p_K]$ and $\sum_{i \in \mathcal{K}} p_i = 1$
$\mathbf{p^{BS}}$	Baseline System content demand
$\mathbf{p^{NF}}$	Network-Friendly System content demand
$\mathbf{z}$	auxiliary variable for linear transformation of the constraints
s	sampling step for the KL-constraint linearisation
M	number of linear cuts for the KL-constraint linearisation

The ineffectiveness of the max fairness constraint is not only intuitive (it compares individual content items rather than the entire content set) but is also substantiated in Section **5.4.2**, where even a highly restrictive max fairness constraint (e.g., a fairness weight of 0.1) proves inadequate in achieving content variety comparable to the baseline scenario. The TV and KL constraints could potentially serve as better substitutes, since they study the cumulative behavior of all the items being recommended. However, our study proved that this is not always true and could maybe require some extra conditions (i.e. extremely low fairness weight). Thus, we conclude in that we need an explicit reformulation of the optimization problem to specifically target the formation of content bubbles.

Diversity in Recommendation Systems. The concept of insufficient diversity in recommended content is well-established in the related literature, often denoted as filter/-content/information bubbles. In [23], an examination of YouTube's video recommendation algorithm explores potential biases and the emergence of information bubbles. The research presents evidence of recommendation bias contributing to the formation of tightly-knit content communities, resulting in the establishment of narrative-specific clusters. The study outlined in [25] categorizes datasets by language, topics, etc., and investigates the impact of bias on fairness, diversity, and exposure in social media platforms. Other related works acknowledge metrics with a similar meaning, such as serendipity. In recommender systems, serendipity refers to the ability to surprise users with relevant and novel recommendations they might not have considered otherwise. Serendipitous recommendations have the potential to stimulate curiosity, creativity, and diversity, by providing a "breadth" in suggestions. In our context, we use the term content bubbles to denote the potential "concentration" of the recommended contents' probability mass function.

3 Problem Setup

We examine a content service that incorporates a recommendation system (RS) within its (web/mobile) platform. In this scenario, when a user engages with content - such as watching, listening, or reading - a list of recommendations is provided by the RS, suggesting additional content for the user to consume next.

We assume the user engages with one or more contents during a session, selected from a catalogue of size K. During the consumption of content $i \in K$, a list of N new contents is recommended to him, and he may respond in one of the following ways:

- follows recommendations with some fixed probability $\mathbf{a} \in (0, 1)$, and picks **one** of the N recommended contents. The item selection process can be random (as *we* do in our work, following the approach in [36]), or it can be based on the position of the item (as in [6],[12]). The choice made by the user depends on the model employed, allowing for generalization to other models in future research. For example, [6] proposes a modeling variation where the user selects out of the N recommendations the most relevant item to what they just viewed.

- ignores the recommendations with probability $1 - \mathbf{a}$, and picks a content j (e.g., through a search bar) with probability $p_{0j} \in (0, 1), \mathbf{p_0} = [p_{01}, p_{02}, ..., p_{0K}]^T$.

We define the demand $p0_j$ for a content j as the fraction of requests that are for this content : $p0_j = \frac{\text{number of requests for content j}}{\text{number of total requests}}$. Thus, we denote $\mathbf{p0} = [p0_1, ..., p0_K]^T$ as the vector with the distribution of demand for each content of the catalogue.

Notably, **p0** also serves as the entry point for each session.

Specifically, **p0** follows a Zipf distribution with a size K and a Zipf parameter equal to **popularity** - an input of the problem, defined in $[0, 1]$. In the case where this Zipf parameter is zero, the distribution becomes uniform (i.e. choosing with an equal probability for all contents). In other cases, the Zipf distribution would mean choosing several contents with higher probability, based on their position. For example, with a Zipf parameter of 1 and $K = 5$, **p0** would be: $\mathbf{p0} = [0.4379, 0.2189, 0.1459, 0.1094, 0.0876]^T$, signifying that the user is more inclined to select content 1, followed by content 2, and so forth.

The modeled session described above captures a number of everyday scenarios, such as watching clips on YouTube ([23],[25]) or TikTok, or engaging with personalized radio. Here, **a** represents the average probability of the user following recommendations. For instance, a value of $\mathbf{a} = 0.5$ was measured for YouTube [16], while Netflix reported a value of 0.8 [17], and a value of $\mathbf{a} = 1$ in the case of AutoPlay.

Content Retrieval Cost: We assume that retrieving content i is linked to a generic cost $c_i \in \mathbb{R}$, where $\mathbf{c} = [c_1, c_2, \ldots, c_K]^T$. This cost, known to the content provider, may depend on factors such as access latency, congestion overhead, popularity, file size, or monetary cost. We assign $c_i = 0$ for all cached content and $c_i = 1$ for non-cached content.

Content Relation Matrix U: Each element $u_{ij} \in [0, 1]$ in this matrix represents a score indicating the level of relevance between content $i \in K$ and content $j \in K$. These scores are known to the RS and are typically obtained from state-of-the-art algorithms implemented by existing recommendation platforms. In our case, the u_{ij} scores were extracted from the Last.FM platform and the MovieLens platform. The quality of these scores directly influences the performance of our recommendation system. However, it's important to note that the process of obtaining them is independent of our optimization framework; our focus lies in using these scores to optimize the recommendation process.

Using all the problem inputs initialized above, we now proceed to define our optimization problem (control variables, constraints and objective function).

Control Variable R: The square $K \times K$ recommendation matrix, over which we optimize. Each element $r_{ij} \in [0, 1]$ in this matrix represents the probability that content $j \in K$ is recommended after a user watches content $i \in K$.

3.1 Baseline Recommendation System (BS-RS)

The Baseline Recommendation System (BS-RS) is any standard Recommendation System. For each content $i \in K$, the BS-RS consistently recommends the N items with the highest u_{ij} values. In essence, the objective of the BS-RS is to optimize the quality of recommendations (QoR), without considering the associated cost.

Thus, the control variable $\mathbf{R}^{BS}$ will be a $K \times K$ matrix with 1's at the positions corresponding to the highest u_{ij} values. We call this **"top N"** policy.

For every content i, BS-RS accomplishes a quality : $q_i^{max} = \sum_{j=1}^{K} r_{ij}^{BS} \cdot u_{ij}$

We aim to design a recommendation policy $\mathbf{R}$, distinct from $\mathbf{R}^{BS}$, that takes into account both the relation matrix $\mathbf{U}$ and the retrieval costs c_i of all contents in the catalog, to address network requirements. In the formulation of our Network-Friendly Recommendation System below, the objective will be minimizing the total cost, while ensuring a certain level of quality of recommendations.

3.2 Network-Friendly Recommendation System (NF-RS)

Network-Friendly Recommendation Systems aim to minimize the total cost of retrieving desired contents, while achieving a Quality of Recommendations comparable to the quality level offered by the BS-RS.

To guarantee the quality of recommendations, the following set of K inequality constraints must be satisfied : $\sum_{i=1}^{K} r_{ij}^{NF} \cdot u_{ij} \geq q \cdot q_i^{max}, \forall i \in K$

In this equation, q is a parameter of the RS determining the percentage of q^{max} quality we wish to offer. When $q \to 0$, QoR is low, and the RS recommends based solely on access cost. Conversely, if $q \to 1$, the RS is the Baseline RS, and it cannot improve network cost. This thesis focuses on values of q $> 50\%$, as we aim to maintain a substantial level of recommendation quality without significant degradation to achieve our goal.

3.2.1 Objective Function of the NF-RS OP

The objective function of the NF-RS optimization problem (OP) pertains to the expected cost per user session. To derive this cost, we model the user request process as an Absorbing Markov Chain (AMC), as outlined in [6].

The user request process comprises periods during which the user follows recommendations (S_R), interspersed with steps during which the user ignores recommendations. Each S_R period can be represented by an absorbing Markov chain, which can conclude at any step with a probability of $1 - \mathbf{a}$ (denoted as an additional absorbing state). The transition

matrix $\mathbf{P}$ (shown bellow) of size $(K+1)\times(K+1)$ models the above Markov chain, where the transient part $\mathbf{Q} = \frac{\alpha}{N}\cdot\mathbf{R}$ corresponds to the user following recommendations according to our control variable $\mathbf{R}$. When the S_R period concludes, the process "renews", and the user re-enters the catalog from the same initial distribution $\mathbf{p0}$, so each S_R period is i.i.d.

$$\mathbf{P} = \left(\begin{array}{c|c} \frac{\alpha}{N}\cdot\mathbf{R} & \begin{array}{c}1-\alpha\\ \vdots \\ 1-\alpha\end{array} \\ \hline 0 \quad \cdots \quad 0 & 1 \end{array}\right)$$

Matrix P. The matrix modeling our Absorbing Markov Chain.

An example of such user session, driven by a RS, is depicted bellow. A user follows recommendations (continuous arrows) or ignores them (dotted arrows).

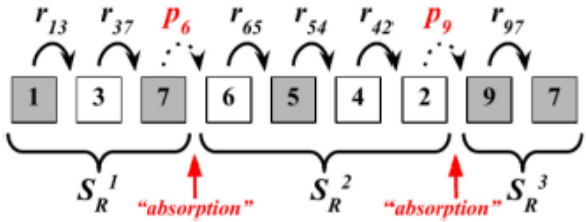


Figure 2. Example of a multi-content user session driven by a Recommendation System.

The aforementioned Absorbing Markov Chain is utilized to calculate the expected cost per recommendation period S_R. Then, the renewal reward theorem [48] is employed, to determine the expected cost of the entire session - which forms the objective of our optimization problem.

Following the proof on [6], the long term expected cost (LTEC) for a long user session, given the recommendation matrix $\mathbf{R}$, is :

$$LTEC = (1-\alpha)\,\mathbf{p}_0^T\cdot\left(\mathbf{I}-\tfrac{\alpha}{N}\cdot\mathbf{R}\right)^{-1}\cdot\mathbf{c}, \quad [6]$$

So, the objective of our optimization problem will be to minimize the $LTEC$. As mentioned earlier, the control variable is the recommendations matrix $\mathbf{R}$. Now, we shall identify the constraints that must be satisfied, and formulate our optimization problem.

3.2.2 NF-RS OP Formulation

The Network-Friendly Optimization Problem (NF-RS OP) described above can be formulated as : Minimize the LTEC (8a)*, by selecting the recommendations $\mathbf{R}$ (control variable) which satisfy the following equality and inequality constraints :

(i) Achieve high quality (8b)*,

(ii) Consist of exactly N recommendations for each content (8c)*, and

(iii) Conform to the definition of control variables (8d)*, where they:
- are probabilities defined in the range $[0, 1]$, and
- never allow the recommendation of a content immediately after it was consumed.

(*) : (8a), (8b), (8c) and (8d) refer to the optimization problem bellow.

NF-RS OP

$$\underset{\mathbf{R}}{\text{minimize}} \quad \frac{\mathbf{p}_0^T \cdot \left(\mathbf{I} - \frac{\alpha}{N} \cdot \mathbf{R}\right)^{-1} \cdot \mathbf{c}}{\frac{1}{1-\alpha}} \qquad (8.a)$$

$$\text{subject to} \quad \sum_{j=1}^{K} r_{ij} \cdot u_{ij} \geq q \cdot q_i^{max}, \quad \forall i \in \mathcal{K} \qquad (8.b)$$

$$\sum_{j=1}^{K} r_{ij} = N, \quad \forall i \in \mathcal{K} \qquad (8.c)$$

$$0 \leq r_{ij} \leq 1 (i \neq j), \quad r_{ii} = 0 \qquad (8.d)$$

Table 3: Important Notation.

$\mathcal{K}$	content catalog size ($\lvert\mathcal{K}\rvert = K$)
$\mathcal{N}$	Number of recommendations
a	Prob. the user follows recommendations
q	Percentage of original quality
q_i^{max}	Maximum baseline quality of content i
r_{ij}	Prob. to recommend j after viewing i
u_{ij}	Similarity scores for content pairs {i,j}
c_i	Access cost for content i $c_i \in \{0, 1\}$
p0	Baseline popularity of contents

In order to establish that the NF-RS OP presented above can be solved, it is imperative to demonstrate its convexity. Bellow, we will verify the convexity of the optimization

problem; if it is not convex, it will be transformed into an equivalent convex problem.

3.2.3 Non-convexity of NF-RS OP

Definition — *Convex Optimization Problem*

A **convex optimization problem** is a problem where all equality constraints are affine functions, all inequality constraints are convex functions, and the objective is a convex function if minimizing - or a concave function if maximizing.

Definition — *Convex Function*

A function $f(\mathbf{x})$ is convex iff it has a convex $domf$ and $\forall \mathbf{x_1}, \mathbf{x_2} \in domf(\mathbf{x})$, and $\forall \lambda \in [0,1]$, the following condition holds : $f(\lambda \mathbf{x_1} + (1-\lambda)\mathbf{x_2}) \leq \lambda f(\mathbf{x_1}) + (1-\lambda) f(\mathbf{x_2})$

In simpler terms, this means that the line segment between any two points on the graph of the function lies above or on the graph of the function, and not below it.

▶ Linear functions are also convex.

The NF-RS OP comprises K^2 variables r_{ij}, and a set of $K^2 + 2 \cdot K$ linear equality and inequality constraints (they are all linear to the optimization variable r_{ij}). Thus, the feasible solution space is convex.

However, the optimization problem is **non-convex** because the objective function is non-convex. Specifically, the objective contains the inverse of a matrix, which is not a convexity-preserving operator.

Namely, some convexity-preserving operations are:

- Non-negative combinations : $f(\boldsymbol{x}) = \sum_{i=1}^{m} c_i f_i(\boldsymbol{x})$
- Composition with affine functions $f(\boldsymbol{x}) = g(\boldsymbol{A}\boldsymbol{x} + \boldsymbol{b})$
- Composition of convex/concave functions $f(\boldsymbol{x}) = h\left(g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})\right)$
- Pointwise maximum/supremum $f(\boldsymbol{x}) = \sup_{i \in I} f_i(\boldsymbol{x})$

Hence, it is necessary to transform the non-convex problem into an equivalent convex one.

3.2.4 Convexifying NF-RS OP

The NF-RS optimization problem is transformed into a convex problem, as demonstrated in [6]. Specifically, in [6], a set of K auxiliary variables is introduced, and they are equated to $\mathbf{z^T} = (1-\alpha) \cdot \mathbf{p_0^T} \cdot (\mathbf{I} - \frac{a}{N} \cdot \mathbf{R})^{-1}$.

Thus, the initial objective function : $\underset{\mathbf{R}}{\text{minimize}} \; \frac{\mathbf{p}_0^T \cdot \left(\mathbf{I} - \frac{\alpha}{N} \cdot \mathbf{R}\right)^{-1} \cdot \mathbf{c}}{\frac{1}{1-\alpha}}$

transforms into : $\underset{\mathbf{z},\mathbf{R}}{\text{minimize}} \quad \mathbf{z}^T\mathbf{c}$ (or equivalently : $\underset{\mathbf{z},\mathbf{R}}{\text{minimize}} \quad \mathbf{c}^T\mathbf{z}$).

Intermediate Step (Equivalent formulation)

$$\underset{\mathbf{z},\mathbf{R}}{\text{minimize}} \quad \mathbf{c^T z} \qquad (8.a)$$

$$\text{subject to} \sum_{j=1}^{K} r_{ij} \cdot u_{ij} \geq q \cdot q_i^{max}, \quad \forall i \in \mathcal{K} \qquad (8.b)$$

$$\sum_{j=1}^{K} r_{ij} = N, \quad \forall i \in \mathcal{K} \qquad (8.c)$$

$$0 \leq r_{ij} \leq 1 (i \neq j), \quad r_{ii} = 0 \qquad (8.d)$$

$$\mathbf{z^T} - \frac{a}{N} \cdot \mathbf{z^T R} = (1-\alpha) \cdot \mathbf{p_0^T} \qquad (8.e)$$

Although the objective is now linear in the variable $\mathbf{z}$, this modification gave rise to a non-convex constraint $(8e)$: $\mathbf{z^T} - \frac{a}{N} \cdot \mathbf{z^T R} = (1-\alpha) \cdot \mathbf{p_0^T}$ (quadratic in $\mathbf{z,\ R}$).

The above formulation falls under the umbrella of non-convex quadratically constrained quadratic program. In this context, it is common to execute a convex relaxation of the quadratic constraints, subsequently solving an approximate convex problem. Nevertheless, we will follow the methodology used in [6], where an additional variable transformation is introduced. Specifically, we define variables $f_{ij} = r_{ij} \cdot z_i$.

So, the non-convex constraint now becomes:

$$z_j - \tfrac{a}{N} \textstyle\sum_{i=1}^{K} r_{ij} \cdot z_i = (1-\alpha) \cdot p_{0j}, \; \forall j \epsilon K$$

$$\Rightarrow z_j - \tfrac{a}{N} \textstyle\sum_{i=1}^{K} f_{ij} = (1-\alpha) \cdot p_{0j}, \; \forall j \epsilon K$$

which is evidently linear - and thus convex.

The meaning of previous optimization variable $\mathbf{R}$ and newly introduced optimization variables $\mathbf{z}$ and $\mathbf{F}$ are studied bellow :

Matrix $\mathbf{R}$: Each element r_{ij} of $\mathbf{R}$ denotes the probability to recommend content $j \in K$ conditioned on the fact that the user is at content $i \in K$.

Matrix $\mathbf{F}$: Each element f_{ij} of $\mathbf{F}$ denotes the percentage of time (in the long run) the user was at i and saw j in his RS list.

Vector $\mathbf{z}$: Each element z_i of $\mathbf{z}$ scaled by $(1-\alpha)$ expresses the long-term probability that item $i \in K$ is requested, i.e. the content demand of NF-RS, i.e. $\mathbf{p^{NF}}$.

Lemma 1

The variable transformation $f_{ij} = z_i \cdot r_{ij}$ is a one-to-one mapping between(z_i, r_{ij}) and (z_i, f_{ij}), given that all contents have a non-zero probability of being requested by the user [6].

The proof of the Lemma goes as following:

• To obtain r_{ij}, one must calculate f_{ij}/z_i. The sole constraint in this context is that z_i must be strictly non-zero to avoid an undefined r_{ij}.

• However, given $f_{ij} \geq 0$ (this condition is satisfied, as f_{ij} represents a percentage value) and $p_{0i} > 0, \forall i \in K$, the variable $z_j = \frac{a}{N} \sum_{i=1}^{K} f_{ij} + (1-\alpha) \cdot p_{0j}$ is compelled to be strictly positive, ensuring it is never zero.

• Consequently, r_{ij} is always uniquely defined, provided that $p_{0i} > 0, \forall i \in K$.

Following the aforementioned steps, we arrive at our **convex NF-RS OP**, utilizing the optimization variables **z, F**. Instead of $\mathbf{z}$, we will now employ the (more accurate) symbolization $\mathbf{p^{NF}}$, as mentioned earlier.

The (now) convex NF-RS OP is depicted bellow.

3.2.5 Convex NF-RS OP

$$\underset{\mathbf{p^{NF}},\ \mathbf{F}}{\text{minimize}} \quad \mathbf{c}^T\mathbf{p^{NF}}, \quad (13.a)$$

subject to

Original constraints

$$\sum_{j=1}^{K} f_{ij} \cdot u_{ij} - p_i^{NF} \cdot q \cdot q_i^{BS} \geq 0, \forall i \in \mathcal{K} \qquad (13.b)$$

$$\sum_{j=1}^{K} f_{ij} - N \cdot p_i^{NF} = 0, \forall i \in \mathcal{K} \qquad (13.c)$$

$$f_{ij} - p_i^{NF} \leq 0 \, \forall i, j \in \mathcal{K} \qquad (13.d)$$

$$f_{ij} \geq 0 \, (i \neq j), \; f_{ii} = 0 \qquad (13.e)$$

Constraint for transformation to convex(auxiliary variable F)

$$p_j^{NF} - \frac{\alpha}{N} \cdot \sum_{i=1}^{K} f_{ij} = p_{0j}, \forall j \in \mathcal{K} \qquad (13.f)$$

3.2.6 Linear NF-RS OP

Notably, the above convex OP **is also linear** to the optimization variables $\mathbf{F}, \mathbf{p^{NF}}$.

This linearity of the Convex NF-RS OP allowed the writers of [6] to solve the NF-RS as a Linear Programming (LP) problem (by using the CPLEX linear solver).

Definition — ***Linear programming***

Linear programming deals with the maximization (or minimization) of a linear objective function, subject to linear constraints. The linear objective and constraints must consist of linear expressions.

Definition — ***Linear expression***

By **linear expression** we mean an expression which is a scalar product : $\sum a_i x_i$. Values a_i represent constraints (i.e. data) and x_i represent variables (i.e unknowns).

Such an expression can also be written in short form as a vector product : $\mathbf{A^T X}$, where $\mathbf{A}$ is the vector of constants and $\mathbf{X}$ is the vector of variables. A *linear constraint* is expressed by an equality or inequality; strict inequality operators ($>$ or $<$) are not allowed in linear constraints.

So, a **symbolic representation of an LP** would be :

$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \sum_{i=1}^{N} c_i x_i \\ \text{subject to} \quad & a_{11}x_1 + a_{12}x_2 ... + a_{1n}x_n \geq b_1 \\ & a_{21}x_1 + a_{22}x_2 ... + a_{2n}x_n \geq b_2 \\ & ....a_{m1}x_1 + a_{m2}x_2 ... + a_{mn}x_n \geq b_m \\ & x_1, x_2, ..., x_n \geq 0 \end{aligned}$$

There are three benefits in solving a Linear Program instead of the convex one:

- Optimality guarantees: the LP-based implementation will never return a sub-optimal solution.
- No need for parameter tuning: For instance, the performance of a heuristic ADMM implementation - like that of [4] - depends on carefully selecting the parameter μ (the penalty on the quadratic term). On the contrary, the CPLEX (linear solver) has no need of tuning.
- The execution time of the LP-based solution is lower.

Now that we have formalized our Linear optimization problem, we are ready to address the diversity issue highlighted in the Introduction. However, before delving into this matter, it is prudent to first address some additional fairness constraints outlined in the Related Work. These fairness constraints will be optional in our optimization problem; i.e., we may choose not use them at times, but in other instances, we may incorporate them to draw additional conclusions.

3.2.7 Other Fairness metrics in NF-RS

In related work ([36]) some **optional fairness constraints** were also included in NF-RS. Specifically, [36] captures fairness as the deviation between the content demand of the BS-RS and that of the NF-RS; thus, the Fair NF-RS system aims to maintain fairness towards the BS-RS. Although the fairness constraints were initially nonlinear, their linear approximation was derived in [36], enabling us to incorporate them without further adjustments.

The set of linear fairness constraints $\mathbf{S}\left(\mathbf{z}, \mathbf{p}^{\mathrm{NF}}\right)$ and their detailed description are available in the Related Work section. Below, we formulate the Linear Fair NF-RS OP by incorporating the fairness constraints $\mathbf{S}\left(\mathbf{z}, \mathbf{p}^{\mathrm{NF}}\right)$ into the Linear NF-RS OP.

Linear Fair NF-RS OP

$$\underset{\mathbf{p^{NF}},\ \mathbf{F}}{\text{minimize}} \quad \mathbf{c}^T\mathbf{p^{NF}}, \quad (13.a)$$

subject to

Original constraints

$$\sum_{j=1}^{K} f_{ij} \cdot u_{ij} - p_i^{NF} \cdot q \cdot q_i^{BS} \geq 0, \forall i \in \mathcal{K} \qquad (13.b)$$

$$\sum_{j=1}^{K} f_{ij} - N \cdot p_i^{NF} = 0, \forall i \in \mathcal{K} \qquad (13.c)$$

$$f_{ij} - p_i^{NF} \leq 0 \, \forall i, j \in \mathcal{K} \qquad (13.d)$$

$$f_{ij} \geq 0 \, (i \neq j), \; f_{ii} = 0 \qquad (13.e)$$

Constraint for transformation to convex(auxiliary variable F)

$$p_j^{NF} - \frac{\alpha}{N} \cdot \sum_{i=1}^{K} f_{ij} = p_{0j}, \forall j \in \mathcal{K} \qquad (13.f)$$

Linear constraints for fairness

$\mathbf{S}(\mathbf{z}, \mathbf{p^{NF}})$

After formulating the Linear (Fair) NF-RS OP, we move on to address the content bubble phenomenon in the next section, which comprises our methodology and contributions.

4 Problem Solving

In order to avoid content bubbles created in NF-RS, an additional constraint was introduced to the Linear NF-RS optimization problem. This supplementary constraint articulates the intention to provide content recommendations in a more "uncertain" manner.

4.1 Entropy as a metric of diversity

In mathematical terms, **entropy** serves as a measure of uncertainty, and thus, the new constraint could be formulated as *maintaining high entropy.*

Definition — ***Entropy***

The entropy of a random variable $X \in \mathcal{X}$ is the average level of "uncertainty" inherent to the variable's possible outcomes, i.e. $H(X) := -\sum_{x \in \mathcal{X}} p(x) log(p(x))$, where $\sum$ denotes the sum over all the variable's possible values.

The entropy metric appears to be a logical choice for addressing the diversity issue in recommendation systems. Diversity implies avoiding deterministic recommendation patterns, where only certain contents are consistently recommended while others are consistently ignored. The entropy of a deterministic system is minimal; the probability of the recommended contents would equal 1, while the probability of the others would equal 0. In contrast, we aim for more uniform recommendations, allowing for greater diversity in the content suggested to users, thus resulting in higher entropy values.

Of course, the entropy is not the only available metric which can be used for this issue. The **Gini Index** has also been proposed in a related thesis as an alternative measure of diversity. Although it was not implemented in this study due to time constraints, it remains a promising avenue for future research.

By introducing the entropy metric as a method for mitigating content bubbles, this work lays the **groundwork** for exploring other diversity metrics (such as the Gini Index). The methodology presented here can serve as a valuable reference for researchers looking to investigate alternative metrics and their impact on recommendation diversity.

However, the following questions still remain unanswered: *On which variable do we aim to maintain high entropy? And, how much entropy is considered as "high entropy"?* These are questions we have to address if we wish to formulate our final problem effectively.

4.2 Diverse NF-RS OP Formulation

Initially, the consideration was to maintain the entropy of the **recommendations matrix** higher than a specified **threshold b**. This implies that a greater variety of contents have the opportunity to be recommended after the user views a particular content. The constraint to be incorporated was therefore expressed as:

$$H(r_{ij}) = -\sum_{j=1}^{K} r_{ij} log(r_{ij}) \geq b, \, \forall i \in \mathcal{K}$$

We will address the specific value of b later in our work.

As we mentioned before, sustaining a high entropy $H(r_{ij})$ implies that the values of elements r_{ij} are closer to 0.5 than to 0 or 1 - thus, a greater variety of content will have a probability of being recommended.

So, the new optimization problem with the entropy constraint will be :

Diverse NF-RS OP

$$\underset{\mathbf{R}}{\text{minimize}} \quad \frac{\mathbf{p}_0^T \cdot \left(\mathbf{I} - \frac{\alpha}{N} \cdot \mathbf{R}\right)^{-1} \cdot \mathbf{c}}{\frac{1}{1-\alpha}} \qquad (8.a)$$

$$\text{subject to} \quad \sum_{j=1}^{K} r_{ij} \cdot u_{ij} \geq q \cdot q_i^{max}, \quad \forall i \in \mathcal{K} \qquad (8.b)$$

$$\sum_{j=1}^{K} r_{ij} = N, \quad \forall i \in \mathcal{K} \qquad (8.c)$$

$$0 \leq r_{ij} \leq 1 (i \neq j), \quad r_{ii} = 0 \qquad (8.d)$$

$$\sum_{j=1}^{K} r_{ij} log(r_{ij}) \leq -b, \, \forall i \in \mathcal{K} \qquad (8.e)$$

The Diverse NF-RS, as introduced, continues to aim at minimizing network costs (similarly with the NF-RS). However, it additionally addresses the diversity issue by imposing a constraint to ensure that the entropy remains above a certain threshold b.

The introduced constraint (8.e) is convex because entropy is a concave function, and (8.e) involves the negative of entropy. However, the Diverse NF-RS OP is **non-convex** due to the presence of the inverse matrix in the objective function.

4.3 Convexifying Diverse NF-RS OP

To address the non-convexity of the Diverse NF-RS OP, we employ a similar procedure as previously done for the NF-RS OP (as outlined in [6]).

Specifically, we introduce the K auxiliary variables $\mathbf{z^T} = (1-\alpha)\cdot\mathbf{p_0^T}\cdot(\mathbf{I}-\frac{a}{N}\cdot\mathbf{R})^{-1}$ and then, also perform an additional variable transformation $f_{ij} = r_{ij}\cdot z_i$.

So, the constraint $(8.e): \sum_{j=1}^{K} r_{ij}log(r_{ij}) \leq -b,\ \forall i \in \mathcal{K}$ is now written as:

$$\sum_{j=1}^{K} \frac{f_{ij}}{z_i} log(\frac{f_{ij}}{z_i}) \leq -b, \Longleftrightarrow \sum_{j=1}^{K} f_{ij}\cdot log(\frac{f_{ij}}{z_i}) + b\cdot z_i \leq 0$$

Let's examine whether this new diversity constraint is indeed convex on $\mathbf{z}, \mathbf{F}$.

4.4 Convex Diverse NF-RS OP

We define : $\phi_i(\mathbf{F}, z_i) = \Big(\sum_{j=1}^{K} f_{ij}\cdot log(\frac{f_{ij}}{z_i})\Big) + b\cdot z_i = \Big(\sum_{j=1}^{K} f_{ij}\cdot(logf_{ij} - logz_i)\Big) + b\cdot z_i$

If every term of the sum is convex, then the sum is convex. So, we remove the linear part, and we only need to prove that $k_i(f_{ij}, z_i) = f_{ij}\cdot log(\frac{f_{ij}}{z_i})$ is convex for every $i, j \in \{1, ..., K\}$.

For this, we will use the **Second-Order Condition of Convexity**, i.e. we will prove that the Hessian matrix of k_i is positive semi-definite at every $f_{ij}, z_i \in domk_i$, where $domk_i$ is an open, convex set.

Theorem 1: ***Second-Order Condition of Convexity***

A twice continuously differentiable f with an open convex domain $domf$ is convex if and only if the following condition holds : $\nabla^2 f(\mathbf{x}) \succcurlyeq \mathbf{0}$ (**positive semi-definite Hessian**) at every $\mathbf{x} \in domf$.

We first calculate the Hessian of k_i:

- $\nabla k_i(f_{ij}, z_i) = \begin{bmatrix} \frac{dk_i(f_{ij}, z_i)}{df_{ij}} \\ \\ \frac{dk_i(f_{ij}, z_i)}{dz_i} \end{bmatrix} = \begin{bmatrix} \frac{d(f_{ij}\cdot(logf_{ij} - logz_i))}{df_{ij}} \\ \\ \frac{d(f_{ij}\cdot(logf_{ij} - logz_i))}{dz_i} \end{bmatrix} = \begin{bmatrix} logf_{ij} + 1 - logz_i \\ \\ -\frac{f_{ij}}{z_i} \end{bmatrix}$

- $D^2k_i(f_{ij}, z_i) = \begin{bmatrix} \frac{d^2k_i(f_{ij}, z_i)}{df_{ij}^2} & \frac{d^2k_i(f_{ij}, z_i)}{df_{ij}\ dz_i} \\ \frac{d^2k_i(f_{ij}, z_i)}{dz_i\ df_{ij}} & \frac{d^2k_i(f_{ij}, z_i)}{dz_i^2} \end{bmatrix} = \begin{bmatrix} \frac{d(logf_{ij} + 1 - logz_i)}{df_{ij}} & \frac{d(-f_{ij}/z_i)}{df_{ij}} \\ \frac{d(logf_{ij} + 1 - logz_i)}{dz_i} & \frac{d(-f_{ij}/z_i)}{dz_i} \end{bmatrix}$

$$= \begin{bmatrix} 1/f_{ij} & -1/z_i \\ -1/z_i & f_{ij}/z_i^2 \end{bmatrix} (\mathbf{A = D^2k_i(f_{ij}, z_i)\ is\ symmetric,\ since\ a_{ij} = a_{ji}})$$

Theorem 2: ***Sylvester's criterion***

A symmetric Hermitian matrix $\mathbf{A}$ is positive semi-definite if and only if all its leading principal minors are non-negative.

The principal minors of the symmetric 2×2 matrix $\mathbf{A}$ are :

- $D_1 = \mathbf{A}_{1,1} = 1/f_{ij} > 0$
- $D_2 = \mathbf{A}_{2,2} = f_{ij}/z_i^2 > 0$
- The determinant $det(\mathbf{A}) = \frac{1}{f_{ij}}\frac{f_{ij}}{z_i^2} - \frac{1}{z_i}\frac{1}{z_i} = \frac{1}{z_i^2} - \frac{1}{z_i^2} = 0$

Since all principal minors of $\mathbf{A}$ are non-negative, the Hessian $D^2k_i(f_{ij}, z_i)$ is positive semi-definite, according to Sylvester's criterion.

Additionally, $domk_i$ is convex, so $k_i(f_{ij}, z_i) = f_{ij} \cdot log(\frac{f_{ij}}{z_i})$ is convex $\forall i, j \in \{1, ..., K\}$.

Consequently, the diversity constraint **is convex**, as we intended to demonstrate.

The convexity of the reformulated diversity constraint enables us to formulate the Convex Diverse NF-RS OP, which will be identical to the Convex NF-RS OP but with the inclusion of the reformulated diversity constraint (which is denoted bellow as 13.g).

Convex Diverse NF-RS OP

$$\underset{\mathbf{p^{NF}},\ \mathbf{F}}{\text{minimize}} \quad \mathbf{c}^T\mathbf{p^{NF}}, \quad (13.a)$$

subject to

Original constraints

$$\sum_{j=1}^{K} f_{ij} \cdot u_{ij} - p_i^{NF} \cdot q \cdot q_i^{BS} \geq 0, \forall i \in \mathcal{K} \qquad (13.b)$$

$$\sum_{j=1}^{K} f_{ij} - N \cdot p_i^{NF} = 0, \forall i \in \mathcal{K} \qquad (13.c)$$

$$f_{ij} - p_i^{NF} \leq 0 \, \forall i, j \in \mathcal{K} \qquad (13.d)$$

$$f_{ij} \geq 0 \, (i \neq j), \; f_{ii} = 0 \qquad (13.e)$$

Constraint for transformation to convex(auxiliary variable F)

$$p_j^{NF} - \frac{\alpha}{N} \cdot \sum_{i=1}^{K} f_{ij} = p_{0j}, \forall j \in \mathcal{K} \qquad (13.f)$$

Diversity constraint

$$\sum_{j=1}^{K} f_{ij} \cdot log(\frac{f_{ij}}{p_i^{NF}}) + b \cdot p_i^{NF} \leq 0, \; \forall i \in \mathcal{K} \qquad (13.g)$$

4.5 Linearisation of Diverse NF-RS OP

In the case of the Diverse NF-RS OP, we are not so fortunate. The entropy constraint involves a logarithmic function, thus, it is **non-linear**.

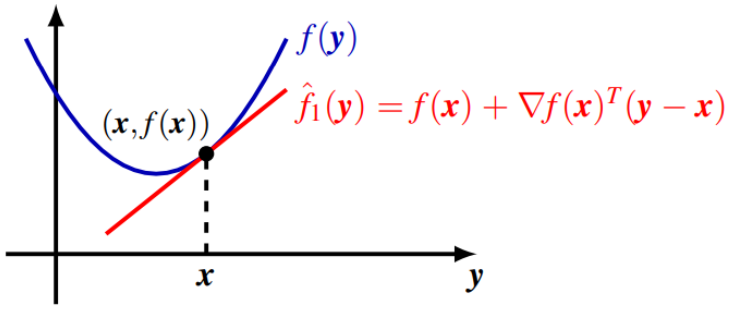
There is a large software ecosystem for non-linear, convex optimization problems; e.g. CVXPY/CVXMOD/CVXOPT. However, due to the advantages outlined above in solving a Linear Programming (LP) problem instead of a convex one, and particularly because the NF-RS has been previously solved as an LP by the authors of [6], we aim to transform our Convex Diverse NF-RS into a LP problem. This approach will enable us to utilize the existing code implementation by incorporating the additional diversity constraint.

The primary technique for transforming convex optimization problems into linear ones is **linear approximation**. This approach simplifies a convex OP by approximating the convex objective function and constraints with linear functions.

Key aspects of linear approximation in the context of convex problems include:

- **First-Order Taylor Approximation:** Linear approximation often involves using the first-order Taylor expansion to approximate a convex function. This involves replacing the function with its linear tangent at a specific point. For convex functions, this linear approximation provides a lower bound for minimization problems.

  The general expression for the first-order Taylor approximation of a (multivariate) function $f(\mathbf{x})$ around a point $\mathbf{x_0}$ is given by: $f(\mathbf{x}) \approx f(\mathbf{x}_0) + \nabla f(\mathbf{x}_0)^T(\mathbf{x} - \mathbf{x}_0)$, where $f(\mathbf{x_0})$ is the value of f at the point $\mathbf{x_0}$, $\nabla f(\mathbf{x}_0)$ is the gradient (vector of partial derivatives) of f evaluated at $\mathbf{x_0}$, and $\mathbf{x} - \mathbf{x_0}$ is the vector representing the deviation from the point $\mathbf{x_0}$.



- **Local Approximation:** Linear approximation is typically performed locally around a current iterate or solution point. Thus, the approximation is valid in the vicinity of the current point.

- **Iterative Refinement:** Linear approximation is often used iteratively. At each iteration, the linear approximation is updated based on the current solution estimate, and the process is repeated until convergence to an optimal solution.

On the next subsection, we will perform a linear approximation (using the First-Order Taylor Approximation mentioned above) of the Diverse NF-RS OP. Then, we will solve the linear problem instead of the Convex Diverse NF-RS OP.

4.5.1 Linear Approximation via Taylor series

We introduce an auxiliary set of K^2 variables $\mathbf{d} \in \mathbb{R}^{K \times K}$, and we demand the following $K^2 + K$ inequalities, which are equivalent to the original diversity constraint (13.g).

$$\sum_{j=1}^{K} d_{ij} \leq -b, \, \forall i \in \mathcal{K}$$
$$r_{ij} \cdot \log(r_{ij}) \leq d_{ij}, \, \forall i, j \in \mathcal{K}$$

The first K inequalities are linear on d_{ij}. The rest K^2 inequalities are **non-linear** on r_{ij}.

To transform them to linear constraints, we approximate the non-linear terms $r_{ij} \cdot \log(r_{ij})$ with a general family of linear cuts. Specifically, we define M lines for every pair of $\{i, j\}$, as $L(r_{ij}) = a_{m,ij} \cdot r_{ij} + b_{m,ij}$. These lines are tangent to the function $g(r_{ij}) = r_{ij} \cdot \log(r_{ij})$ in the interval $r_{ij} \in (0, 1]$, which is of our interest.

There are many ways to choose at which points we will sample $g(r_{ij})$. In these points, the tangent lines $L(r_{ij})$ will be calculated, via **Taylor series approximation**.

We will implement two ways of sampling $g(r_{ij})$ and keep the best one:

- EXPONENTIAL SAMPLING : Sample g at the points $\left\{e^{-(m-1)s}, -(m-1)se^{-(m-1)s})\right\}$
- LINEAR SAMPLING : Sample g at the points $\left\{\frac{m}{100}, \frac{m}{100}\log(\frac{m}{100})\right\}$

A. Exponential sampling

We first calculate the 1st-order Taylor approximation on points $\left\{e^{-(m-1)s}, g(e^{-(m-1)s})\right\}$:

$$L(r_{ij}) = g'\left(e^{-(m-1)s}\right)\left(r_{ij} - e^{-(m-1)s}\right) + g(e^{-(m-1)s}) =$$
$$= (1 + \log(e^{-(m-1)s}))(r_{ij} - e^{-(m-1)s}) + e^{-(m-1)s}\log e^{-(m-1)s} =$$
$$= (1 - (m-1)s)r_{ij} - e^{-(m-1)s}$$

These lines $L(r_{ij})$ are tangent to the actual $r_{ij} \cdot log(r_{ij})$ function for every pair of {i,j}. So, instead of using the K^2 non-linear inequalities $r_{ij} \cdot log(r_{ij}) \leq d_{ij}$, we use the following $M \cdot K^2$ inequalities that are linear on the variables r_{ij} and d_{ij} :

$$(1-(m-1)s)r_{ij} - e^{-(m-1)s} \leq d_{ij}, \quad \forall i,j \in \mathcal{K},\ m = 1, ..., M$$

Thus, the linear equivalent of the entropy constraint will be:

$$\sum_{j=1}^{K} d_{ij} \leq -b,\ \forall i \in \mathcal{K}$$
$$(1-(m-1)s)r_{ij} - e^{-(m-1)s} \leq d_{ij},\ \forall i,j \in \mathcal{K},\ m = 1, ..., M$$

And it transforms as follows, after applying the transformation $r_{ij} = f_{ij}/p_i^{NF}$:

$$\sum_{j=1}^{K} d_{ij} \leq -b,\ \forall i \in \mathcal{K}$$
$$(1-(m-1)s)f_{ij} - e^{-(m-1)s}\, p_i^{NF} \leq d_{ij}\ p_i^{NF},\ \forall i,j \in \mathcal{K},\ m = 1, ..., M$$

The last K^2 constraints are linear on f_{ij}, but **quadratic** on the variables p_i^{NF}, d_{ij}.

We will address this problem later. First, let's see the linear sampling of $g(r_{ij})$.

B. Linear sampling

We first calculate the 1st-order Taylor approximation on points $\left\{\frac{m}{100}, g\left(\frac{m}{100}\right)\right\}$:

$$L(r_{ij}) = g'\left(\frac{m}{100}\right)\left(r_{ij} - \frac{m}{100}\right) + g\left(\frac{m}{100}\right) = \left(\left(1 + \log\left(\frac{m}{100}\right)\right)\left(r_{ij} - \frac{m}{100}\right) + \frac{m}{100}\log\left(\frac{m}{100}\right) =$$

$$= \left(1 + \log\left(\frac{m}{100}\right)\right) r_{ij} - \frac{m}{100}$$

These lines $L(r_{ij})$ are tangent to the actual $r_{ij} \cdot log(r_{ij})$ function for every pair of {i,j}. So, instead of using the K^2 non-linear inequalities $r_{ij} \cdot log(r_{ij}) \leq d_{ij}$, we use the following $M \cdot K^2$ inequalities that are linear on the variables r_{ij} and d_{ij} :

$$\left(1 + \log\left(\frac{m}{100}\right)\right) r_{ij} - \frac{m}{100} \leq d_{ij}, \quad \forall i, j \in \mathcal{K},\ m = 1, ..., M$$

Thus, the linear equivalent of the entropy constraint is:

$$\sum_{j=1}^{K} d_{ij} \leq -b,\ \forall i \in \mathcal{K}$$
$$\left(1 + \log\left(\frac{m}{100}\right)\right) r_{ij} - \frac{m}{100} \leq d_{ij},\ \forall i, j \in \mathcal{K},\ m = 1, ..., M$$

And it transforms as follows, after applying the transformation $r_{ij} = f_{ij}/p_i^{NF}$:

$$\sum_{j=1}^{K} d_{ij} \leq -b,\ \forall i \in \mathcal{K}$$
$$\left(1 + \log\left(\frac{m}{100}\right)\right) f_{ij} - \frac{m}{100}\, p_i^{NF} \leq d_{ij}\ p_i^{NF},\ \forall i, j \in \mathcal{K},\ m = 1, ..., M$$

The last K^2 constraints are linear on f_{ij}, but they are **quadratic** on p_i^{NF}, d_{ij}.

This is an issue that must be addressed; otherwise, the previous approximation would appear futile in making the convex problem linear. The initial approach was to explore potential modifications to the problem formulation. While there may be other methods to tackle the issue, this approach proved effective for us; thus, we present it bellow.

4.5.2 Problem redefinition for achieving linearity

The problem redefinition we propose is the following:

If the new constraint referred to the vector $\mathbf{p^{NF}}$ (content demand) instead of the recommendations matrix $\mathbf{R^{NF}}$, the linearization would have functioned correctly. *However, does the meaning remain consistent?*

Ensuring that the entropy of $\mathbf{p^{NF}}$ stays high implies that more contents will have a probability of being demanded. In other words, the user will demand a more diverse set of

contents overall, and the RS will have to provide him with such diverse contents. This is because the user's demands shape recommendations.

Therefore, we posit that maintaining diversity in content demand holds the same significance as ensuring diversity in recommended videos (i.e., ensuring many videos have a good probability of being recommended). This perspective allows us to proceed as following.

Redefined constraint : $\sum_{i=1}^{K} p_i^{NF}\, log(p_i^{NF}) \leq -b$

Repeating the linear approximation methods above for $g(p_i^{NF}) = p_i^{NF}\, log(p_i^{NF})$, we get:

- $L(p_i^{NF}) = (1-(m-1)s)p_i^{NF} - e^{-(m-1)s} \leq d_i$ for **exponentially** sampled points.
- $L(p_i^{NF}) = \left(1 + \log\left(\frac{m}{100}\right)\right) p_i^{NF} - \frac{m}{100} \leq d_i$ for **linearly** sampled points.

And the constraint becomes equivalent to the following $M \cdot K + 1$ linear ones:

A. Exponential sampling

$$\sum_{i=1}^{K} d_i \leq -b$$
$$(1-(m-1)s)p_i^{NF} - e^{-(m-1)s} \leq d_i,\ \forall i \in \mathcal{K},\ m = 1, ..., M$$

B. Linear sampling

$$\sum_{i=1}^{K} d_i \leq -b$$
$$\left(1 + \log\left(\frac{m}{100}\right)\right) p_i^{NF} - \frac{m}{100} \leq d_i,\ \forall i \in \mathcal{K},\ m = 1, ..., M$$

In our code, we successfully implemented both linear and exponential sampling methods. The linearly sampled data provides a significantly improved approximation of the curve $\mathbf{p}\log(\mathbf{p})$, accompanied by substantially lower execution time. Consequently, we have opted to proceed with the linear sampling implementation for our ongoing experiments.

Bellow, the plots for the two approximations (green and red colour for the exponential and the linear sampling respectively), along with the real graph of the $\mathbf{p}log(\mathbf{p})$ function (blue colour), are illustrated using MATLAB :

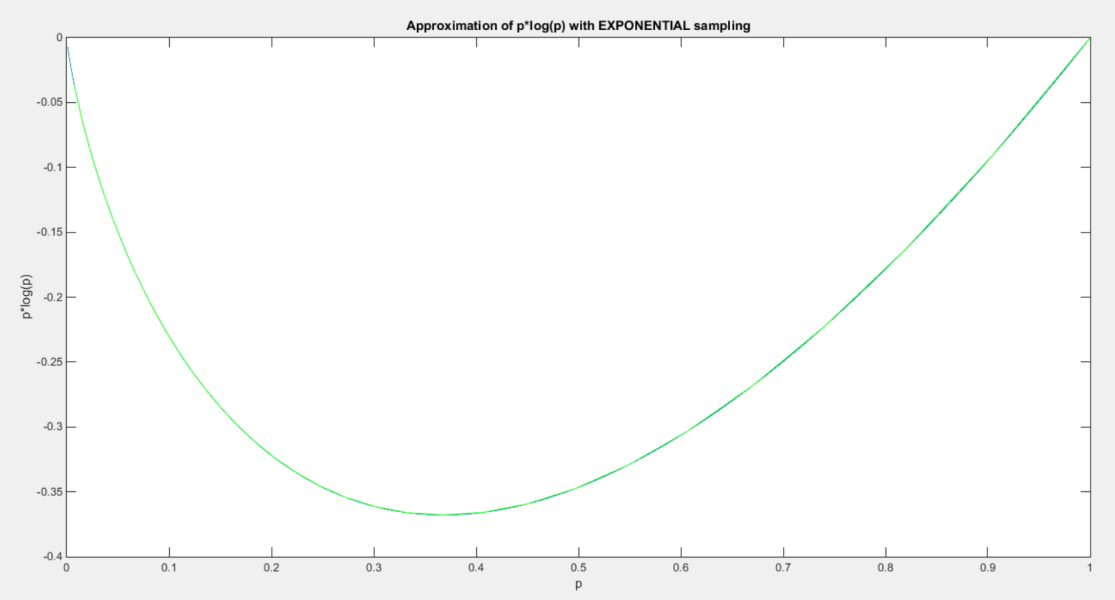


Figure 2: Function $\mathbf{p}log(\mathbf{p})$ (blue colour) and its exponentially sampled approximation (green).

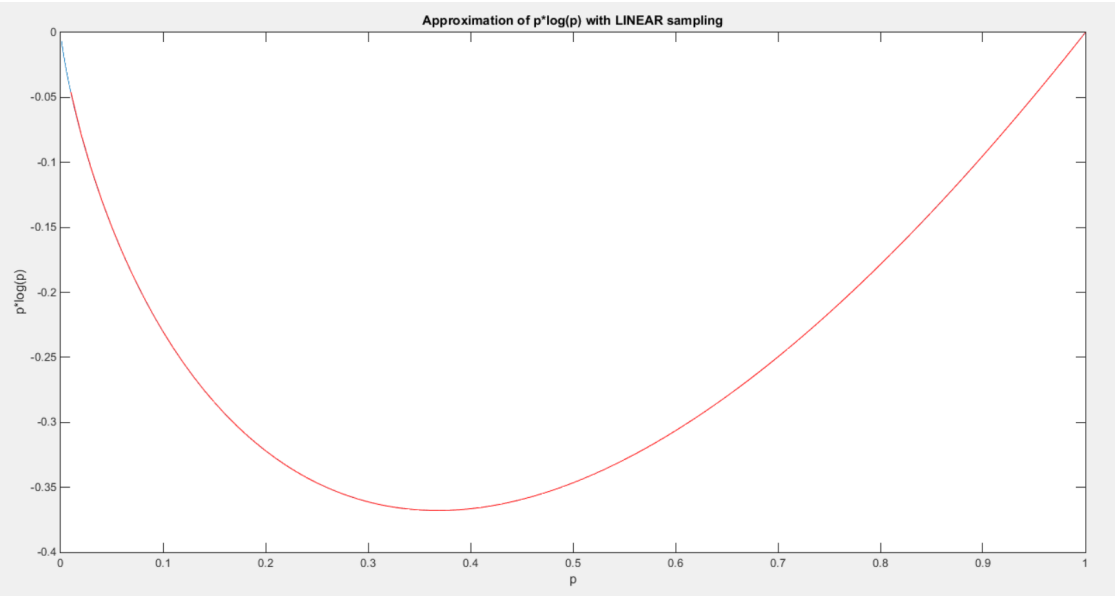


Figure 3: Function $\mathbf{p}log(\mathbf{p})$ (blue colour) and its linearly sampled approximation (red colour).

Upon closer examination, it becomes evident that the green line (exponentially sampled data) deviates more from the curve $\mathbf{p}\log(\mathbf{p})$ compared to the red line (linearly sampled

data). This demonstrates that linearly sampled data provide a superior approximation.

The Linear (Fair) Diverse NF-RS OP is presented bellow. The linear fairness constraints $\mathbf{S}(\mathbf{z}, \mathbf{p^{NF}})$ are also included for completeness. However, it must be emphasized that their usage is optional, and not all experiments will incorporate them.

4.6 Linear Diverse NF-RS OP

$$\underset{\mathbf{z},\mathbf{d},\mathbf{p^{NF}},\ \mathbf{F}}{\text{minimize}} \quad \mathbf{c}^T\mathbf{p^{NF}}, \quad (13.a)$$

subject to

Original constraints

$$\sum_{j=1}^{K} f_{ij} \cdot u_{ij} - p_i^{NF} \cdot q \cdot q_i^{BS} \geq 0, \forall i \in \mathcal{K} \qquad (13.b)$$

$$\sum_{j=1}^{K} f_{ij} - N \cdot p_i^{NF} = 0, \forall i \in \mathcal{K} \qquad (13.c)$$

$$f_{ij} - p_i^{NF} \leq 0 \, \forall i, j \in \mathcal{K} \qquad (13.d)$$

$$f_{ij} \geq 0 \, (i \neq j), \; f_{ii} = 0 \qquad (13.e)$$

Constraint for transformation to convex(auxiliary variable F)

$$p_j^{NF} - \frac{\alpha}{N} \cdot \sum_{i=1}^{K} f_{ij} = p_{0j}, \forall j \in \mathcal{K} \qquad (13.f)$$

Entropy constraint linearised (optional)

$$\sum_{i=1}^{K} d_i \leq -b, \quad (\mathbf{b} > \mathbf{0})$$

$$\left(1 + \log\left(\frac{m}{100}\right)\right) p_i^{NF} - \frac{m}{100} \leq d_i, \; \forall i \in \mathcal{K}, \; m = 1, ..., M \qquad (13.g)$$

Linear constraints for fairness (optional)

$\mathbf{S}(\mathbf{z}, \mathbf{p^{NF}})$

4.7 Quantifying Desired Diversity

However, the problem is not complete yet. Earlier in our work, we posed the question:

How much entropy is considered "high entropy"?

At that time, we opted to defer addressing this question and simply assumed that we aimed to sustain entropy higher than a specified **threshold b**. Now, we will investigate the meaningful significance of the b value for our problem.

We initiated our research with the Baseline RS, characterized by a cost ($cost^{BS}$) and an entropy value (H^{BS}). Then, we introduced the Network-Friendly RS, featuring a significantly lower cost ($cost^{NF} < cost^{BS}$) and a smaller entropy value ($H^{NF} < H^{BS}$), indicating reduced diversity in content recommendations. Lastly, we incorporated an additional constraint related to content diversity into the NF-RS, resulting in the Diverse NF-RS, which aims to maintain a cost similar to $cost^{NF}$ and an entropy value similar to H^{BS}.

Nevertheless, it's crucial that the entropy increases only to the extent that it also permits maintaining a low cost. In essence, the Diverse NF-RS endeavors to find a point that offers **a favorable trade-off between cost and diversity**.

To achieve this, we introduce a new variable called **"bubble metric"**. This variable is an input to the optimization problem, it ranges from 0 to 1, and it represents the percentage of H^{BS} we aim to achieve in our Diverse NF-RS..

So, the value of the previously specified **threshold b** is : $\text{b} = \text{bubble metric} \cdot H^{BS}$

Certainly, when the **"bubble metric"** is set to zero, there is no entropy constraint imposed. Conversely, when the **"bubble metric"** is set to one, we wish that the entropy of $\mathbf{p^{NF}}$ approaches H^{BS}.

Now that our Linear Diverse NF-RS OP is formulated and the value of **b** has been specified, we can solve the optimization problem, by adding the new diversity constraint in the code implementation of the Linear NF-RS OP. The process followed is detailed in the section bellow.

4.8 Code implementation

Our primary work, regarding the code implementation, involved including the constraints related to content bubble restrictions into the linear NF-RS problem.

Specifically, our contributions can be summarized as follows:

- We computed the baseline entropy $\mathbf{H^{BS}}$ for the specified input arguments.
- We introduced the new variable **"bubble metric"** - strictly defined in [0,1].
- From these two values, we calculated the value of **b** (b = bubble metric $\cdot$ H^{BS}).
- We included our auxiliary optimization variables d_{ij} to the solver.
- We incorporated the desired linear constraints (when bubble metric $\neq 0$).

After these modifications, the code can be tested for various "bubble metric" inputs to observe how the cost of the Diverse NF-RS changes relative to its entropy. Of course, these steps required study of the provided code and familiarity with the CPLEX tool used.

CPLEX Problem Formulation

Definition — ***CPLEX Optimization Studio***

IBM ILOG CPLEX Optimization Studio is an analytical decision support toolkit for rapid development and deployment of optimization models, using mathematical and constraint programming. It combines an integrated development environment (IDE) with the powerful Optimization Programming Language (OPL) and high-performance ILOG CPLEX optimizer solvers.

The CPLEX Optimizer solves linear programming problems. It provides a Python language interface, which is the one we used. Bellow we explain the **steps used to define and solve an LP in Python, the way CPLEX documentation requires**.

Step 1: Install and import CPLEX (Community Edition)
Step 2: Create an instance of a problem
Step 3: Choose whether the objective function will be minimized or maximized
Step 4: Create the decision variables, and their bounds, then add them to the problem
Step 5: Create the linear constraints and add them to the problem
Step 6: Solve the problem

In creating the decision variables and the linear constraints of the problem (steps 4, 5), a certain formula has to be used. We explain this with an example Python implementation :

Mathematical Representation of an LP - example

$$\begin{aligned} \underset{\mathbf{p}}{\text{minimize}} \quad & \mathbf{c^T p} \\ \text{subject to} \quad & \sum_{i=1}^{K} p_i = 1 \quad , \quad \mathbf{p} \geq \mathbf{0} \end{aligned}$$

The decision variable of this LP is $\mathbf{p}$. We would normally initialize it with zeros of required size ($length(\mathbf{p}) = K$), but we notice that the objective function contains $\mathbf{p}$. Thus, we initialize $\mathbf{p}$ so that it also considers the multipliers c_i. The decision variable's bounds are given by the last constraint : $\mathbf{p} \geq \mathbf{0}$. So, we define 0 as the lower bound and *inf* as the upper bound. Last, the linear constraint of the LP is : $\sum_{i=1}^{K} p_i = 1$.

So, the code implementing Steps **1 - 6**, for the specific example would be:

```
import cplex                                                          # 1
problem = cplex.Cplex()                                               # 2
problem.objective.set_sense(problem.objective.sense.minimize)         # 3

objective = [c_i for x in range(K)] # c^T p
lower_bounds = [0.0] * K
upper_bounds = [1.0 * cplex.infinity] * K
problem.variables.add(obj=obj,lb=lower,ub=upper)                      # 4

constraint = []                    # sum p_i = 1
l1 = list(range(K))                # indexes of p vector (p1, p2,..,pK)
l2 = [1.0 for i in range(K)]       # coefficients of p vector
tmp = [l1,l2]
constraint.append(tmp)
senses = ['E']*1                   # 'E' denotes Equality
rhs = [1.0]*1                      # the constant part

problem.linear_constraints.add(lin_expr=constr,senses=s,rhs=r) # 5
problem.solve()                                                       # 6
```

Senses must be either a list of single-character strings or a string containing the senses of the linear constraints. Each entry must be one of 'G', 'L', 'E', and 'R', indicating greater-than, less-than, equality, and ranged constraints, respectively. **Rhs** is a list of floats, specifying the right-hand side of each linear constraint.

We applied the formula detailed above to incorporate our specific bubble constraints into the code for the Linear (Fair) NF-RS OP.

Finally, we performed **tests with different values of the input arguments** to collect sufficient data for drawing conclusions. We also developed a MATLAB code to utilize the numerical results from these tests, creating **plots** that illustrate the relationship between the system's cost and entropy. The numerical result matrices and plots for each experiment are presented in the following section.

5 Results

In this section, we first describe the details of the **datasets** we used, which provide the crucial relevance matrix **U**. Then, we offer an overview of the specific values chosen for each **input argument** and the rationale behind those choices. Following this, we present the results for these arguments in many different scenarios, also proving the existence of content bubbles in NF-RS. Last, we compare the BS-RS, the NF-RS and our Diverse NF-RS, sometimes activating other fairness notions as well, as proposed in the work of [36].

5.1 Datasets and Input Arguments

In order to evaluate the baseline and the proposed algorithms, we will use two public datasets that have also been used in related work on network-friendly recommendation systems [6],[36]. We'll explain here how each dataset needs to be preprocessed, in order to construct the utility matrix **U**, that is the key input to the various algorithms.

MovieLens RS
MovieLens is a web-based recommender system and virtual community that recommends movies for its users to watch. It contains about 11 million ratings for about 8500 movies. The site uses item-based and user-based **collaborative filtering**. In addition, to address the cold-start problem for new users, MovieLens asks new users to rate how much they enjoy watching various groups of movies (e.g. movies with dark humour versus romantic comedies). The preferences recorded by this survey allow the system to make initial recommendations, even before the user has rated a large number of movies on the website. For each user, MovieLens predicts how the user will rate any given movie on the website. Based on these predicted ratings, the system recommends movies that the user is likely to rate highly. The website suggests that users rate as many fully watched films as possible, so that the recommendations given will be more accurate, since the system would then have a better sample of the user's film tastes.

Last.fm RS
Last.fm is a web-based recommender system and virtual community that recommends songs for its users to listen to. Last.fm automatically generates a profile page for every user which includes basic information such as their user name, avatar, date of registration and the total number of tracks played. Profile pages are visible to all, together with a list of top artists and tracks, and the 10 most recently played tracks. Each user's profile has a "Taste-o-Meter" which gives a rating of how compatible the user's music taste is. Last.fm features a personal recommendations page that is only visible to the user concerned and lists suggested new music and events, all tailored to the user's own preferences. Last.fm will play tracks that do not appear in the user's library, but are often played by other users with similar musical tastes. As this approach leverages the behavior of users, it is an example of a **collaborative filtering** technique [22].

The specific data we extract from these platforms includes the similarity (relevance) scores obtained from collaborative filtering for all catalogue items, i.e. the elements of **U** matrix. These relevance scores were obtained from the datasets of Last.fm and MovieLens platforms, as precisely defined in [6]. Specifically, the **'getSimilar' method** was used to derive the relevance scores of contents from the Last.fm database. In the MovieLens data, item-to-item **collaborative filtering** was applied to fill in missing user ratings and, then, **cosine distance** was employed to calculate the relevance between each pair of contents.

The **U** matrix defines the **catalog size** K, representing the number of available contents on the specific platform. However, it's important to note that for our given platforms, not all contents were utilized. For instance, in the case of MovieLens, we did not include all 8500 movies; instead, we considered only 1060 of them. Similarly, for the Last.fm platform, we considered 757 items of the platform's available contents. So, the catalogue size for our implementation is either $K = 1060$ for MovieLens, or $K = 757$ for Last.fm.

There are more datasets that can (and have been) used in similar works. However, without loss of generality, we will focus on these two datasets in this evaluation.

Input Arguments

a : The probability of following recommendations, denoted as α, is typically defined in the range $(0, 1)$. We choose a value of **0.8** - which is also what is used in Netflix- as it strikes a balance between being not too strict, but substantial enough to have a significant impact on the results. Then, we increase the value of α to **0.99** to illustrate an extreme scenario where the user almost always follows recommendations - as in Auto-play.

popularity : The **p0** vector represents the distribution of the initial content demand for all contents, i.e. the initial number of requests for a content out of the total requests. To define this vector, we need the value of K (catalogue size) and a popularity value, which determines **how content demand is initially distributed**. Specifically, the popularity value serves as the Zipf distribution parameter. In our implementation, we consider two extreme cases: a popularity of **0** (resulting in a uniform distribution) and a popularity of **1**.

L : The value of L represents the user's session length, i.e. how many items they will view in one session, on average. To observe the creation of a content bubble, a relatively large value of L is chosen (as no bubble can be created for a very small L). We set our session length to be **40** items for all the experiments we implemented.

N : The value of N represents the number of items recommended to the user. We experiment with different values of N, starting with $\mathbf{N = 2}$, and then considering a larger value of 10 items. We believe that for a catalogue size close to 1000 items, $\mathbf{N = 10}$ is a sufficiently

large number of recommended contents, and we do not increase it further. With a larger N, the user would be suggested a substantial portion of the content catalog, especially in the case of a large viewing session (e.g. for L = 40, as chosen). Such a scenario would deviate from a realistic representation.

C : The value of C represents the number of items cached in the vicinity of the user. These items have zero cost for the network - while the rest have a content cost equal to one - and are encouraged to be recommended in the case of NF-RS. Initially, we set the cache size to **20**, and later decrease it to **5**. A cache size larger than 20 would not be realistic for a catalog of only approximately 1000 items.

caching policy : As mentioned in Section 3, the caching policy for the Baseline RS is the "top N" policy, implying that the N most relative contents will be recommended. Consequently, the recommendation matrix $\mathbf{R^{BS}}$ will contain 1's at positions corresponding to the highest u_{ij} values, and the content demand vector $\mathbf{p^{BS}}$ will represent the distribution for this top N policy. The C items with the largest $\mathbf{p^{BS}}$ values will be selected to be cached. We adhere to this approach in all the presented results.

q : This parameter determines the percentage of q_{max} (Baseline quality) we aim to offer. We focus on values of q greater than 0.5 (i.e. $> 50\%$), as we wish to maintain a substantial level of recommendation quality while achieving our diversity goal. Hence, we initially explore q values set to **0.8**, followed by an examination of extreme cases where q is either **0.5** or **0.99**. This approach provides a comprehensive understanding of how the q parameter influences the problem solution.

bubble metric : The bubble metric defines the amount of BS-RS diversity we aim to attain. We first conduct all experiments with a bubble metric set to **0**, representing the standard NF-RS implementation. Subsequently, we run various scenarios with different bubble metric values ($\leq$ **1**) to generate the entropy-cost curve (trade-off). The bubble metric will be referred to as 'bubble' in the matrices presented below.

Fairness mode : This is an optional parameter in our problem, indicating whether another fairness metric is included as an additional constraint in the optimization problem (that is when Fairness mode is defined). The three available options, when specified, are **KL**, **TV**, and **MAX** fairness metric.

Fairness weight : This parameter is also optional and should only be defined when the fairness mode is specified. It represents the weight of the fairness constraint, denoted as the constant value c_f in the optimization problem. We explore various values of this parameter in each scenario, to comprehend how each additional fairness metric impacts diversity: 1) without our entropy constraint and 2) in conjunction with the entropy constraint.

On the following subsections, we present the results of the experiments we conducted.

5.2 Diversity in BS-RS Vs in NF-RS

Below are a set of results, obtained from the Last.Fm dataset, which provide **evidence** of decreased diversity in NF-RS. Although the specific results are detailed in subsequent subsections, we present numerical values here (**Table 4**), accompanied by a scatter plot (**Figure 4**) for better visualization, to demonstrate the existence of the content bubble problem discussed in previous sections. The calculations include network costs and entropy values for both the BS-RS and the NF-RS.

The results in **Table 4** are presented in **pairs**. Each pair includes data for the BS-RS and the NF-RS of the same specific input arguments (i.e. the ones described in Section **5.1**), while a double line is used to separate the distinct pairs. The first column indicates whether the results are for BS-RS or NF-RS, while the second column displays the network cost for the respective RS. The third column expresses this cost as a percentage of the BS cost. Subsequently, the entropy of the RS is computed in the fourth column, followed by the percentage of this entropy relative to the BS entropy in the fifth column. This percentage for the NF-RS illustrates the decrease in diversity compared to the BS-RS. The values of cost and entropy for each RS have also been plotted in **Figure 4**. A more detailed exploration of this effect is available in another thesis authored by one of our colleagues.

Examples of reduced diversity (entropy) in NF-RS compared to BS-RS:

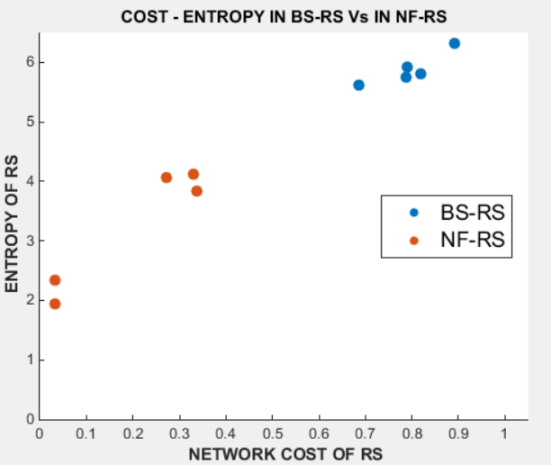


Figure 4: Cost-Entropy scatter plot.

RS	cost	% of $cost^{BS}$	H^{RS}	% of H^{BS}
BS	0.817	100%	5.808	100%
NF	0.032	3.9%	1.938	**33%**
BS	0.687	100%	5.628	100%
NF	0.337	49%	3.829	**67%**
BS	0.891	100%	6.312	100%
NF	0.271	30%	4.070	**64%**
BS	0.790	100%	5.917	100%
NF	0.329	41%	4.130	**69%**
BS	0.787	100%	5.745	100%
NF	0.033	4%	2.349	**40%**

Table 4: Cost-Entropy numerical values.

We observe scenarios where the NF-RS exhibits a **moderate reduction** in diversity, such as when 69% of H^{BS} is achieved. This scenario corresponds to a large value of N (number of recommendations), which poses challenges in decreasing costs. The cost is big either way, thereby allowing for a wider choice of recommendations (close to that of BS). Conversely, there are cases where the system's diversity is **considerably decreased**, such as when 33% of H^{BS} is achieved. This scenario corresponds to a large value of α (user always following recommendations), which allows for a great reduction in costs and

thus, a restricted choice of recommendations. These preliminary findings indicate that *while enforcing network-friendliness does not consistently lead to a significant decrease in diversity, it is* **essential to impose our entropy constraint** *to ensure a consistent level of diversity in recommendations.*

In the next subsection, we examine the results obtained from our Diverse NF-RS also incorporating the entropy constraint. The results have the same format as in **Table 4**.

5.3 Diverse NF-RS

5.3.1 Lastfm pop0 a0.8 N2 C20 CPtop Q0.8 L40 No Fairness

In our initial test, we use the **Last.Fm** dataset and consider a scenario with moderately defined input parameters aiming for realism. Specifically, (i) **popularity** is set to 0 - a uniform initial demand for all contents; (ii) **a** is set to 0.8 - moderate strictness level of following recommendations; (iii) **N** is set to 2 - a realistic choice given the dataset's size of 757 items: e.g. YouTube has a library of over 800 million contents and suggests approximately 200 contents per recommendation instance; (iv) **C** is set to 20 - relatively large cache capacity considering only 2 recommendations are provided; (v) **q** is set to 0.8 - a commendable QoR that is neither too lenient nor overly restrictive; (vi) **L** is set to 40 - extended viewing sessions.

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BS	0.89078	100%	6.31276	100%
NF - bubble 0	0.41072	46%	4.58777	72%
NF - bubble 0.10	0.41072	46%	4.63642	72%
NF - bubble 0.70	0.41072	46%	4.63682	73%
NF - bubble 0.75	0.41072	46%	4.63750	73.5%
NF - bubble 0.80	0.42211	47%	4.92127	78%
NF - bubble 0.82	0.43425	48%	5.03492	79%
NF - bubble 0.85	0.46149	52%	5.21413	82%
NF - bubble 0.90	0.54023	60%	5.55556	88%
NF - bubble 0.95	0.64721	72%	5.91701	93%
NF - bubble 1	0.76969	86%	6.27383	99.4%

Table 5: Cost-Entropy numerical values for Lastfm pop0 a0.8 N2 C20 CPtop Q0.8 L40.

Notably, 72% of **H^{BS}** is achieved without the bubble constraint. Hence, it is reasonable to anticipate shifts in results for bubble metric values exceeding 0.7; we sample in this range.

Elevating the cost from a minimum of 46% of BS (in NF with a bubble metric of 0) to 52% of BS, results in an enhanced entropy of 82% of **H^{BS}**, surpassing the initial 72%. This represents a favorable trade-off between cost and entropy. We plot the values of **Table 5**

and observe the nature of the curve, which we anticipate to be convex. Such convexity indicates that higher diversity can be achieved with a relatively modest increase in cost.

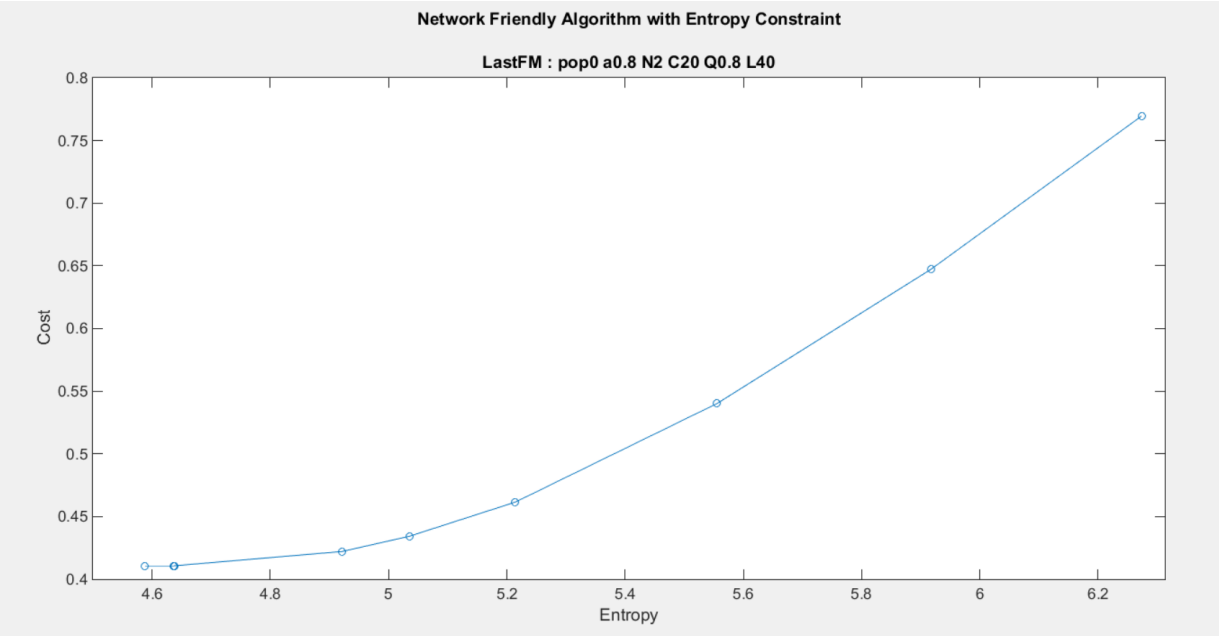


Figure 5: Cost-Entropy plot for Lastfm pop0 a0.8 N2 C20 CPtop Q0.8 L40, **Diverse NF-RS**.

Certainly, a convex curve is evident, suggesting a favorable cost-entropy trade-off and encouraging further experimentation. In the subsequent subsections, we will systematically **vary one input argument at a time** to evaluate the behavior of the Diverse NF-RS.

5.3.2 Lastfm pop0 a0.99 N2 C20 CPtop Q0.8 L40 No Fairness

First, we modify α from 0.8 to 0.99, indicating that users will almost invariably adhere to recommendations. This heightened compliance results in **minimal network costs**, primarily due to the prevalence of cached content recommendations, which users are readily accepting. Consequently, we anticipate a decrease in costs compared to the previous scenario.

However, this reduction in costs also resulted in **decreased entropy** for the NF-RS, as users primarily interact with cached contents, which restrict the variety of recommendations. Nevertheless, there is a possibility that the *combination of low costs and the initially poor NF entropy* **could pave the way for a significant cost-entropy trade-off**. Given the minimal initial network costs, there might be room for prioritizing other factors over cost, such as diversity.

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BASELINE	0.81793	100%	5.80810	100%
NF - bubble 0	0.03197	3.9%	1.93827	33%
NF - bubble 0.10	0.03197	3.9%	2.10223	36%
NF - bubble 0.40	0.03433	4.2%	2.29268	39%
NF - bubble 0.60	0.11894	14%	3.43656	59%
NF - bubble 0.70	0.20485	25%	3.98508	68%
NF - bubble 0.75	0.25016	30%	4.25031	73%
NF - bubble 0.80	0.29595	36%	4.50037	77%
NF - bubble 0.85	0.34243	41%	4.74097	81%
NF - bubble 0.90	0.40389	49%	5.04006	86%
NF - bubble 1	0.58346	71%	5.71487	98%

Table 6: Cost-Entropy numerical values for Lastfm pop0 a0.99 N2 C20 CPtop Q0.8 L40.

Strategically increasing the cost to 30% of the BS-RS (from an initial 3.9%) yields a notable improvement, reaching 73% of H^{BS} from the initial 33%. This solidifies our primary hypothesis that a low initial cost allows for significant enhancements in entropy. We can now confidently conclude that higher values of **a** lead to an **improved trade-off** between entropy and cost; the convex nature of the graph in **Figure 6** also confirms this.

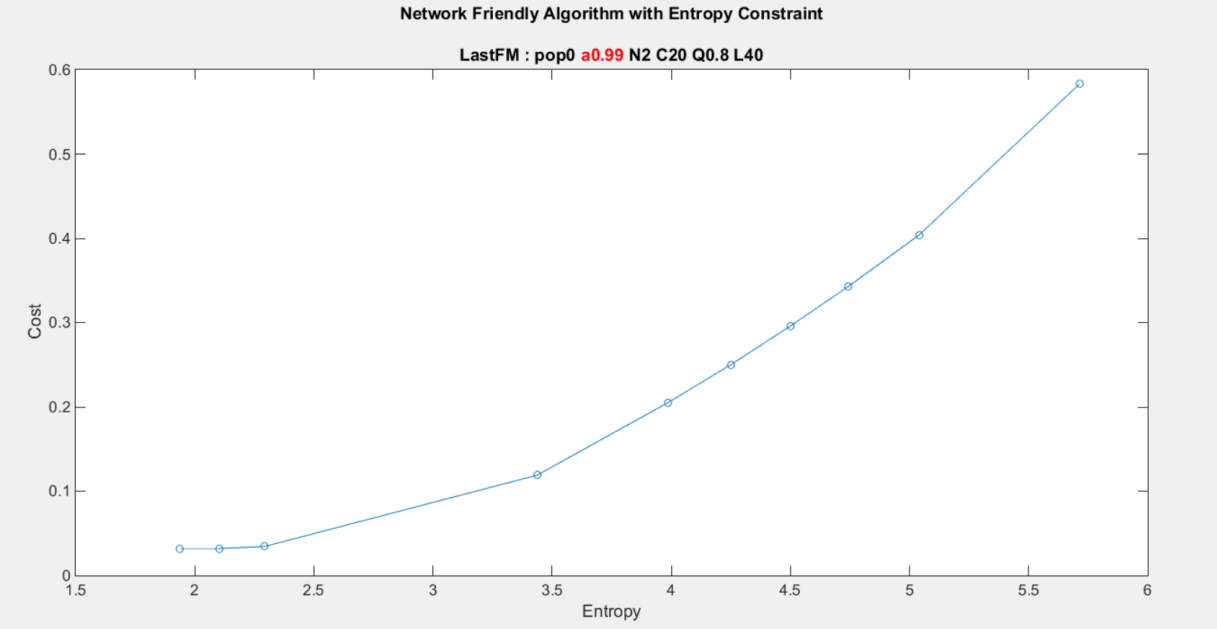


Figure 6: Cost-Entropy plot for Lastfm pop0 a0.99 N2 C20 CPtop Q0.8 L40, **Diverse NF-RS**.

5.3.3 Lastfm pop0 a0.8 N10 C20 CPtop Q0.8 L40 No Fairness

The subsequent adjustment involves altering the number of recommendations. Precisely, we will maintain the input parameters from section **5.3.1** and modify **N** from 2 to 10.

With this increase in recommended content, we anticipate a **rise in network costs**; because the cache hit ratio (i.e. network gains) of cached and related contents to recommended contents tends to be smaller for larger N values. In particular, let M represent the number of cached **and** related contents in **U**. If M is greater than the recommended number of videos (M > N), the cache hit ratio will be favorable. However, if M is less than the recommended number of videos (M < N), it necessitates recommending related, non-cached items, incurring higher costs. Unfortunately, this is the prevalent scenario, as it is quite impossible - for this catalogue and cache size - to feature a substantial number of contents that are *both* cached and related. For a specific M value, and considering the cache hit ratio (CHR) defined as CHR $= \frac{M}{N}$, increasing N diminishes the success rate and escalates the cost for a given entropy value.

This rise in network costs results in a **greater initial diversity** in NF recommendations, as more content options meet the "relaxed" network cost criteria, resembling choices closer to the BS-RS. Unfortunately, we anticipate that this heightened initial diversity may limit the scope for improvements in cost-entropy trade-offs. As a result, we expect a **less favorable trade-off** in this scenario compared to that observed in section **5.3.1**.

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BASELINE	0.73710	100%	6.01646	100%
NF - bubble 0	0.49529	67%	5.02111	83%
NF - bubble 0.40	0.49529	67%	5.02111	83%
NF - bubble 0.85	0.49529	67%	5.02111	83%
NF - bubble 0.90	0.50986	69%	5.27271	87%
NF - bubble 0.95	0.55584	75%	5.59254	92%
NF - bubble 1	0.65427	88%	5.95426	99%

Table 7: Cost-Entropy numerical values for Lastfm pop0 a0.8 N10 C20 CPtop Q0.8 L40.

We confirm our initial suspicion that increasing **N** leads to a less favorable entropy-cost trade-off. Nevertheless, it's crucial to note that the trade-off should still retain convexity. For instance, a mere 2% increment in NF cost yields a noticeable 4% increase in NF entropy (as seen in **Table 7**), which is a positive sign. The specific relationship between cost and entropy is depicted in **Figure 7** provided below.

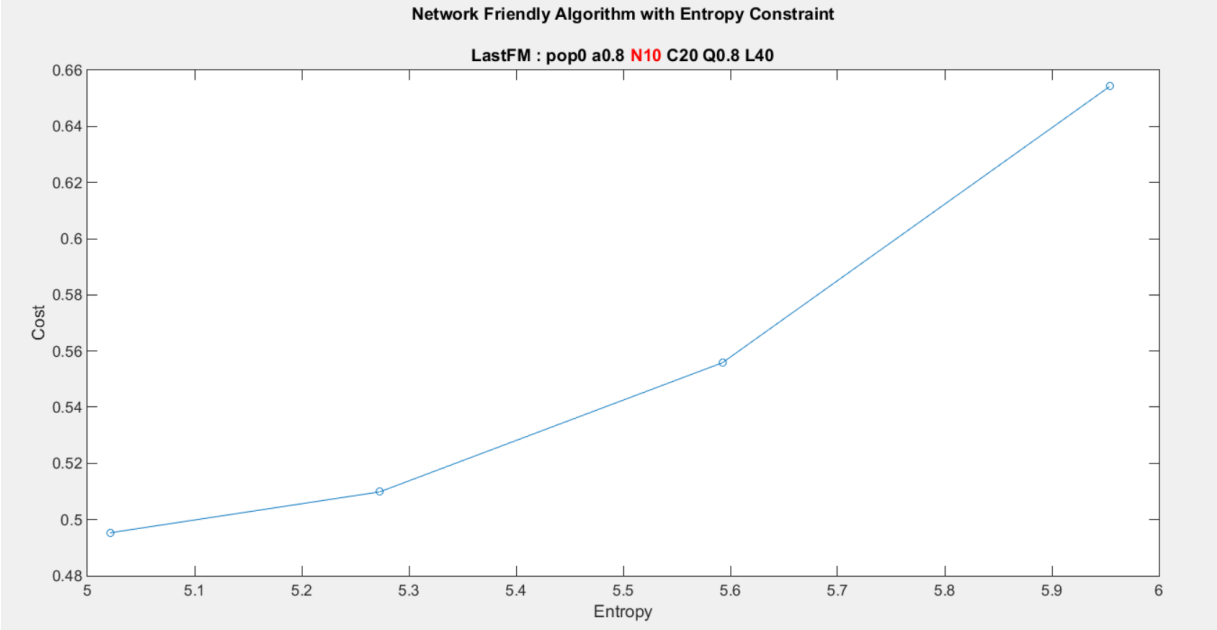


Figure 7: Cost-Entropy plot for Lastfm pop0 a0.8 N10 C20 CPtop Q0.8 L40, **Diverse NF-RS**.

5.3.4 Lastfm pop0 a0.99 N10 C20 CPtop Q0.8 L40 No Fairness

The less favorable trade-off observed in the previous subsection piqued our interest in exploring the effects of simultaneously increasing both **N** (which is detrimental to the trade-off) and **a** (which is beneficial for the trade-off). In this context, we have now adjusted the values of both of these input arguments.

ALGORITHM	COST	% of $cost^{BS}$	ENTROPY	% of H^{BS} achieved
BASELINE	0.68703	100%	5.62819	100%
NF - bubble 0	0.33720	49%	3.82980	67%
NF - bubble 0.40	0.33720	49%	3.82980	67%
NF - bubble 0.60	0.33720	49%	3.82980	67%
NF - bubble 0.70	0.33746	49%	3.90946	68%
NF - bubble 0.80	0.37034	53%	4.44870	79%
NF - bubble 0.95	0.47627	69%	5.19651	92%
NF - bubble 1	0.52875	77%	5.46831	97%

Table 8: Cost-Entropy numerical values for Lastfm pop0 a0.99 N10 C20 CPtop Q0.8 L40.

We observe that, while a higher value of parameter **a** might alleviate the substantial cost associated with a larger **N**, the compromised trade-off resulting from the larger N value **persists**. The cost-entropy trade-off is presented in **Figure 8** below.

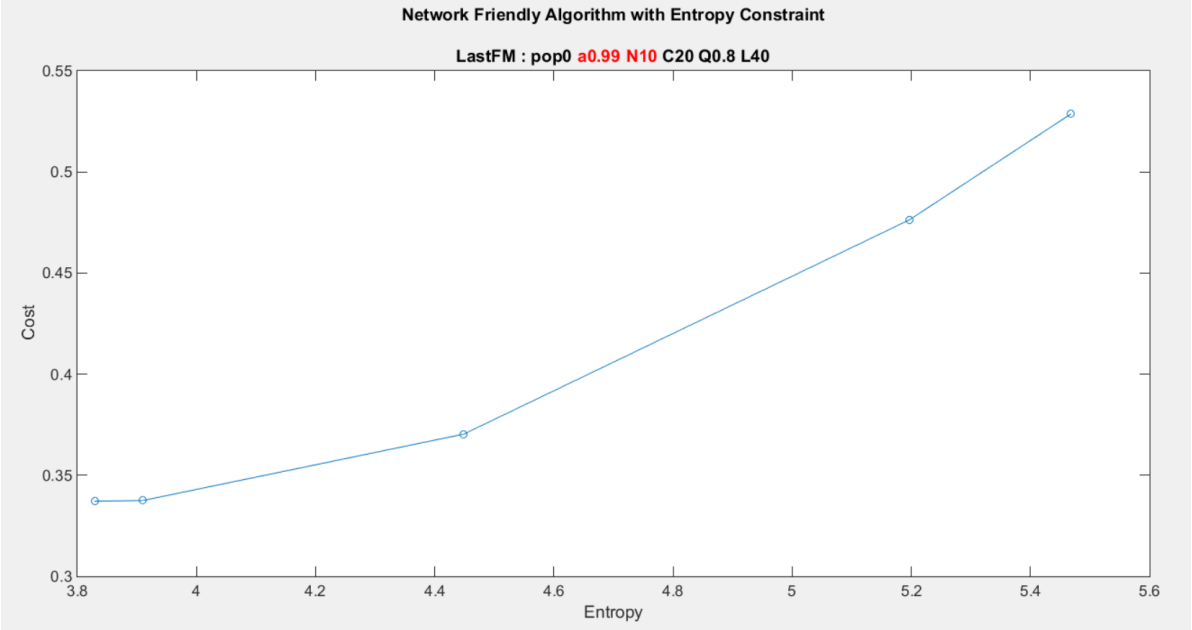


Figure 8: Cost-Entropy plot for Lastfm pop0 a0.99 N10 C20 CPtop Q0.8 L40, **Diverse NF-RS**.

5.3.5 Lastfm pop0 a0.8 N2 C5 CPtop Q0.8 L40 No Fairness

The subsequent adjustment involves altering the cache size of our problem. Decreasing the cache size would inevitably lead to **increased network costs**, as there would be a diminished probability of related content being cached. Similar to the scenario of increasing **N**, which also led to higher initial network costs, we anticipate that this modification would result in a **less favorable trade-off**.

ALGORITHM	COST	% of $cost^{BS}$	ENTROPY	% of H^{BS} achieved
BASELINE	0.96344	100%	6.31276	100%
NF - bubble 0	0.61038	63%	4.59893	72%
NF - bubble 0.70	0.61038	63%	4.59893	72%
NF - bubble 0.80	0.62151	64%	4.82147	76%
NF - bubble 0.90	0.70044	72%	5.57939	88%
NF - bubble 0.95	0.76909	79%	5.92205	94%
NF - bubble 1	0.85579	88%	6.26312	99%

Table 9: Cost-Entropy numerical values for Lastfm pop0 a0.8 N2 C5 CPtop Q0.8 L40.

Our initial suspicion is confirmed; however, it's worth noting that the trade-off still retains convexity. For instance, with a 9% increase in NF cost, there is a substantial 16% increase in NF entropy. The cost-entropy trade-off is presented in **Figure 9** below.

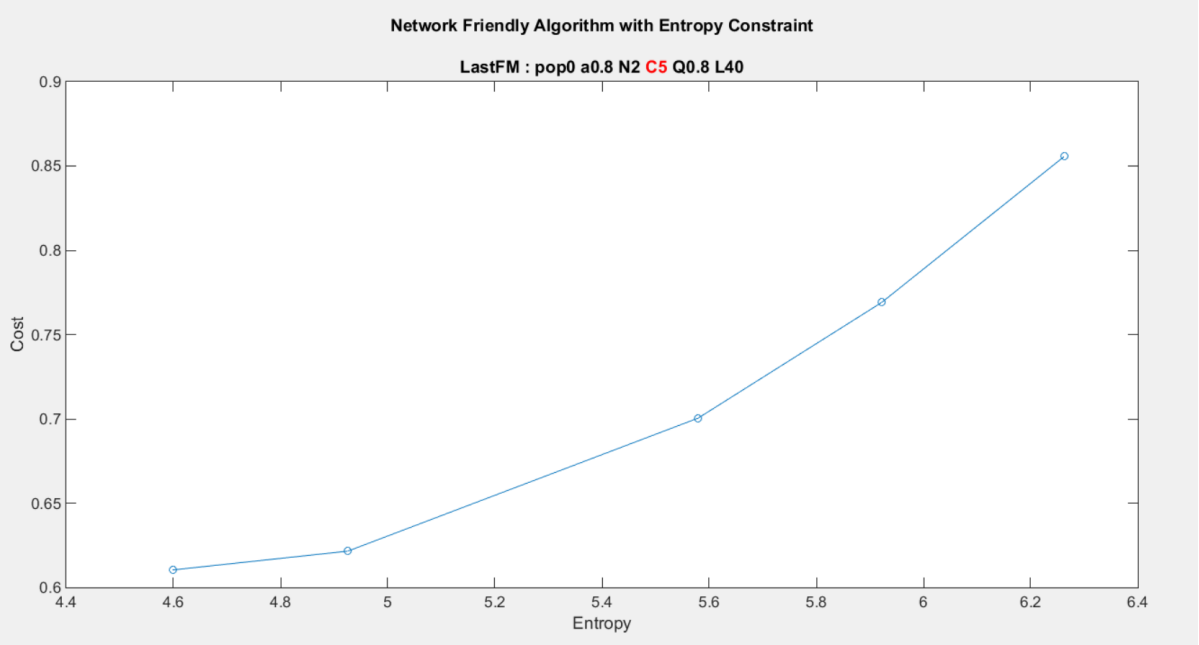


Figure 9: Cost-Entropy plot for Lastfm pop0 a0.8 N2 C5 CPtop Q0.8 L40, **Diverse NF-RS**.

5.3.6 Lastfm pop0 a0.8 N2 C20 CPtop Q0.5 L40 No Fairness

We now proceed to experiment with the value of **Q** (quality of recommendations - QoR). Decreasing **Q** implies that the quality constraint is not so tight. A higher degree of elasticity regarding the quality constraint would contribute to an **improved entropy-cost trade-off**, as it allows for the consideration of numerous alternative solutions that may offer higher entropy while maintaining a similar cost level.

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BASELINE	0.89078	100%	6.31276	100%
NF - bubble 0	0.27167	30%	4.07049	64%
NF - bubble 0.70	0.27324	30%	4.31054	68%
NF - bubble 0.75	0.28998	32%	4.57350	72%
NF - bubble 0.80	0.34348	38%	4.85347	76%
NF - bubble 0.85	0.43856	49%	5.22508	82%
NF - bubble 0.90	0.53895	60%	5.59316	88%
NF - bubble 0.95	0.64721	72%	5.94419	94%
NF - bubble 1	0.76969	86%	6.28570	99%

Table 10: Cost-Entropy numerical values for Lastfm pop0 a0.8 N2 C20 CPtop Q0.5 L40.

Remarkably, the trade-off is highly favorable. Even without a cost increment, there is a 4% increase in entropy, while with just a 2% increase in NF cost, there is a significant 8% increase in NF entropy. The cost-entropy trade-off is presented in **Figure 10** below.

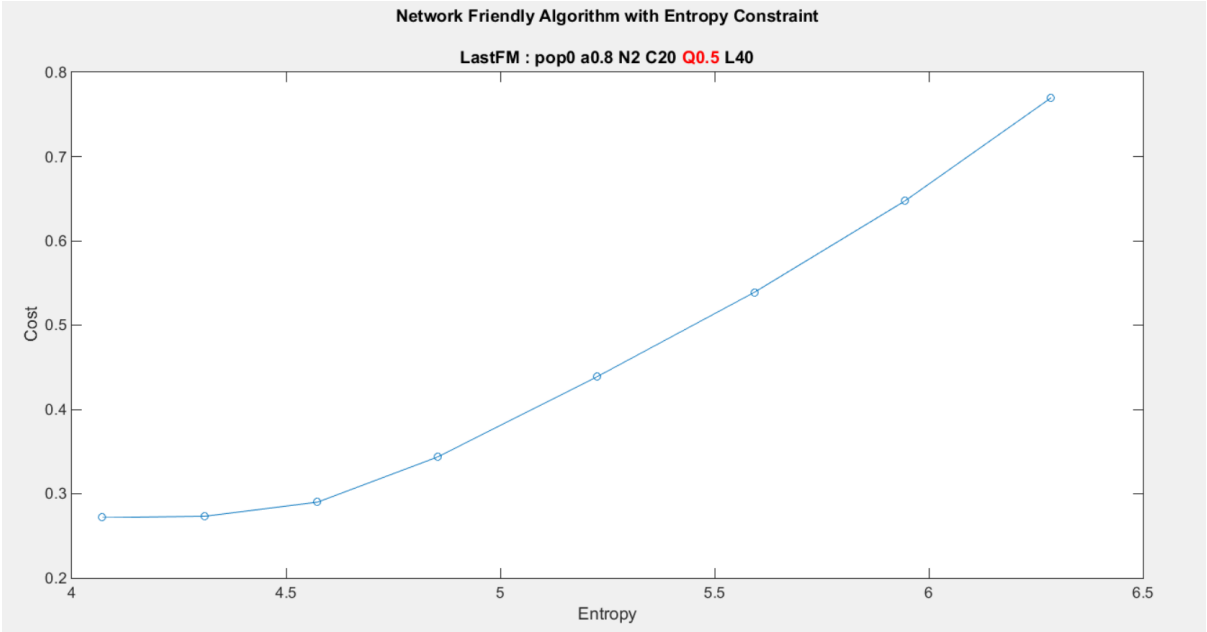


Figure 10: Cost-Entropy plot for Lastfm pop0 a0.8 N2 C20 CPtop Q0.5 L40, **Diverse NF-RS**.

5.3.7 Lastfm pop0 a0.8 N2 C20 CPtop Q0.99 L40 No Fairness

We also examine the extreme case where **Q** is 0.99, indicating a stringent requirement for excellent recommendations' quality. In this highly constrained scenario, we anticipate both **extremely high costs** and a **poor trade-off**, as there might not be much room for improvement in the trade-off while satisfying such a high QoR constraint.

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BASELINE	0.89078	100%	6.31276	100%
NF - bubble 0	0.80527	90%	6.12899	96%
NF - bubble 0.10	0.80527	90%	6.12899	96%
NF - bubble 0.60	0.80527	90%	6.12899	96%
NF - bubble 0.70	0.80527	90%	6.12911	97%
NF - bubble 0.80	0.80527	90%	6.12911	97%
NF - bubble 0.90	0.80527	90%	6.12911	97%
NF - bubble 1	0.81876	91%	6.24325	99%

Table 11: Cost-Entropy numerical values for Lastfm pop0 a0.8 N2 C20 CPtop Q0.99 L40.

Indeed, the initial NF cost is very high, paralleled by an elevated entropy. In the extreme scenario where we aim for 100% of $\mathbf{H}^{\mathbf{BS}}$, the entropy does increase to 99%, with a 1% compromise in cost.

Also, our suspicion about the unfavorable entropy-cost trade-off is confirmed. This phenomenon is attributed to the limited size of our catalog, making it highly unlikely that any alternative solutions satisfying the quality constraint of ≥ 0.99 exist with lower cost and substantial entropy.

The cost-entropy trade-off is presented in **Figure 11** below. The plot appears as a line because there are only two available entropy values in this context, resulting in a straightforward representation.

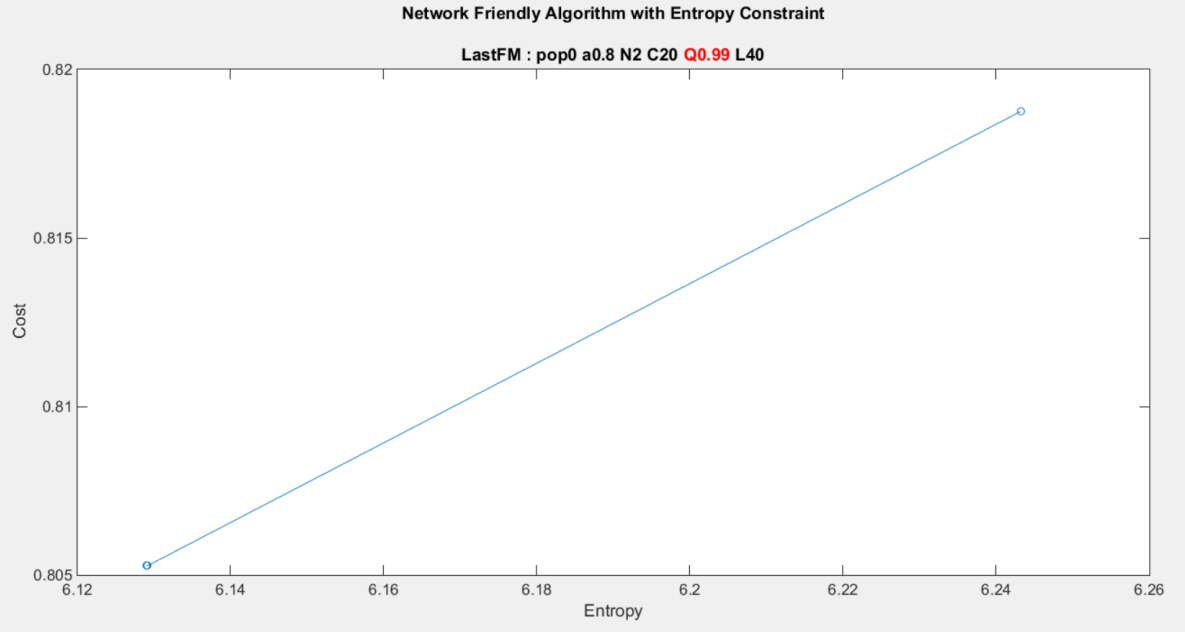


Figure 11: Cost-Entropy plot for Lastfm pop0 a0.8 N2 C20 CPtop Q0.99 L40, **Diverse NF-RS**.

5.3.8 Lastfm pop1 a0.8 N2 C20 CPtop Q0.8 L40 No Fairness

Last, we experiment with the popularity value. Transitioning from a popularity value of 0 to 1 signifies a change in the distribution of the initial content demand, denoted as **p0**.

For a popularity value of 0, **p0** adheres to a Zipf* distribution with a zipf parameter set to 0, effectively equivalent to a uniform distribution: $\mathbf{p0} = [\frac{1}{K}, \frac{1}{K}, \ldots, \frac{1}{K}]$.

In the case of a popularity value of 1, **p0** follows a Zipf* distribution with a zipf parameter of 1. The distribution formula for each item i is given by $p0_i = \frac{(i+1)^{-1}}{\sum_{j=1}^{K} j^{-1}}$. So, the values of **p0** for a Zipf parameter of 1 and a small catalog size of K = 5, will be: $\mathbf{p0} = [0.4379, 0.2189, 0.1459, 0.1094, 0.0876]$. These values of **p0** indicate that the first content has the highest probability of being chosen when the user ***enters*** the platform, with subsequent items having decreasing probabilities.

$\star$ Zipf distributions are utilized for sampling data in accordance with Zipf's law, which posits that the n-th common term is 1/n times as frequent as the most common term.

Following the selection of the initial content, the user either adheres to the system's recommendations or not. In the latter case, the user chooses ***again*** from the Zipf distribution vector **p0**. In summary, a popularity of 0 implies that whenever the user does *not* follow recommendations, they choose from the catalog with *equal* probability for each content. Conversely, a popularity of 1 signifies that the user selects from the catalog with *a descending* probability based on the content's position in the list.

In the context of selecting elements to be cached (according to a "top" policy), preference is given to those with the highest $\mathbf{p^{BS}}$ values. The posterior content demand, $\mathbf{p^{BS}}$, is influenced by **p0**, as expressed by the formula: $\mathbf{p^{BS}} = (1-\alpha)\mathbf{p0^T}(\mathbf{I} - \frac{a}{N}\mathbf{R})^{-1}$. Consequently, variations in **p0** affect the network cost due to their impact on the selection of cached content. Specifically, items with higher demand are more likely to be cached. Thus, we anticipate **lower network costs** for a popularity value of 1 but we also expect **a decrease in diversity** due to increased popularity bias. Consequently, we expect the cost-entropy trade-off to resemble that of popularity 0, perhaps slightly worse.

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BASELINE	0.79018	100%	5.91797	100%
NF - bubble 0	0.32912	41%	4.13001	69%
NF - bubble 0.70	0.32912	41%	4.15314	70%
NF - bubble 0.75	0.33467	42%	4.37356	73%
NF - bubble 0.80	0.35796	45%	4.62883	78%
NF - bubble 0.85	0.39522	50%	4.88194	82%
NF - bubble 0.90	0.44470	56%	5.15382	87%
NF - bubble 0.95	0.52940	67%	5.47965	93%
NF - bubble 1	0.62774	79%	5.82893	99%

Table 12: Cost-Entropy numerical values for Lastfm pop1 a0.8 N2 C20 CPtop Q0.8 L40.

Our intuitions are confirmed, as we can see in **Figure 12** below.

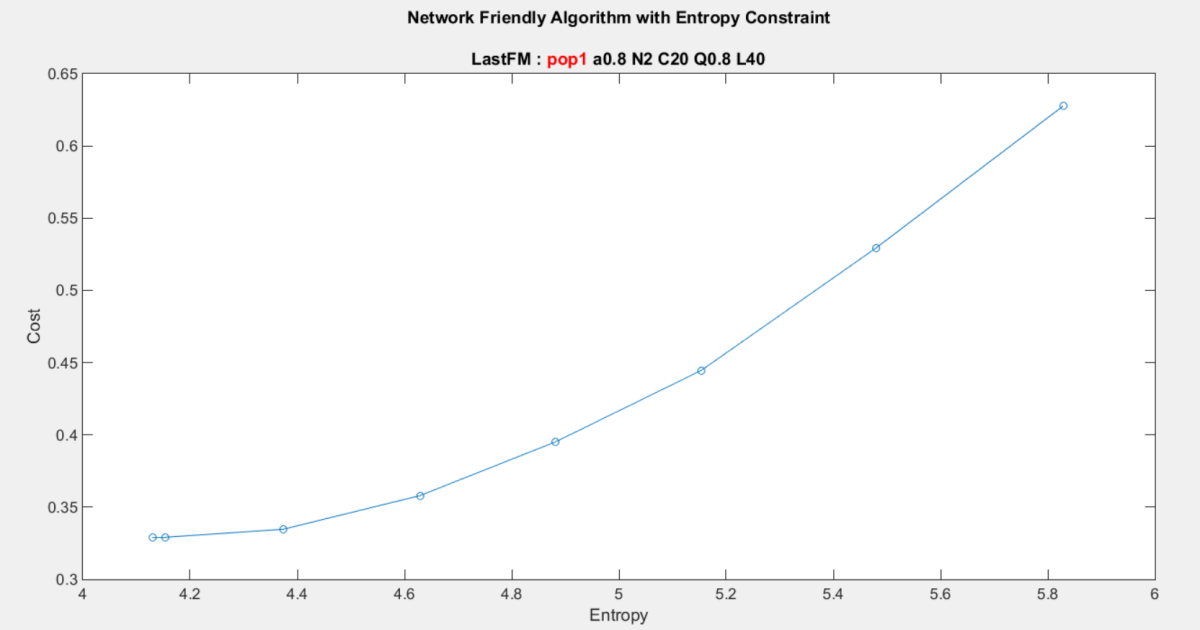


Figure 12: Cost-Entropy plot for Lastfm pop1 a0.8 N2 C20 CPtop Q0.8 L40,**Diverse NF-RS**.

5.3.9 Lastfm pop1 a0.99 N2 C20 CPtop Q0.8 L40 No Fairness

Having a user who always follows recommendations and caching these recommendations solely based on their popularity represents the scenario where we anticipate the lowest network costs compared to all the previously discussed cases. We were particularly interested in exploring this scenario.

ALGORITHM	COST	% of $cost^{BS}$	ENTROPY	% of H^{BS} achieved
BASELINE	0.78742	100%	5.74528	100%
NF - bubble 0	0.03275	4%	2.34916	40%
NF - bubble 0.10	0.03275	4%	2.61673	45%
NF - bubble 0.50	0.03355	4.2%	2.83142	49%
NF - bubble 0.55	0.05036	6%	3.11591	54%
NF - bubble 0.60	0.08533	10%	3.39780	60%
NF - bubble 0.70	0.17530	22%	3.96947	69%
NF - bubble 0.80	0.26843	34%	4.46336	78%
NF - bubble 0.90	0.38141	48%	4.98193	87%
NF - bubble 1	0.56048	71%	5.65959	99%

Table 13: Cost-Entropy numerical values for Lastfm pop1 a0.99 N2 C20 CPtop Q0.8 L40.

The cost associated with this paradigm indeed represents the lowest cost achieved among all the examples, and the cost-entropy trade-off is exceptionally favorable as well,

as depicted in **Figure 13** below.

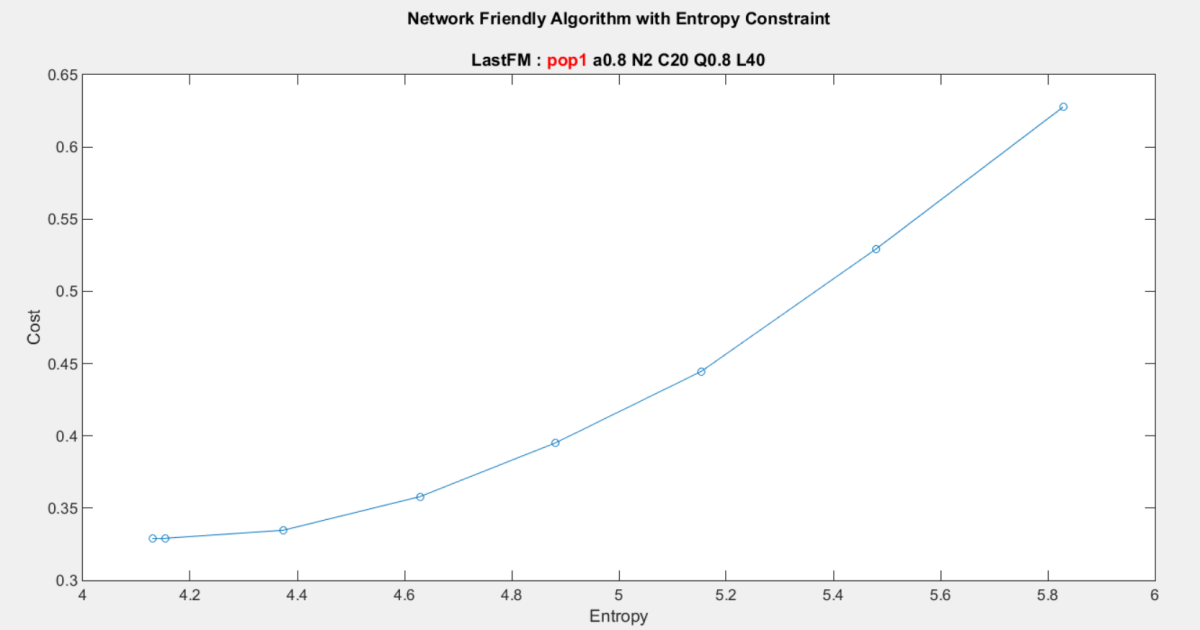


Figure 13: Cost-Entropy plot for Lastfm pop1 a0.99 N2 C20 CPtop Q0.8 L40,**Diverse NF-RS**.

However, it's crucial to acknowledge that the execution time for this particular instance is exceptionally high, and the value of $\alpha = 0.99$ is not practical or realistic.

We shall now proceed to present the results derived from the MovieLens dataset.

5.3.10 MovieLens pop0 a0.8 N2 C20 CPtop Q0.8 L40 No Fairness

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BASELINE	0.89411	100%	6.50775	100%
NF - bubble 0	0.33977	38%	4.58428	70%
NF - bubble 0.80	0.36793	41%	5.02457	77%
NF - bubble 0.85	0.45472	50%	5.44687	84%
NF - bubble 0.90	0.48819	54%	5.58523	86%
NF - bubble 0.95	0.59521	66%	5.93344	92%
NF - bubble 1	0.66438	74%	6.14127	95%

Table 14: Cost-Entropy numerical values for MovieLens pop0 a0.8 N2 C20 CPtop Q0.8 L40.

5.3.11 MovieLens pop0 a0.99 N2 C20 CPtop Q0.8 L40 No Fairness

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BASELINE	0.70070	100%	5.52828	100%
NF - bubble 0	0.03075	4%	3.18366	57%
NF - bubble 0.70	0.12522	18%	3.82161	69%
NF - bubble 0.80	0.21697	30%	4.26000	77%
NF - bubble 0.90	0.30109	43%	4.71087	85%
NF - bubble 1	0.35814	51%	5.00970	91%

Table 15: Cost-Entropy numerical values for MovieLens pop0 a0.99 N2 C20 CPtop Q0.8 L40.

5.3.12 MovieLens pop0 a0.8 N10 C20 CPtop Q0.8 L40 No Fairness

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BASELINE	0.88560	100%	6.64107	100%
NF - bubble 0	0.39304	44%	4.91802	74%
NF - bubble 0.80	0.41120	46%	5.09041	77%
NF - bubble 0.85	0.43800	49%	5.26393	79%
NF - bubble 0.90	0.52977	60%	5.71450	86%
NF - bubble 0.95	0.63960	72%	6.11336	92%
NF - bubble 1	0.71374	80%	6.36386	96%

Table 16: Cost-Entropy numerical values for MovieLens pop0 a0.8 N10 C20 CPtop Q0.8 L40.

5.3.13 MovieLens pop0 a0.8 N2 C5 CPtop Q0.8 L40 No Fairness

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BASELINE	0.96529	100%	6.50775	100%
NF - bubble 0	0.48796	50%	4.30808	66%
NF - bubble 0.70	0.49877	51%	4.41813	68%
NF - bubble 0.80	0.53805	55%	4.96588	77%
NF - bubble 0.85	0.58133	60%	5.27592	81%
NF - bubble 0.90	0.64702	67%	5.61199	86%
NF - bubble 0.95	0.72761	75%	5.95123	92%
NF - bubble 1	0.77897	80%	6.22498	96%

Table 17: Cost-Entropy numerical values for MovieLens pop0 a0.8 N2 C5 CPtop Q0.8 L40.

5.3.14 MovieLens pop0 a0.8 N2 C20 CPtop Q0.5 L40 No Fairness

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BASELINE	0.89411	100%	6.50775	100%
NF - bubble 0	0.25024	28%	4.11403	63%
NF - bubble 0.70	0.25080	29%	4.47753	69%
NF - bubble 0.80	0.35343	40%	5.00887	77%
NF - bubble 0.85	0.40965	46%	5.27164	81%
NF - bubble 0.90	0.48795	55%	5.58126	86%
NF - bubble 0.95	0.59521	66%	5.92596	91%
NF - bubble 1	0.66438	74%	6.13496	95%

Table 18: Cost-Entropy numerical values for MovieLens pop0 a0.8 N2 C20 CPtop Q0.5 L40.

5.3.15 MovieLens pop1 a0.8 N2 C20 CPtop Q0.8 L40 No Fairness

ALGORITHM	COST	% of cost^{BS}	ENTROPY	% of H^{BS} achieved
BASELINE	0.77506	100%	5.96419	100%
NF - bubble 0	0.24239	31%	4.04675	68%
NF - bubble 0.70	0.24239	31%	4.12983	69%
NF - bubble 0.80	0.30276	39%	4.64603	78%
NF - bubble 0.85	0.35332	45%	4.89763	82%
NF - bubble 0.90	0.40461	52%	5.15431	87%
NF - bubble 0.95	0.45671	58%	5.40899	91%
NF - bubble 1	0.52839	68%	5.70310	96%

Table 19: Cost-Entropy numerical values for MovieLens pop1 a0.8 N2 C20 CPtop Q0.8 L40.

To avoid repetition, we consolidate all the results for the Movielens dataset into a single plot **in the next subsection**. This will allow us to compare and analyze the Movielens results alongside the Last.Fm results, which were also plotted together for a comprehensive evaluation of the Diverse NF-RS.

Specifically, we will present the NF entropy and NF cost of all the scenarios **as a percentage** of the BS entropy and cost. Then, we will extract our final results.

5.3.16 Conclusions for Diverse NF-RS

Last.Fm: the red line corresponds to the initial case; any line positioned above the red line signifies a **greater** cost; any line below the red line indicates a **lower** cost.

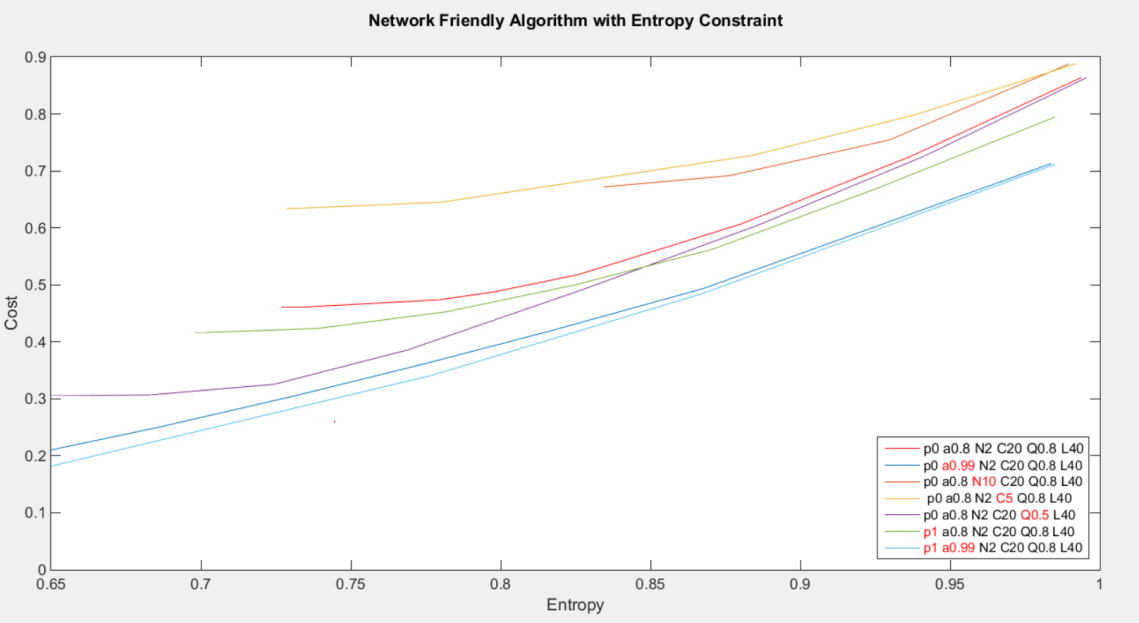


Figure 14: Cost-Entropy plot for Lastfm, **Diverse NF-RS**, all cases.

MovieLens:

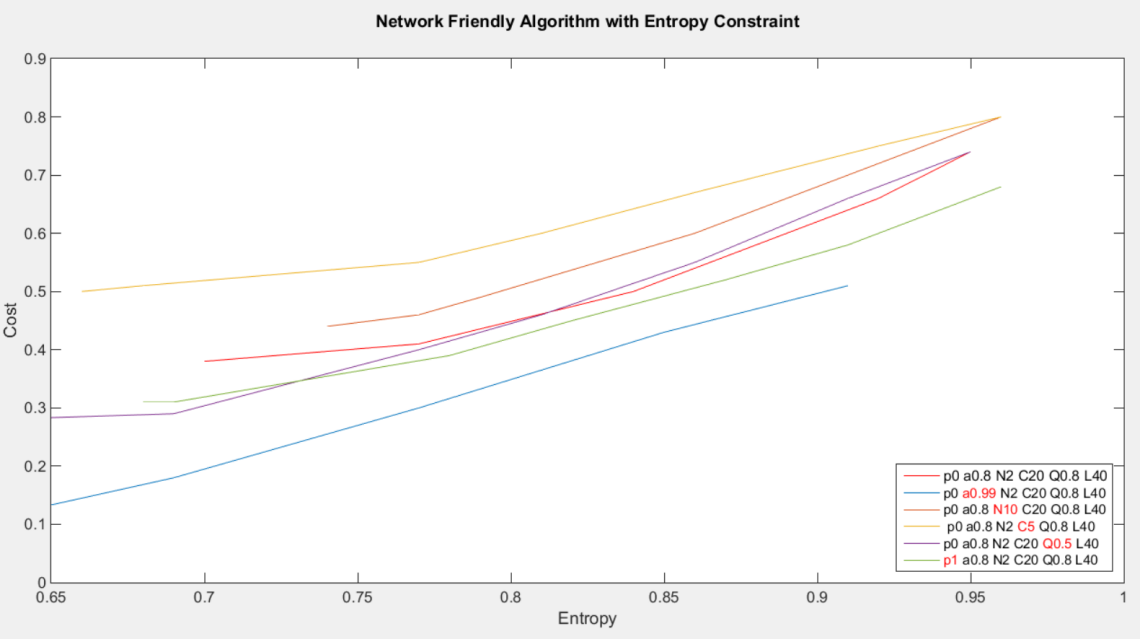


Figure 15: Cost-Entropy plot for MovieLens, **Diverse NF-RS**, all cases.

Based on the analysis of both plots, we draw conclusions regarding the optimal behaviour in terms of the trade-off between entropy and cost.

The most favourable outcomes were observed in cases characterized by:

- a larger popularity,
- higher values of α,
- smaller values of N,
- larger values of C,
- smaller values of QoR.

5.4 Fair Diverse NF-RS

We will now conduct our entropy and fairness-constrained problem for the case where: **a0.8 N2 C20 CPtop Q0.8 L40**, since these are some realistic input values. We deliberately did not opt for the less-costly scenario with $\alpha = 0.99$ due to its extreme nature, disallowing the user from discontinuing following the recommendations. Additionally, we refrained from selecting Q as 0.5 (allowing smaller costs), as it represents a very low quality of recommendations, while our aim is to maintain a greater quality level. Exploring popularity values, we found that for $\alpha = 0.8$, there wasn't a significant difference in outcomes for popularity 0 or 1. This means that a great trade-off can be achieved either by starting with a uniformly distributed vector **p0** or by a Zipf distribution. We will consider results for both values of popularity for the LstFM dataset, and for a 0 popularity for the MovieLens dataset.

Then, our goal is to answer the following questions about each fairness metric:

1. Do the achieved trade-offs become worse when this specific fairness constraint is active too (apart from the bubble constraint)?

2. Can we actually achieve the cost-bubble trade-offs for "free", as a side-effect of one of the other metrics introduced in previous work?

Below are the results obtained after running our code for **each fairness metric** (i.e. KL, MAX and TV), while also including the entropy constraint with different weights (i.e. bubble metric values).

5.4.1 Lastfm pop0 a0.8 N2 C20 CPtop Q0.8 L40 KL

ALGORITHM	FAIRNESS	COST	% of cost^{BS}	ENTROPY	% of H^{BS}
BS	-	0.89078	100%	6.31276	100%
NF - bubble 0.00	-	0.41072	46%	4.58777	72%
NF - bubble 0.80	-	0.42211	47%	4.92127	78%
NF - bubble 0.85	-	0.46149	52%	5.21413	82%
NF - bubble 0.90	-	0.54023	60%	5.55556	88%
NF - bubble 0.95	-	0.64721	72%	5.88091	93%
NF - bubble 1.00	-	0.76969	86%	6.27383	99.4%
NF - bubble 0.00	**KL 0.01**	0.77148	86%	6.1111	96.8%
NF - bubble 0.00	**KL 0.1**	0.46048	51%	5.11269	80%
NF - bubble 0.80	**KL 0.1**	0.46048	51%	5.11269	80%
NF - bubble 0.85	**KL 0.1**	0.47334	53%	5.25147	83%
NF - bubble 0.90	**KL 0.1**	0.54026	60%	5.55667	88%
NF - bubble 0.95	**KL 0.1**	0.64721	72%	5.92370	94%
NF - bubble 1.00	**KL 0.1**	0.76969	86%	6.28771	99.6%
NF - bubble 0.00	**KL 0.3**	0.41072	46%	**4.67054**	**74%**
NF - bubble 0.80	**KL 0.3**	0.42211	47%	4.92127	78%
NF - bubble 0.85	**KL 0.3**	0.46149	52%	5.21413	82%
NF - bubble 0.90	**KL 0.3**	0.54023	60%	5.55556	88%
NF - bubble 0.95	**KL 0.3**	0.64721	72%	5.92434	94%
NF - bubble 1.00	**KL 0.3**	0.76969	86%	6.28671	99.6%

Table 20: Cost-Entropy numerical values for LastFm pop0 a0.8 N2 C20 Q0.8 L40 KL

RESULTS:

1. Do the achieved trade-offs become worse when the KL fairness constraint is active too (apart from the bubble constraint)?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint alone is preserved when the 'KL' fairness constraint is also active, for KL fairness weights of 0.1 and 0.3. Therefore, we can still achieve good costs and low bubble phenomena at a fair, bubble-constrained system. In fact, the trade-off even slightly improves in the end, eventually surpassing the graph obtained with just the entropy constraint. However, for a tight KL weight (0.01), the entropy-cost trade-off is bad.

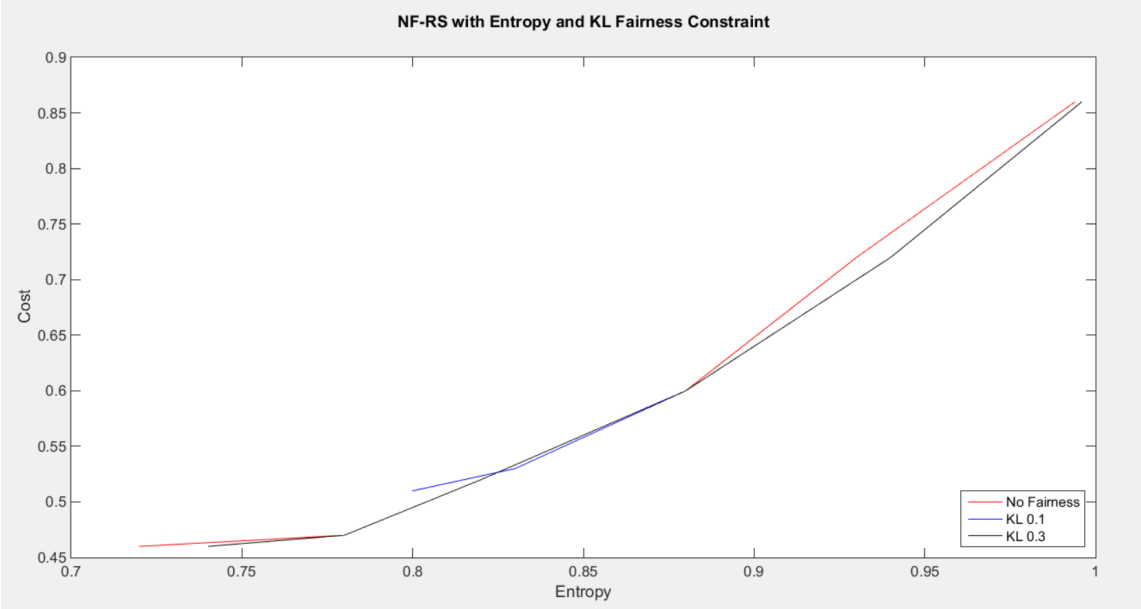


Figure 16: Cost-Entropy plot for Diverse KL-Fair NF-RS

2. Can we achieve the cost-bubble trade-offs for "free", as a side-effect of the KL metric?

   No. Indeed, an improvement in entropy is observed when the KL constraint is included, especially with an extremely tight setting. For example, with the KL fairness weight set to 0.1, the fairness constraint alone achieves a notable 8% increase in entropy with a cost sacrifice of 5%. A similar result is attained with a bubble constraint equal to 0.85 when no fairness constraint is included. Hence, for a tight KL constraint, addressing the entropy problem can be achieved up to a point by only incorporating the KL constraint. However, for higher values of fairness weight, the inclusion of the entropy constraint becomes imperative to effectively address the entropy issue. Also, the KL-Fair NF-RS might increase its entropy as the fairness weight gets smaller, but this implementation still does not come with as good a trade-off as that offered by the Diverse NF-RS (by changing the bubble weight). This can be observed below:

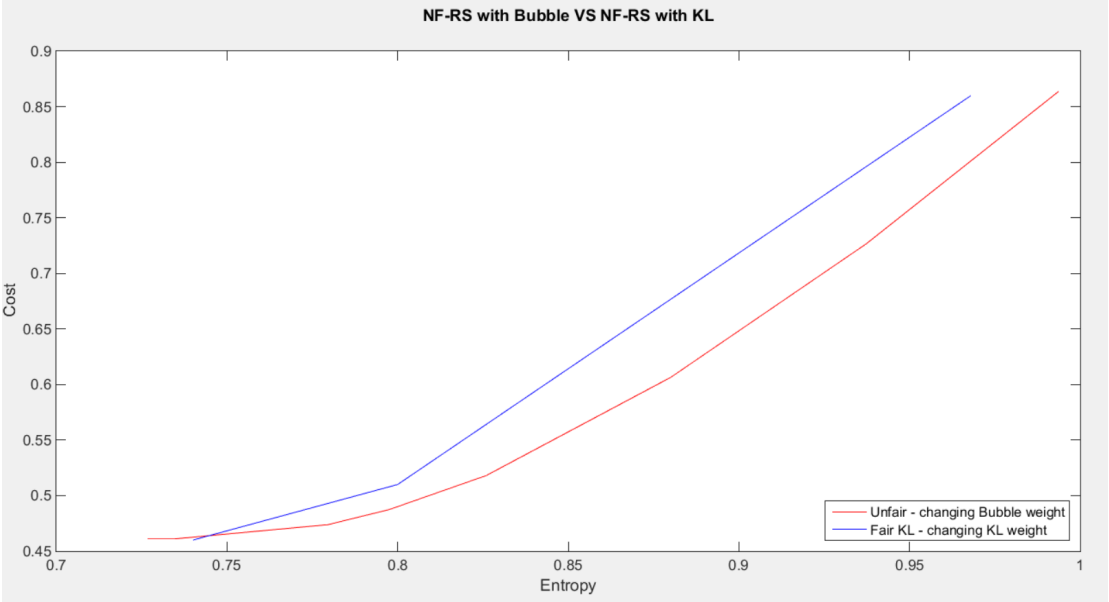


Figure 17: Cost-Entropy plot for Diverse NF-RS **Vs** KL-Fair NF-RS

5.4.2 Lastfm pop0 a0.8 N2 C20 CPtop Q0.8 L40 max

ALGORITHM	FAIRNESS	COST	% of cost^{BS}	ENTROPY	% of H^{BS}
BASELINE	-	0.89078	100%	6.31276	100%
NF - bubble 0.00	-	0.41072	46%	4.58777	72%
NF - bubble 0.80	-	0.42211	47%	4.92127	78%
NF - bubble 0.85	-	0.46149	52%	5.21413	82%
NF - bubble 0.90	-	0.54023	60%	5.55556	88%
NF - bubble 0.95	-	0.64721	72%	5.88091	93%
NF - bubble 1.00	-	0.76969	86%	6.27383	99.4%
NF - bubble 0.00	**MAX 0.1**	0.41072	46%	4.67085	74%
NF - bubble 0.80	**MAX 0.1**	0.42211	47%	4.92127	78%
NF - bubble 0.85	**MAX 0.1**	0.46149	52%	5.21413	82%
NF - bubble 0.90	**MAX 0.1**	0.54023	60%	5.55556	88%
NF - bubble 0.95	**MAX 0.1**	0.64721	72%	5.88091	93%
NF - bubble 1.00	**MAX 0.1**	0.76969	86%	6.21289	98%
NF - bubble 0.00	**MAX 0.3**	0.41072	46%	**4.62509**	**73%**
NF - bubble 0.80	**MAX 0.3**	0.42211	47%	4.92127	78%
NF - bubble 0.85	**MAX 0.3**	0.46149	52%	5.21413	82%
NF - bubble 0.90	**MAX 0.3**	0.54023	60%	5.55556	88%
NF - bubble 0.95	**MAX 0.3**	0.64721	72%	5.88091	93%
NF - bubble 1.00	**MAX 0.3**	0.76969	86%	**6.27450**	**99.4%**

Table 21: Cost-Entropy numerical values for LastFm pop0 a0.8 N2 C20 Q0.8 L40 MAX

Since the results seem to be very similar with the non-constrained problem, we also run some examples for an even smaller MAX fairness weight, i.e. for 0.01 (very constrained).

ALGORITHM	FAIRNESS	COST	% of cost^{BS}	ENTROPY	% of H^{BS}
BASELINE	-	0.89078	100%	6.31276	100%
NF - bubble 0.00	**MAX 0.01**	0.69078	77%	5.96409	94%
NF - bubble 0.95	**MAX 0.01**	0.69078	77%	6.0041	95%
NF - bubble 1.00	**MAX 0.01**	0.76969	86%	6.26231	99.2%

Table 22: Cost-Entropy numerical values for LastFm pop0 a0.8 N2 C20 Q0.8 L40 MAX

We conclude that **under highly constrained** fairness conditions, the achieved **trade-off becomes unfavorable**. Optimal trade-offs suggest avoiding excessively tight constraints on either entropy or fairness.

RESULTS:

1. Do the achieved trade-offs become worse when the MAX fairness constraint is active too (apart from the bubble constraint)?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint alone is preserved when the MAX fairness constraint is also active, provided that the MAX weight is not too tight (e.g. 0.01). The trade-off only becomes slightly worse (1.4% in contrast with the entropy of the unfair bubble-constrained problem) in the extreme case where the bubble = 1, for a small MAX weight set to 0.1.

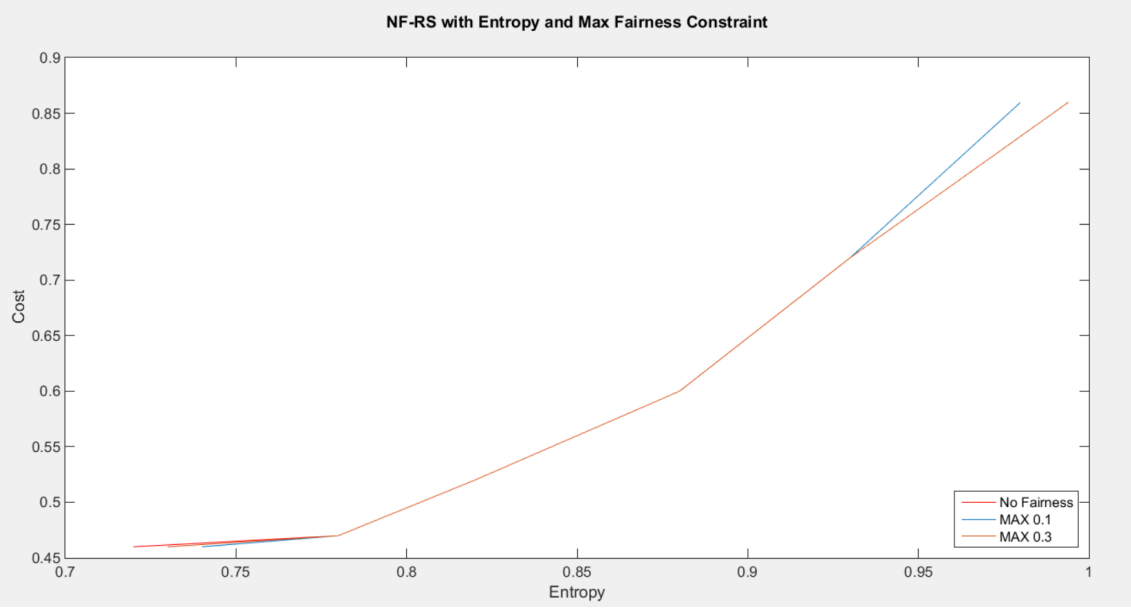


Figure 18: Cost-Entropy plot for Diverse MAX-Fair NF-RS

2. Can we achieve the cost-bubble trade-offs for "free", as a side-effect of the MAX metric?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint cannot be replicated solely by the inclusion of the MAX fairness metric. Incorporating a MAX fairness constraint - without a bubble constraint - results in a linear relationship between cost and entropy. Thus, the MAX fairness constraint alone does not effectively address the entropy problem. To achieve higher entropy values, it is essential to incorporate the entropy constraint into our problem. The trade-offs for 1) the MAX-Fair NF-RS with different MAX weight values and 2) the Fair NF-RS with different bubble values can be observed below:

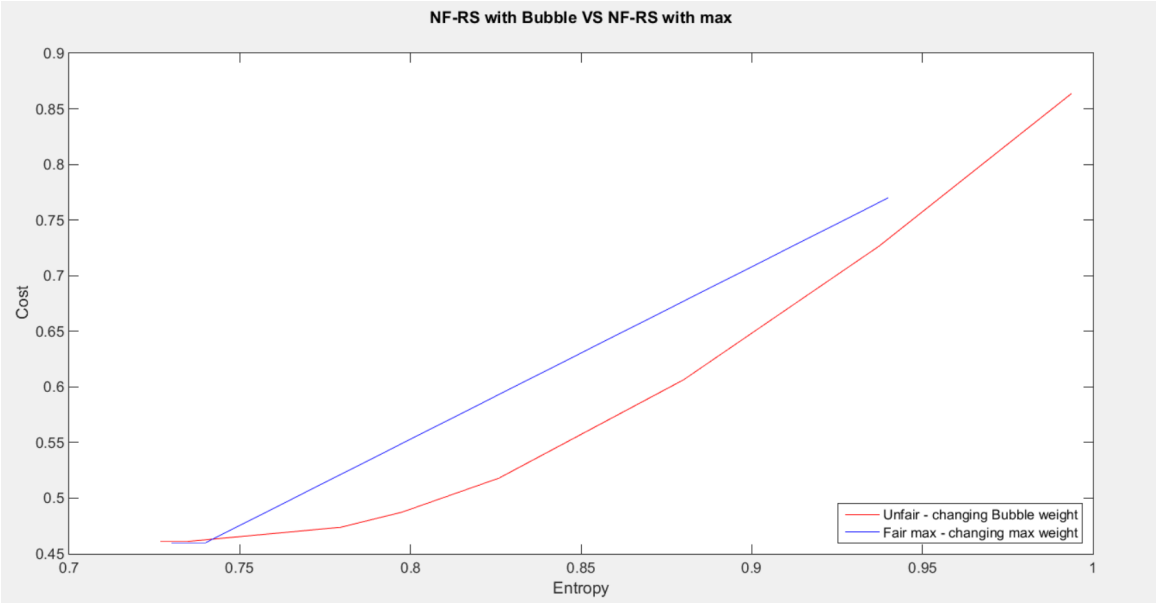


Figure 19: Cost-Entropy plot for Diverse NF-RS **Vs** MAX-Fair NF-RS

5.4.3 Lastfm pop0 a0.8 N2 C20 CPtop Q0.8 L40 TV

ALGORITHM	FAIRNESS	COST	% of cost^{BS}	ENTROPY	% of H^{BS}
BASELINE	-	0.89078	100%	6.31276	100%
NF - bubble 0.00	-	0.41072	46%	4.58777	72%
NF - bubble 0.80	-	0.42211	47%	4.92127	78%
NF - bubble 0.85	-	0.46149	52%	5.21413	82%
NF - bubble 0.90	-	0.54023	60%	5.55556	88%
NF - bubble 0.95	-	0.64721	72%	5.88091	93%
NF - bubble 1.00	-	0.76969	86%	6.27383	99.4%
NF - bubble 0.00	**TV 0.1**	0.79078	88%	6.03081	95%
NF - bubble 0.80	**TV 0.1**	0.79078	88%	6.17009	97.7%
NF - bubble 0.90	**TV 0.1**	0.79078	88%	6.17680	97.8%
NF - bubble 1.00	**TV 0.1**	0.79078	88%	6.18338	98%
NF - bubble 0.00	**TV 0.3**	0.59078	66%	**5.52909**	**87%**
NF - bubble 0.80	**TV 0.3**	0.59078	66%	5.56682	88%
NF - bubble 0.85	**TV 0.3**	0.59078	66%	5.57558	88%
NF - bubble 0.90	**TV 0.3**	0.59078	66%	5.58459	88%
NF - bubble 1.00	**TV 0.3**	0.65800	74%	5.92664	94%
NF - bubble 0.00	**TV 0.5**	0.41161	46%	4.66700	74%
NF - bubble 0.80	**TV 0.5**	0.41381	46.5%	4.80056	76%
NF - bubble 0.85	**TV 0.5**	0.43704	49%	5.04176	80%
NF - bubble 0.90	**TV 0.5**	0.47572	53.5%	5.27249	83%
NF - bubble 0.95	**TV 0.5**	0.52806	59%	5.51530	87%
NF - bubble 1.00	**TV 0.5**	0.64588	72.5%	5.87300	93%

Table 23: Cost-Entropy numerical values for LastFm pop0 a0.8 N2 C20 Q0.8 L40 TV

For tightly constrained fairness conditions (small TV weight), the entropy increases significantly even without the entropy constraint. Specifically, for bubble = 0 and:

- TV weight equal to 0.1 : a 95% of $\mathbf{H^{BS}}$ is achieved with 88% of $cost^{BS}$

- TV weight equal to 0.3 : a 87% of $\mathbf{H^{BS}}$ is achieved with 66% of $cost^{BS}$

Tightly constraining the TV fairness also addresses the entropy problem up to a certain point. Conversely, for a TV weight ≥ 0.5, the TV fairness constraint alone is not sufficient to increase diversity, and an entropy constraint has to be included. However, we again have to check whether the TV-constrained problem alone can achieve as high a cost-entropy trade-off as the bubble-constrained problem does.

RESULTS:

1. Do the achieved trade-offs become worse when the TV fairness constraint is active too (apart from the bubble constraint)?

   Not much. The favorable entropy-cost trade-off achieved by the entropy constraint alone is preserved when the TV fairness constraint is also active, if the TV fairness weight is not very tight. Again, for an extremely tight TV constraint the TV fairness constraint increases the network cost significantly, creating a more unfavorable cost-entropy trade-off.

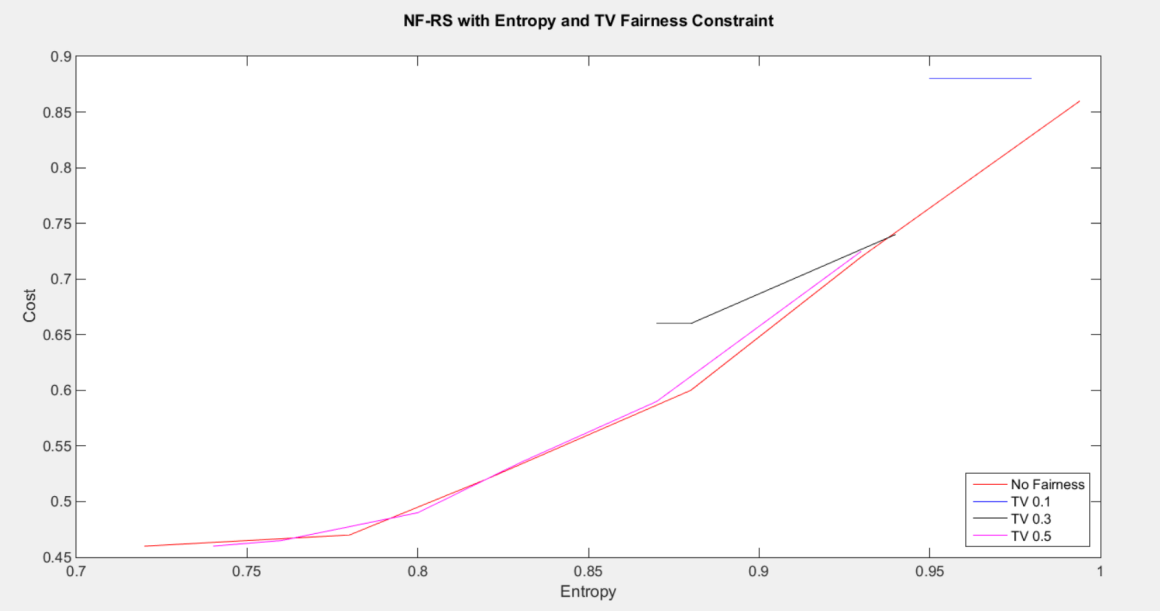


Figure 20: Cost-Entropy plot for Diverse TV-Fair NF-RS

2. Can we achieve the cost-bubble trade-offs for "free", as a side-effect of the TV metric?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint cannot be replicated solely by the inclusion of the TV fairness metric. Incorporating a TV fairness constraint - without a bubble constraint - results in a worse relationship between cost and entropy, because high entropy is indeed succeeded but it comes with a bigger cost. Thus, the TV fairness constraint alone does not effectively address the entropy problem. To achieve higher entropy values, it is essential to incorporate the entropy constraint into our problem. The trade-offs for 1) the TV-Fair NF-RS with different TV weight values and 2) the Fair NF-RS with different bubble values can be observed below:

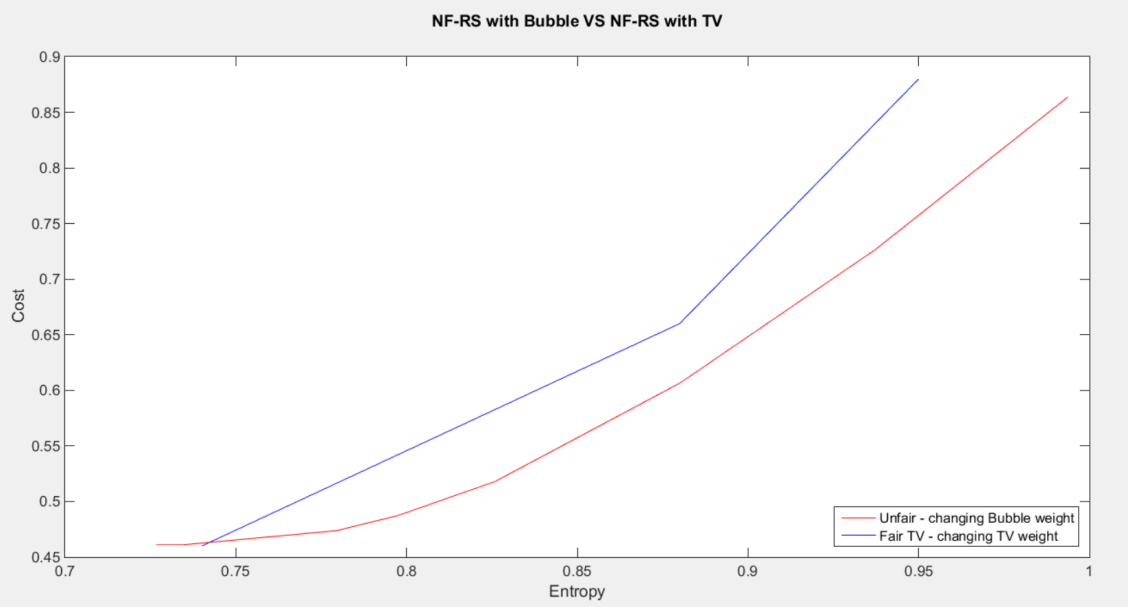


Figure 21: Cost-Entropy plot for Diverse NF-RS **Vs** TV-Fair NF-RS

The results for the three cases discussed above can be summarized as:

1. Do the achieved trade-offs become worse when the fairness constraint is active too (apart from the bubble constraint)?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint alone is preserved when another fairness constraint is also active, unless the fairness constraint is extremely tight. This scenario is extreme, so it does not concern us.

2. Can we achieve the cost-bubble trade-offs for "free", as a side-effect of the fairness metrics already implemented in the wowmom paper?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint cannot be replicated solely by the inclusion of the existing fairness metrics. Thus, these fairness constraints alone do not effectively address the entropy problem. This was demonstrated through several plots showing that while good entropy can sometimes be achieved with fairness constraints, it comes with a higher cost compared to our proposed Diverse NF-RS.

Next we will repeat the same experiments for a popularity of 1.

5.4.4 Lastfm pop1 a0.8 N2 C20 CPtop Q0.8 L40 KL

ALGORITHM	FAIRNESS	COST	% of cost^{BS}	ENTROPY	% of H^{BS}
BASELINE	-	0.79018	100%	5.91797	100%
NF - bubble 0.00	-	0.32912	41%	4.13001	69%
NF - bubble 0.70	-	0.32912	41%	4.15314	70%
NF - bubble 0.80	-	0.35796	45%	4.62883	78%
NF - bubble 0.85	-	0.39522	50%	4.88194	82%
NF - bubble 0.90	-	0.44470	56%	5.15382	87%
NF - bubble 0.95	-	0.52940	67%	5.47965	93%
NF - bubble 1.00	-	0.62774	79%	5.82893	99%
NF - bubble 0.00	**KL 0.01**	0.62608	79%	5.50991	93%
NF - bubble 0.00	**KL 0.1**	0.35688	45%	4.43312	75%
NF - bubble 0.80	**KL 0.1**	0.36788	46.5%	4.65874	79%
NF - bubble 0.85	**KL 0.1**	0.40173	50%	4.90749	83%
NF - bubble 0.90	**KL 0.1**	0.44750	56.5%	5.16215	87%
NF - bubble 0.95	**KL 0.1**	0.52940	67%	5.48100	92.6%
NF - bubble 1.00	**KL 0.1**	0.62774	79.5%	5.84035	98.6%
NF - bubble 0.00	**KL 0.3**	0.32912	42%	**4.10646**	**69%**
NF - bubble 0.80	**KL 0.3**	0.35796	45%	4.62883	78%
NF - bubble 0.85	**KL 0.3**	0.39522	50%	4.88194	82.5%
NF - bubble 0.90	**KL 0.3**	0.44470	56%	5.15383	87%
NF - bubble 0.95	**KL 0.3**	0.52940	67%	5.48123	92.6%
NF - bubble 1.00	**KL 0.3**	0.62774	79%	5.84125	98.7%

Table 24: Cost-Entropy numerical values for LastFm pop1 a0.8 N2 C20 Q0.8 L40 KL

The outcomes closely resemble those observed in the scenario where popularity is set to 0. This suggests that positive results can be achieved for both types of initial content demand **p0** (uniform and Zipf distributions).

RESULTS:

1. Does the achieved trade-off become worse when the KL fairness constraint is active too (apart from the bubble constraint)?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint alone is preserved when the KL fairness constraint is also active, as depicted bellow:

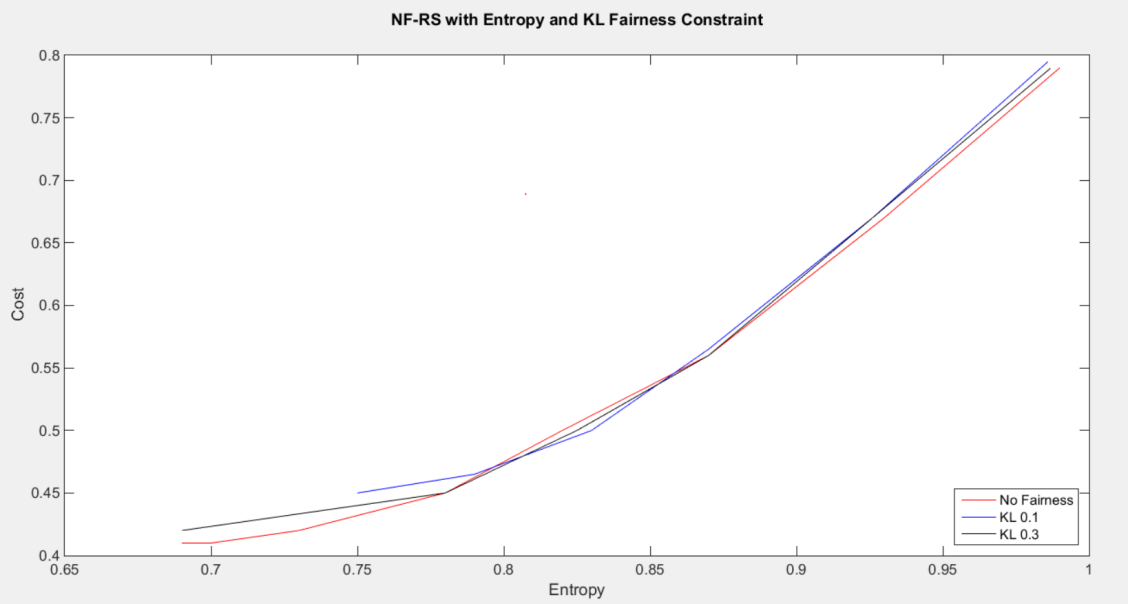


Figure 22: Cost-Entropy plot for Diverse KL-Fair NF-RS

2. Can we achieve the cost-bubble trade-off for "free", as a side-effect of the KL fairness metric?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint cannot be replicated solely by the inclusion of the KL fairness metric. This is demonstrated in the following plot:

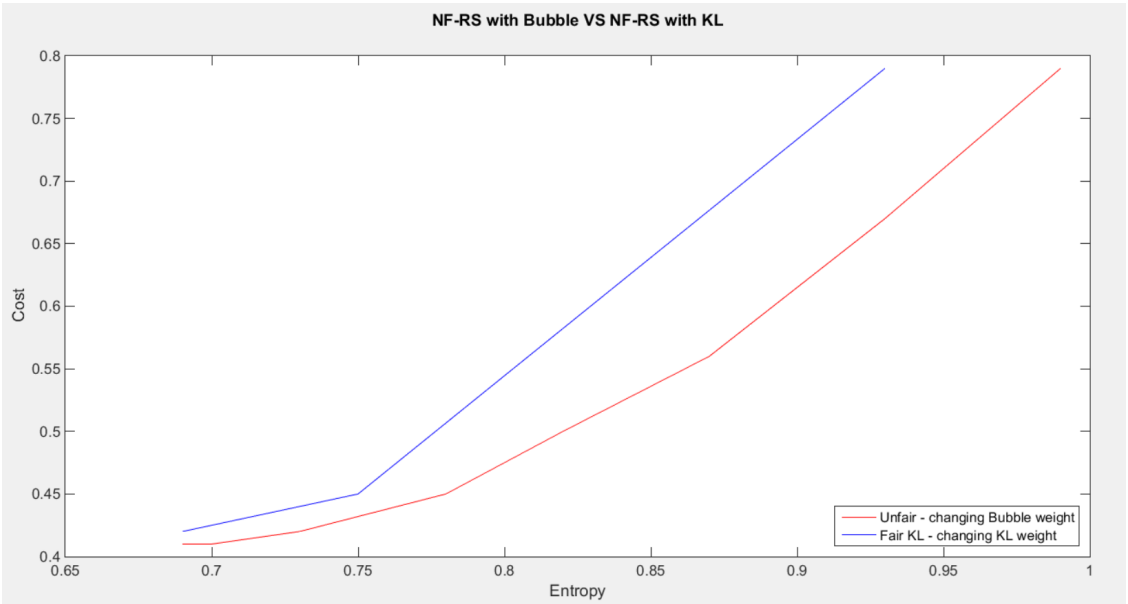


Figure 23: Cost-Entropy plot for Diverse NF-RS **Vs** KL-Fair NF-RS

5.4.5 Lastfm pop1 a0.8 N2 C20 CPtop Q0.8 L40 MAX

ALGORITHM	FAIRNESS	COST	% of cost^{BS}	ENTROPY	% of H^{BS}
BASELINE	-	0.79018	100%	5.91797	100%
NF - bubble 0.00	-	0.32912	41%	4.13001	69%
NF - bubble 0.70	-	0.32912	41%	4.15314	70%
NF - bubble 0.75	-	0.33467	42%	4.37356	73%
NF - bubble 0.80	-	0.35796	45%	4.62883	78%
NF - bubble 0.85	-	0.39522	50%	4.88194	82%
NF - bubble 0.90	-	0.44470	56%	5.15382	87%
NF - bubble 0.95	-	0.52940	67%	5.47965	93%
NF - bubble 1.00	-	0.62774	79%	5.82893	99%
NF - bubble 0.00	**MAX 0.01**	0.59019	75%	5.16763	87%
NF - bubble 0.00	**MAX 0.1**	0.32912	41.6%	4.16964	70%
NF - bubble 0.80	**MAX 0.1**	0.35796	45%	4.62884	78%
NF - bubble 0.85	**MAX 0.1**	0.39522	50%	4.88194	82.5%
NF - bubble 0.90	**MAX 0.1**	0.44470	56%	5.15382	87%
NF - bubble 0.95	**MAX 0.1**	0.52940	67%	5.47228	92.5%
NF - bubble 1.00	**MAX 0.1**	0.62774	79%	5.78765	97.8%
NF - bubble 0.00	**MAX 0.3**	0.32912	41%	**4.10553**	**69%**
NF - bubble 0.80	**MAX 0.3**	0.35796	45%	4.62883	78%
NF - bubble 0.85	**MAX 0.3**	0.39522	50%	4.88194	82.5%
NF - bubble 0.90	**MAX 0.3**	0.44470	56%	5.15382	87%
NF - bubble 0.95	**MAX 0.3**	0.52940	67%	5.47006	92.5%
NF - bubble 1.00	**MAX 0.3**	0.62774	79.5%	5.78578	98%

Table 25: Cost-Entropy numerical values for LastFm pop1 a0.8 N2 C20 Q0.8 L40 MAX

The outcomes closely resemble those observed in the scenario where popularity is set to 0. This suggests that positive results can be achieved for both types of initial content demand **p0** (uniform and Zipf distributions).

RESULTS:

1. Does the achieved trade-off become worse when the MAX fairness constraint is active too (apart from the bubble constraint)?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint alone is preserved when the MAX fairness constraint is also active, as depicted bellow:

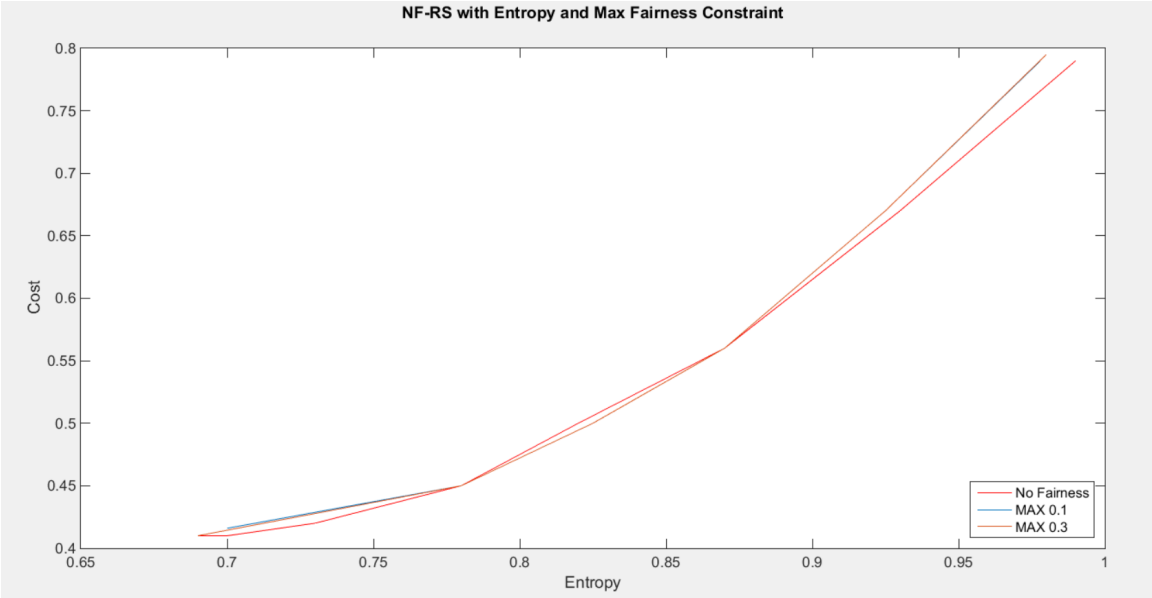


Figure 24: Cost-Entropy plot for Diverse MAX-Fair NF-RS

2. Can we achieve the cost-bubble trade-off for "free", as a side-effect of the MAX fairness metric?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint cannot be replicated solely by the inclusion of the MAX fairness metric. This is demonstrated in the following plot:

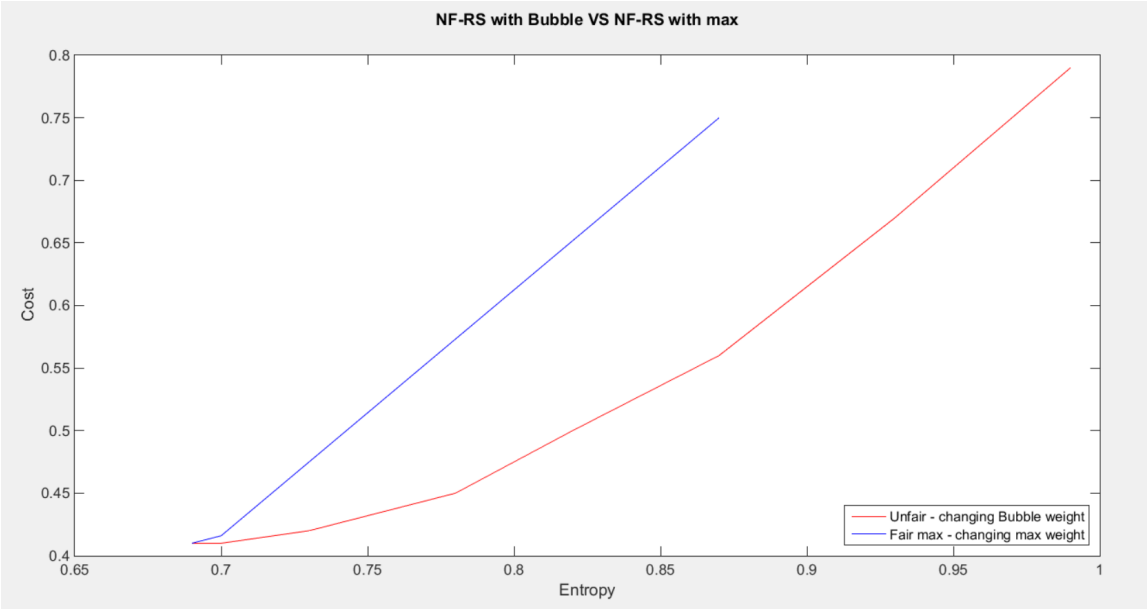


Figure 25: Cost-Entropy plot for Diverse NF-RS **Vs** MAX-Fair NF-RS

5.4.6 Lastfm pop1 a0.8 N2 C20 CPtop Q0.8 L40 TV

ALGORITHM	FAIRNESS	COST	% of cost^{BS}	ENTROPY	% of H^{BS}
BASELINE	-	0.79018	100%	5.91797	100%
NF - bubble 0.00	-	0.32912	41%	4.13001	69%
NF - bubble 0.80	-	0.35796	45%	4.62883	78%
NF - bubble 0.85	-	0.39522	50%	4.88194	82%
NF - bubble 0.90	-	0.44470	56%	5.15382	87%
NF - bubble 0.95	-	0.52940	67%	5.47965	93%
NF - bubble 1.00	-	0.62774	79%	5.82893	99%
NF - bubble 0.00	**TV 0.1**	0.69018	87.4%	5.75497	97.3%
NF - bubble 0.80	**TV 0.1**	0.69018	87.4%	5.76927	97.5%
NF - bubble 0.85	**TV 0.1**	0.69018	87.4%	5.77121	97.5%
NF - bubble 0.90	**TV 0.1**	0.69018	87.4%	5.77146	97.5%
NF - bubble 0.95	**TV 0.1**	0.69018	87.4%	5.77146	97.5%
NF - bubble 1.00	**TV 0.1**	0.69018	87.4%	5.78433	97.8%
NF - bubble 0.00	**TV 0.3**	0.49018	62%	5.06851	85.6%
NF - bubble 0.80	**TV 0.3**	0.49018	62%	5.08230	85.8%
NF - bubble 0.85	**TV 0.3**	0.49018	62%	5.08864	86%
NF - bubble 0.90	**TV 0.3**	0.49018	62%	5.12652	86.6%
NF - bubble 0.95	**TV 0.3**	0.51360	65%	5.35546	90%
NF - bubble 1.00	**TV 0.3**	0.56759	72%	5.59507	95%
NF - bubble 0.00	**TV 0.5**	0.32916	41%	4.13990	70%
NF - bubble 0.80	**TV 0.5**	0.35040	44%	4.55558	77%
NF - bubble 0.85	**TV 0.5**	0.38157	48%	4.78398	80%
NF - bubble 0.90	**TV 0.5**	0.42181	53%	5.01288	85%
NF - bubble 0.95	**TV 0.5**	0.46822	59%	5.24227	89%
NF - bubble 1.00	**TV 0.5**	0.51836	65%	5.46200	92%

Table 26: Cost-Entropy numerical values for LastFm pop1 a0.8 N2 C20 Q0.8 L40 TV

RESULTS:

1. Do the achieved trade-offs become worse when the TV fairness constraint is active too (apart from the bubble constraint)?

   A little bit. The favorable entropy-cost trade-off achieved by the entropy constraint alone is preserved when the TV fairness constraint is also active, as long as the fairness constraint is not too tight. For a tighter TV constraint, the trade-off is slightly worse, as depicted below. However, we are not very concerned about this, as it is just one example among many (and the difference is not huge either).

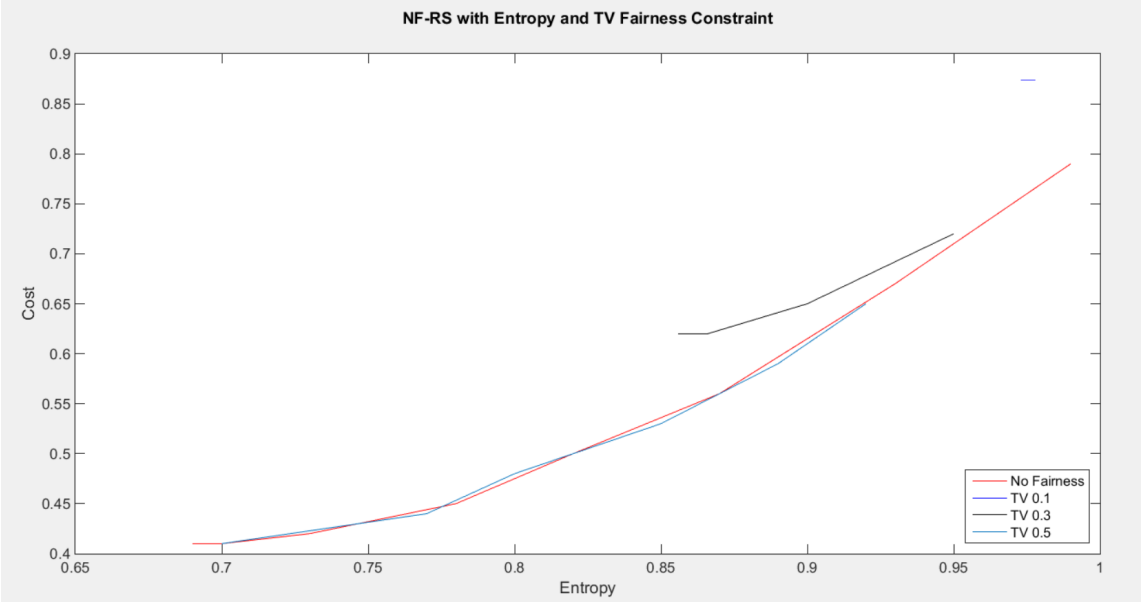


Figure 26: Cost-Entropy plot for Diverse TV-Fair NF-RS

2. Can we achieve the cost-bubble trade-off for "free", as a side-effect of the TV fairness metric?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint cannot be replicated solely by the inclusion of the TV fairness metric. This is demonstrated in the following plot:

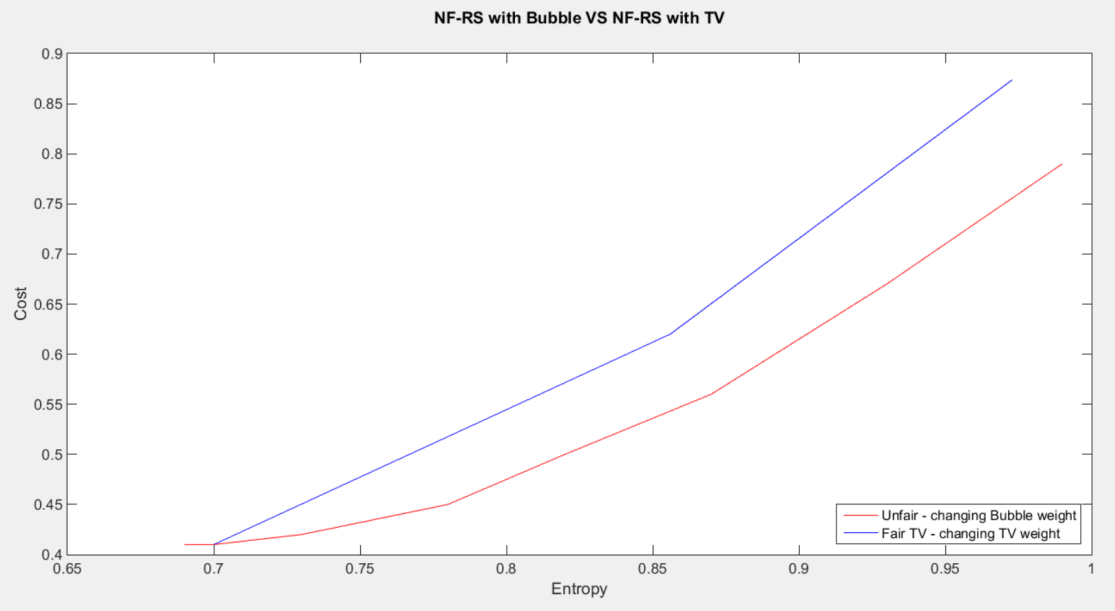


Figure 27: Cost-Entropy plot for Diverse NF-RS **Vs** TV-Fair NF-RS

As for the MovieLens dataset the results are similar, provng that they are generic and not only due to the specific dataset of LastFM. They are presented bellow:

5.4.7 MovieLens pop0 a0.8 N2 C20 CPtop Q0.8 L40 KL

ALGORITHM	FAIRNESS	COST	% of cost^{BS}	ENTROPY	% of H^{BS}
BASELINE	-	0.89411	100%	6.50775	100%
NF - bubble 0.00	-	0.33977	38%	4.58428	70%
NF - bubble 0.80	-	0.36793	41%	5.02457	77%
NF - bubble 0.85	-	0.45472	50%	5.44687	84%
NF - bubble 0.90	-	0.48819	54%	5.58523	86%
NF - bubble 0.95	-	0.59521	66%	5.93344	92%
NF - bubble 1.00	-	0.66438	74%	6.14127	95%
NF - bubble 0.00	**KL 0.01**	0.75947	85%	6.24048	95%
NF - bubble 0.00	**KL 0.1**	0.44337	49%	5.28015	81%
NF - bubble 0.80	**KL 0.1**	0.44337	49%	5.28015	81%
NF - bubble 0.85	**KL 0.1**	0.45978	51%	5.40084	83%
NF - bubble 0.90	**KL 0.1**	0.51173	57%	5.65455	87%
NF - bubble 0.95	**KL 0.1**	0.59521	66%	5.98588	92%
NF - bubble 1.00	**KL 0.1**	0.71355	80%	6.39409	98%
NF - bubble 0.00	**KL 0.3**	0.33977	38%	**4.77925**	**73%**
NF - bubble 0.80	**KL 0.3**	0.36793	41%	5.02457	77%
NF - bubble 0.85	**KL 0.3**	0.42031	47%	5.28759	81%
NF - bubble 0.90	**KL 0.3**	0.48819	55%	5.58524	86%
NF - bubble 0.95	**KL 0.3**	0.59521	66%	5.98457	92%
NF - bubble 1.00	**KL 0.3**	0.71355	80%	6.39282	98%

Table 27: Cost-Entropy numerical values for LastFm pop0 a0.8 N2 C20 Q0.8 L40 KL

RESULTS:

1. Do the achieved trade-offs become worse when the KL fairness constraint is active too (apart from the bubble constraint)?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint alone is preserved when the KL fairness constraint is also active. The main difference is that, for a very tight KL constraint, the minimum cost achieved is larger than that of the unfair problem, as depicted below.

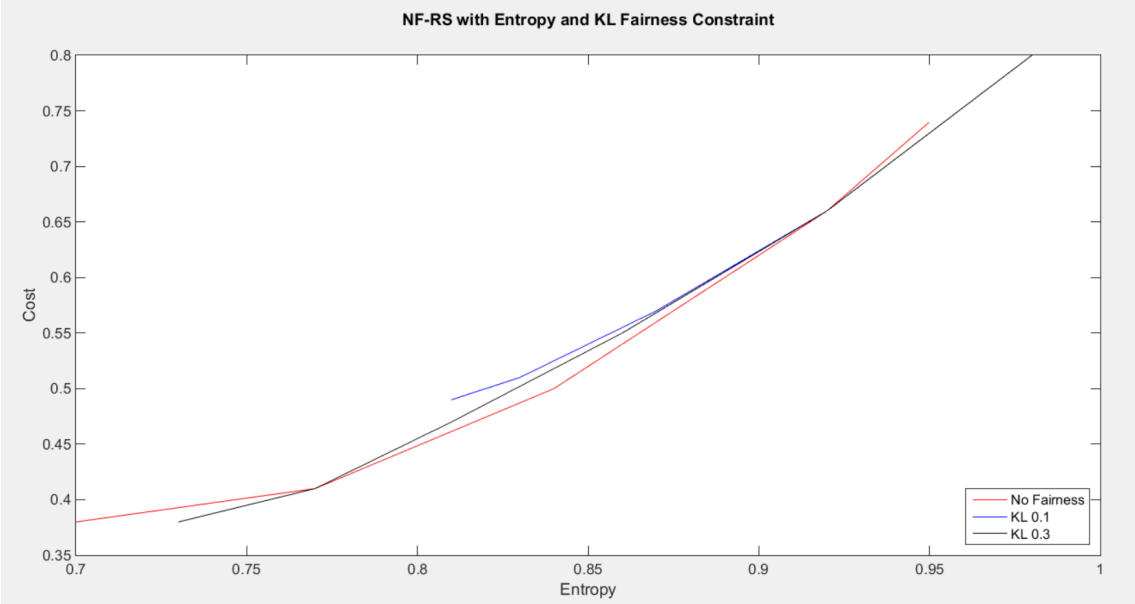


Figure 28: Cost-Entropy plot for Diverse KL-Fair NF-RS

2. Can we achieve the cost-bubble trade-off for "free", as a side-effect of the KL fairness metric?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint cannot be replicated solely by the inclusion of the KL fairness metric:

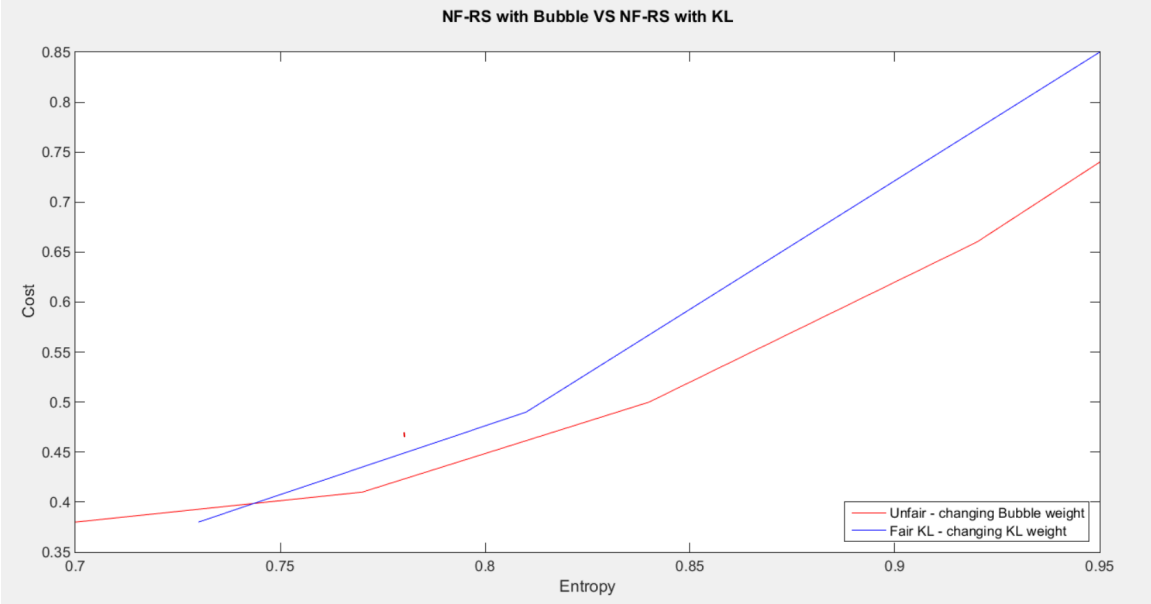


Figure 29: Cost-Entropy plot for Diverse NF-RS **Vs** KL-Fair NF-RS

5.4.8 MovieLens pop0 a0.8 N2 C20 CPtop Q0.8 L40 MAX

ALGORITHM	FAIRNESS	COST	% of cost^{BS}	ENTROPY	% of H^{BS}
BASELINE	-	0.89411	100%	6.50775	100%
NF - bubble 0.00	-	0.33977	38%	4.58428	70%
NF - bubble 0.80	-	0.36793	41%	5.02457	77%
NF - bubble 0.85	-	0.45472	50%	5.44687	84%
NF - bubble 0.90	-	0.48819	54%	5.58523	86%
NF - bubble 0.95	-	0.59521	66%	5.93344	92%
NF - bubble 1.00	-	0.66438	74%	6.14127	95%
NF - bubble 0.00	**MAX 0.01**	0.69411	78%	6.15510	94%
NF - bubble 0.00	**MAX 0.1**	0.33977	38%	4.71714	72%
NF - bubble 0.80	**MAX 0.1**	0.36793	41%	5.02457	77%
NF - bubble 0.85	**MAX 0.1**	0.42031	47%	5.28517	81%
NF - bubble 0.90	**MAX 0.1**	0.48819	55%	5.58524	85%
NF - bubble 0.95	**MAX 0.1**	0.59521	66%	5.93553	91%
NF - bubble 1.00	**MAX 0.1**	0.71355	80%	6.36928	97%
NF - bubble 0.00	**MAX 0.3**	0.33977	38%	**4.77110**	**73%**
NF - bubble 0.80	**MAX 0.3**	0.36793	41%	5.02457	77%
NF - bubble 0.85	**MAX 0.3**	0.42031	47%	5.28517	81%
NF - bubble 0.90	**MAX 0.3**	0.48819	55%	5.58524	85%
NF - bubble 0.95	**MAX 0.3**	0.59521	66%	5.93206	91%
NF - bubble 1.00	**MAX 0.3**	0.71355	80%	6.36928	97%

Table 28: Cost-Entropy numerical values for LastFm pop0 a0.8 N2 C20 Q0.8 L40 MAX

RESULTS:

1. Do the achieved trade-offs become worse when the MAX fairness constraint is active too (apart from the bubble constraint)?

   No. The favorable entropy-cost trade-off attained by the entropy constraint alone experiences a minor deterioration when the MAX fairness constraint is also in effect. This impact is not significant, and thus, the mentioned issue does not cause any concern for us.

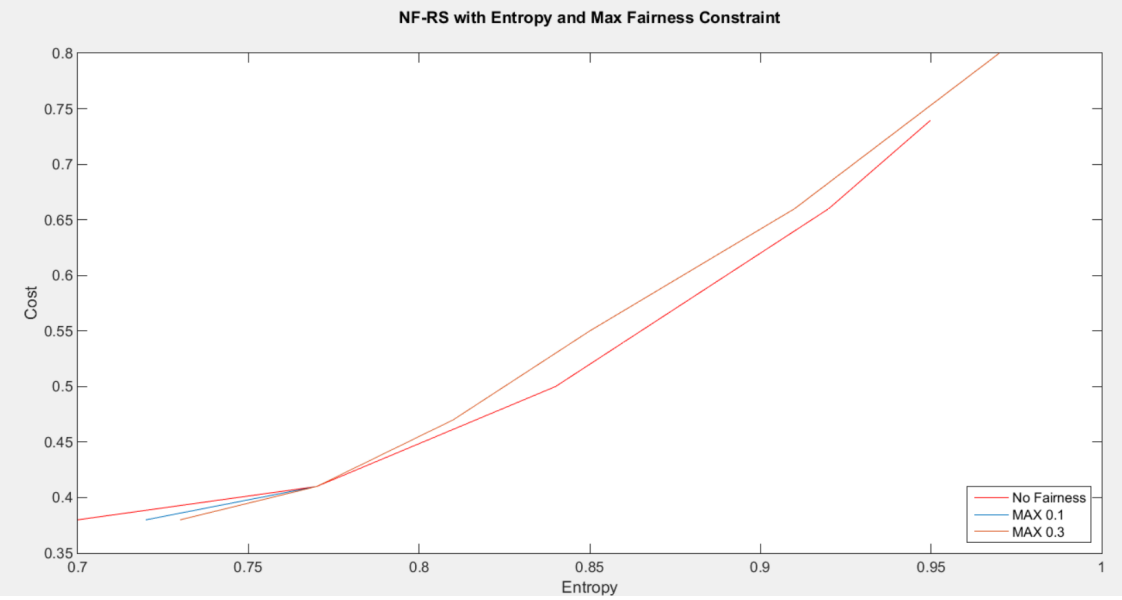


Figure 30: Cost-Entropy plot for Diverse MAX-Fair NF-RS

2. Can we achieve the cost-bubble trade-off for "free", as a side-effect of the MAX fairness metric?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint cannot be replicated solely by the inclusion of the MAX fairness metric. This is demonstrated in the following plot:

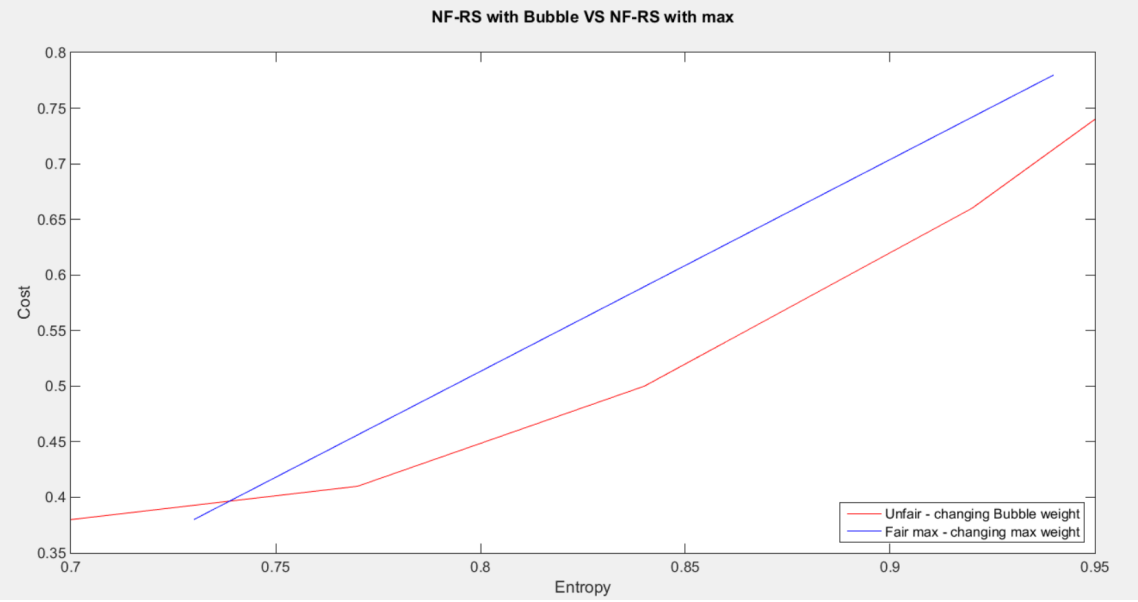


Figure 31: Cost-Entropy plot for Diverse NF-RS **Vs** MAX-Fair NF-RS

5.4.9 MovieLens pop0 a0.8 N2 C20 CPtop Q0.8 L40 TV

ALGORITHM	FAIRNESS	COST	% of cost^{BS}	ENTROPY	% of H^{BS}
BASELINE	-	0.89411	100%	6.50775	100%
NF - bubble 0.00	-	0.33977	38%	4.58428	70%
NF - bubble 0.80	-	0.36793	41%	5.02457	77%
NF - bubble 0.85	-	0.45472	50%	5.44687	84%
NF - bubble 0.90	-	0.48819	54%	5.58523	86%
NF - bubble 0.95	-	0.59521	66%	5.93344	92%
NF - bubble 1.00	-	0.66438	74%	6.14127	95%
NF - bubble 0.00	**TV 0.01**	0.88411	99%	6.48699	99%
NF - bubble 0.00	**TV 0.1**	0.79411	89%	6.34325	97%
NF - bubble 0.00	**TV 0.3**	0.59411	66%	**5.60110**	**86%**
NF - bubble 0.80	**TV 0.3**	0.59411	66%	5.73341	88%
NF - bubble 0.90	**TV 0.3**	0.59411	66%	5.73341	88%
NF - bubble 0.95	**TV 0.3**	0.59411	66%	5.87591	90%
NF - bubble 1.00	**TV 0.3**	0.65390	73%	6.13212	94%
NF - bubble 0.00	**TV 0.5**	0.39411	44%	5.06093	77%
NF - bubble 0.85	**TV 0.5**	0.41063	45%	5.21910	80%
NF - bubble 0.90	**TV 0.5**	0.467465	52%	5.46122	84%
NF - bubble 0.95	**TV 0.5**	0.52554	58%	5.70384	88%
NF - bubble 1.00	**TV 0.5**	0.58508	65%	5.94414	91%

Table 29: Cost-Entropy numerical values for LastFm pop0 a0.8 N2 C20 Q0.8 L40 TV

RESULTS:

1. Do the achieved trade-offs become worse when the TV fairness constraint is active too (apart from the bubble constraint)?

   Only for too tight TV constraints. The favorable entropy-cost trade-off attained by the entropy constraint alone is preserved when the TV fairness constraint is also in effect, except from the cases where the TV constraint is too tight (e.g. for a TV weight of 0.3). This behavior can be seen bellow:

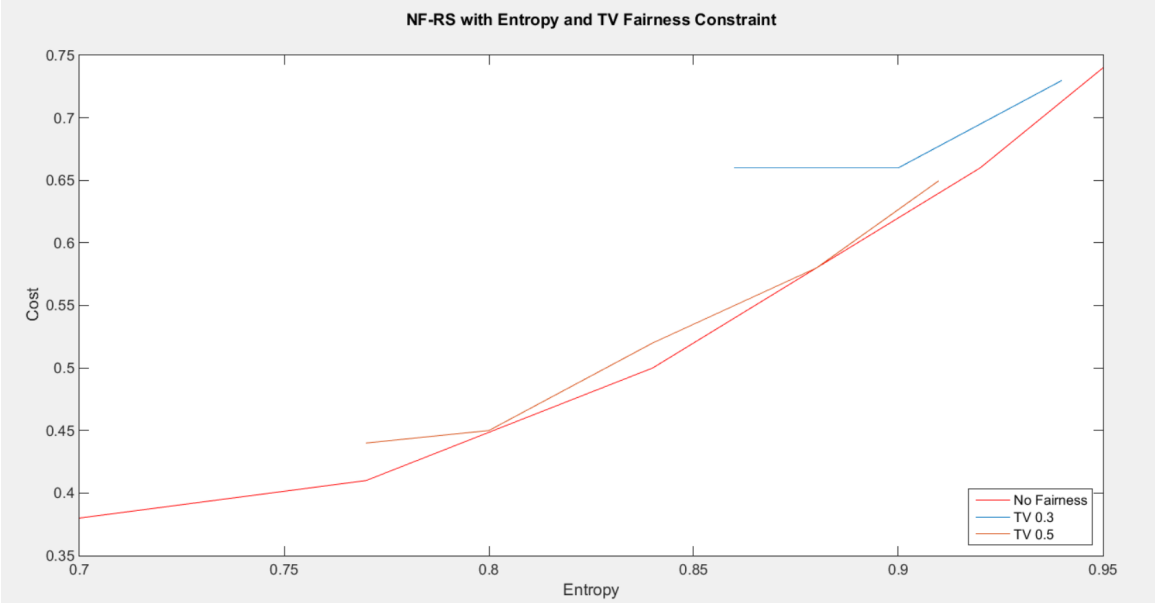


Figure 32: Cost-Entropy plot for Diverse TV-Fair NF-RS

2. Can we achieve the cost-bubble trade-off for "free", as a side-effect of the TV fairness metric?

   No. The favorable entropy-cost trade-off achieved by the entropy constraint cannot be replicated solely by the inclusion of the TV fairness metric. This is demonstrated in the following plot:

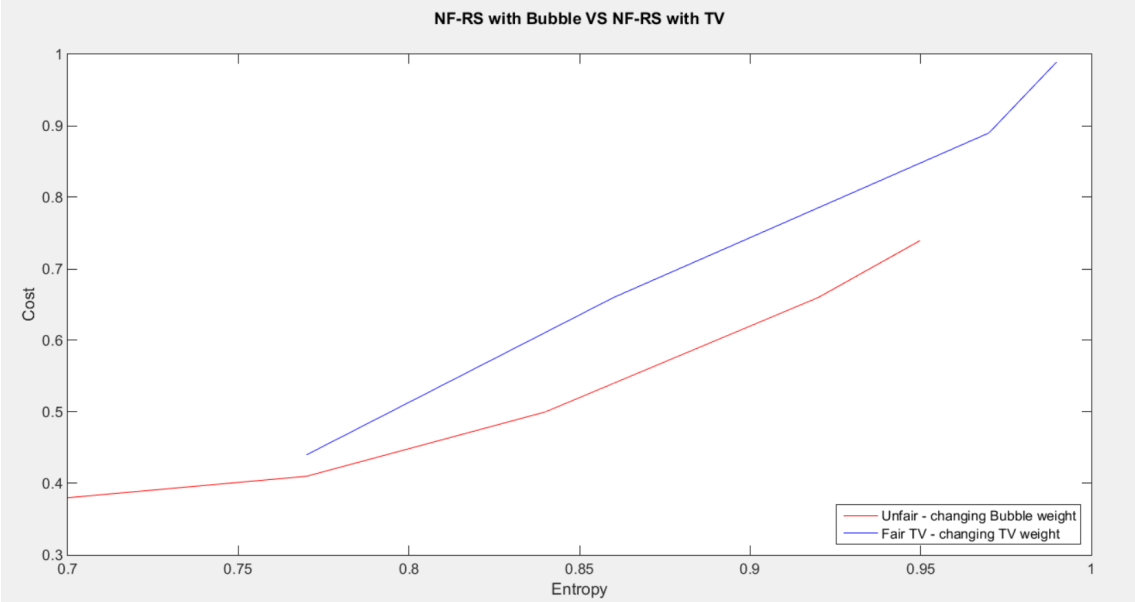


Figure 33: Cost-Entropy plot for Diverse NF-RS **Vs** TV-Fair NF-RS

6 Discussion and Future Work

This thesis has delved into an examination of content bubbles, a phenomenon attributed to the heightened degree of personalization in contemporary RSs. The emphasis given nowadays on minimizing network costs - leading to the development of NF-RS - contributes to the creation of content bubbles by further constraining content diversity. Our research has **confirmed this diversity restriction**, particularly in NF-RS, where constant suggestions usually exhibit a singular nature. After highlighting the inadequacy of existing implementations in addressing this issue, a **specific solution** for the content bubble problem **is presented**. Our approach is designed to minimize the bubble phenomenon created by vanilla recommenders, while maintaining cost efficiency. The introduced framework is capable of providing **optimal trade-offs between cost and diversity** across various scenarios. Furthermore, the bubble metric introduced in this study complements existing fairness metrics in relevant literature in two ways : 1) it performs a distinct role that other metrics fail to achieve, and 2) it can collaborate with existing metrics without undermining each other's efficacy.

Regarding possible future work, the potential use of the Gini coefficient as an alternative measure of diversity has also been proposed in a related thesis. The formal definition of the Gini index enables its use as a measure of diversity, providing an opportunity to substitute the entropy constraint with a Gini index constraint. Although it was not implemented in this study due to time constraints, this work lays the **groundwork** for exploring it in future research.

Another promising direction for future research could involve introducing dynamic adjustments to the number of recommended contents and cached contents, rather than relying on static inputs. This approach aims to achieve a more optimal trade-off. It is essential for this strategy to be implemented with the collaboration of the platforms, acknowledging the potential for dynamic changes in these values. The degree of flexibility should be carefully considered, ensuring that adjustments are within reasonable limits; e.g. recommending 5 contents instead of 7, if this proves to offer a more favorable trade-off, and the platform approves such changes.

Last, future work could involve exploring different ways of picking just **one** recommendation out of the N suggestions provided. The user's choice depends on the model employed, allowing for generalization to other models in future research. In our work, this selection is random (a uniform distribution is used). However, [6] suggests a different perspective, where the user chooses the most relevant item based on what they just viewed. Something similar could also be implemented.

References

[1] T. Spyropoulos and P. Sermpezis, "Soft cache hits and the impact of alternative content recommendations on mobile edge caching," in *Proc. ACM Workshop on Challenged Networks (CHANTS)*, 2016.

[2] L.-E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Caching-aware recommendations: Nudging user preferences towards better caching performance," in *Proc. IEEE INFOCOM,* 2017.

[3] P. Sermpezis, T. Giannakas, T. Spyropoulos, and L. Vigneri, "Soft cache hits: Improving performance through recommendation and delivery of related content," *I*EEE Journal Selected Areas in Communications, 2018.

[4] T. Giannakas, P. Sermpezis, and T. Spyropoulos, "Show me the cache: Optimizing cache-friendly recommendations for sequential content access," in *P*roc. IEEE WoWMoM, 2018.

[5] S. Kastanakis, P. Sermpezis, V. Kotronis, and X. Dimitropoulos, "Cabaret: Leveraging recommendation systems for mobile edge caching," in *P*roc. ACM SIGCOMM workshops, 2018.

[6] T. Giannakas, P. Sermpezis, and T. Spyropoulos, "Network Friendly Recommendations: Optimizing for Long Viewing Sessions," in *I*EEE Trans. on Mobile Comp., 2020.

[7] B. Zhu andW. Chen, "Coded caching with joint content recommendation and user grouping," in *P*roc. IEEE GLOBECOM, 2018.

[8] L. E. Chatzieleftheriou, M. Karaliopoulos, and I. Koutsopoulos, "Jointly optimizing content caching and recommendations in small cell networks," *I*EEE Trans. on Mobile Computing, vol. 18, no. 1, 2019.

[9] M. Costantini, T. Spyropoulos, T. Giannakas, and P. Sermpezis, "Approximation guarantees for the joint optimization of caching and recommendation," in *P*roc. IEEE ICC, 2020.

[10] M. Garetto, E. Leonardi, and G. Neglia, "Similarity caching: Theory and algorithms," in *P*roc. IEEE INFOCOM, 2020.

[11] K. Qi, B. Chen, C. Yang, and S. Han, "Optimizing caching and recommendation towards user satisfaction," in *I*EEE WCSP, 2018.

[12] T. Giannakas, T. Spyropoulos, and P. Sermpezis, "The order of things: Position-aware network-friendly recommendations in long viewing sessions," in *P*roc. WiOpt, 2019.

[13] L. Chatzieleftheriou, G. Darzanos, M. Karaliopoulos, and I. Koutsopoulos, "Joint user association, content caching and recommendations in wireless edge networks," *P*ER, vol. 46, no. 3, pp. 12–17, 2019.

[14] S. Gupta and S. Moharir, "Effect of recommendations on serving content with unknown demand," *A*CM TOMPECS, vol. 4, no. 1, p. 4, 2019.

[15] Z. Lin and W. Chen, "Joint pushing and recommendation for susceptible users with time-varying connectivity," in *I*EEE GLOBECOM, 2018.

[16] L. Song and C. Fragouli, "Making recommendations bandwidth aware," *I*EEE Trans. Information Theory, vol. 64, no. 11, 2018.

[17] Z. Lin and W. Chen, "Content pushing over multiuser miso downlinks with multicast beamforming and recommendation: A cross-layer approach," *I*EEE Trans. on Communications, vol. 67, no. 10, 2019.

[18] T. Giannakas, A. Giovanidis, and T. Spyropoulos, "Soba: Session optimal mdp-based network friendly recommendations," in *P*roc. IEEE INFOCOM, 2021.

[19] P. Sermpezis, S. Kastanakis, J. I. Pinheiro, F. Assis, D. Menasch´e, and T. Spyropoulos, "Towards qos-aware recommendations," in *A*CM RecSys workshops (CARS workshop), 2020.

[20] D. Krishnappa, M. Zink, C. Griwodz, and P. Halvorsen, "Cache-centric video recommendation: an approach to improve the efficiency of youtube caches," *A*CM Transactions on Multimedia Computing, Communications, and Applications (TOMM), vol. 11, no. 4, p. 48, 2015.

[21] D. Munaro, C. Delgado, and D. S. Menasch´e, "Content recommendation and service costs in swarming systems," in *P*roc. IEEE ICC, 2015.

[22] Haupt Jon, "Last.fm: People-Powered Online Radio", in *Music Reference Services Quarterly*, 2010.

[23] Baris Kirdemir and Nitin Agarwal, "Exploring Bias and Information Bubbles in YouTube's Video Recommendation Networks", in *COMPLEX NETWORKS 2021*, 2022.

[24] F. Maxwell Harper, Joseph A. Konstan, "The MovieLens Datasets: History and Context", in *ACM Transactions on Interactive Intelligent Systems*, 2015.

[25] Baris Kirdemir, Joseph Kready, Esther Mead, Muhammad Hussain, Nitin Agarwal and Donald Adjeroh, "Assessing Bias in YouTube's Video Recommendation Algorithm in a Cross-lingual and Cross-topical Context", in *Social, Cultural, and Behavioral Modeling, 14th International Conference*, 2021.

[26] H. Abdollahpouri and R. Burke, "Multi-stakeholder recommendation and its connection to multi-sided fairness," in *P*roc. RMSE workshop at ACM RecSys, 2019.

[27] R. Burke, "Multisided fairness for recommendation," in *W*orkshop on Fairness, Accountability, Transparency in Machine Learning, 2017.

[28] R. Burke, N. Sonboli, and A. Ordonez-Gauger, "Balanced neighborhoods for multi-sided fairness in recommendation," in *C*onference on Fairness, Accountability and Transparency, 2018, pp. 202–214.

[29] B. Edizel, F. Bonchi, S. Hajian, A. Panisson, and T. Tassa, "Fairecsys: Mitigating algorithmic bias in recommender systems," *I*nternational Journal of Data Science and Analytics, vol. 9, no. 2, pp. 197–213, 2020.

[30] G. K. Patro, A. Chakraborty, N. Ganguly, and K. Gummadi, "Incremental fairness in two-sided market platforms: On smoothly updating recommendations", in *P*roc. AAAI conf. on Artificial Intelligence, 2020.

[31] D. Pessach and E. Shmueli, "Algorithmic fairness", in *a*rXiv preprint arXiv:2001.09784, 2020.

[32] D. Sacharidis, K. Mouratidis, and D. Kleftogiannis, "A common approach for consumer and provider fairness in recommendations", in *P*roc. ACM RecSys (Late-breaking Results), 2019.

[33] H. Steck, "Calibrated recommendations", in Proc. ACM RecSys, 2018.

[34] K. Yang and J. Stoyanovich, "Measuring fairness in ranked outputs", in *P*roc. SSDBM, 2017.

[35] W. Liu and R. Burke, "Personalizing fairness-aware re-ranking", in *P*roc. FATREC workshop at ACM RecSys, 2018.

[36] T. Giannakas, P. Sermpezis, A. Giovanidis, T. Spyropoulos, and G. Arvanitakis, "Fairness in Network Friendly Recommendations" in *I*EEE WoWMoM, 2021.

[37] Zeshan Fayyaz,Mahsa Ebrahimian,Dina Nawara,Ahmed Ibrahim andRasha Kashef, "Recommendation Systems: Algorithms, Challenges, Metrics, and Business Opportunities", 2020

[38] Yifan Wang, Weizhi Ma, Min Zhang, Yiqun Liu, and Shaoping Ma, "A Survey on the Fairness of Recommender Systems", in *A*CM Transactions on Information Systems, 2023.

[39] Michael D. Ekstrand, Mucun Tian, Ion Madrazo Azpiazu, Jennifer D. Ekstrand, Oghenemaro Anuyah, David McNeill, and Maria Soledad Pera, "All the cool kids, how do they fit in?: Popularity and demographic biases in recommender evaluation

and effectiveness.", in *P*roceedings of the 1st Conference on Fairness, Accountability and Transparency, 2018.

[40] Rashidul Islam, Kamrun Naher Keya, Ziqian Zeng, Shimei Pan, and James Foulds, "Debiasing career recommendations with neural fair collaborative filtering", in *P*roceedings of the Web Conference Association for Computing Machinery , New York, NY, 3779–3790, 2021.

[41] Yingqiang Ge, Shuchang Liu, Ruoyuan Gao, Yikun Xian, Yunqi Li, Xiangyu Zhao, Changhua Pei, Fei Sun, Junfeng Ge, Wenwu Ou, and Yongfeng Zhang. "Towards long-term fairness in recommendation", in *Proceedings of the 14th ACM International Conference on Web Search and Data Mining Association for Computing Machinery, New York, NY, 445–453*, 2021.

[42] Gourab K. Patro, Arpita Biswas, Niloy Ganguly, Krishna P. Gummadi, and Abhijnan Chakraborty, "FairRec: Two-sided fairness for personalized recommendations in two-sided platforms.", in *Proceedings of the Web Conference Association for Computing Machinery*, 2020.

[43] Paolo Cremonesi, Franca Garzotto, Sara Negro, Alessandro Vittorio Papadopoulos and Roberto Turrin , "Looking for "Good" Recommendations: A Comparative Evaluation of Recommender Systems", in *IFIP Conference on Human-Computer Interaction*, 2011.

[44] Athanasios N. Nikolakopoulos, Xia Ning, Christian Desrosiers and George Karypis Chapter, "Trust Your Neighbors: A Comprehensive Survey of Neighborhood-Based Methods for Recommender Systems", 2021.

[45] Athanasios N. Nikolakopoulos, Maria Kalantzi and John D. Garofalakis, "On the Use of Lanczos Vectors for Efficient Latent Factor-Based Top-N Recommendation", in *WIMS '14: Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics*, 2014.

[46] Yusfi Adilaksa, Aina Musdholifah, "Recommendation System for Elective Courses using Content-based Filtering and Weighted Cosine Similarity", in *4th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2021.

[47] Matevž Kunaver, Tomaž Požrl, "Diversity in recommender systems – A survey", in *Elsevier B.V : www.elsevier.com/locate/knosys*, 2017.

[48] M. Harchol-Balter, Performance Modeling and Design of Computer Systems: Queueing Theory in Action. New York, NY, USA: Cambridge Univ. Press, 2013.