World Wide Web (2021) 24:1995–2018
https://doi.org/10.1007/s11280-021-00946-8

A fairness-aware multi-stakeholder recommender system

Naime Ranjbar Kermany[1,2] · Weiliang Zhao[1] · Jian Yang[1] · Jia Wu[1] · Luiz Pizzato[2]

Received: 21 December 2020 / Revised: 3 July 2021 / Accepted: 2 September 2021 /
Published online: 22 September 2021

Abstract

Traditional recommender systems mainly focus on the accuracy of recommendation, which lead to recommender systems reinforcing popular items and ignoring lesser-known items. There is increasing evidence that providing good recommendations of surprising items can lead to better user satisfaction. Users may be delightfully surprised if long-tail items are brought to them. Marketplaces need to keep providers satisfied by making sure that their items get enough exposure. In this work, we propose a fairness-aware multi-stakeholder recommender system that uses a multi-objective evolutionary algorithm to make a trade-off between provider coverage, long-tail inclusion, personalized diversity, and recommendation accuracy. Experimental results against real-world datasets show that the proposed method significantly improves the diversity of recommended items in a personalized matter and the coverage of providers with no or minor loss of accuracy.

Keywords Multi-stakeholder recommender systems · Long-tail recommendation · Multi-objective evolutionary optimization · P-fairness · Personalized diversity

✉ Naime Ranjbar Kermany
naime.ranjbar-kermany@mq.edu.au

Weiliang Zhao
weiliangzhao.email@gmail.com

Jian Yang
jian.yang@mq.edu.au

Jia Wu
jia.wu@mq.edu.au

Luiz Pizzato
luiz.pizzato1@cba.com.au

[1] Department of Computing, Macquarie University, Sydney, NSW 2019, Australia

[2] AI Labs, Commonwealth Bank of Australia, Sydney, NSW 2015, Australia

1 Introduction

Recommender system (RS) has been emerged as an information filtering tool for e-commerce. These systems learn users' preferences to predict the rating of unknown items and provide personalized recommendation for them [4].

The academic research in RSs mostly focus on providing the personalized recommendations that best meet the needs of users [2]. It is quite reasonable as users will leave the system if the systems cannot provide their desired items. However, users are one kind of the most significant stakeholders in any RS but not the only one [3]. There are many real-world recommendation domains in which the needs of other stakeholders are important to be taken into account. The consideration of the objectives of various stakeholders in the recommendation process is called as multi-stakeholder recommendation [3]. A Multi-Stakeholder Recommender System (MSRS) consists of three main stakeholders: the users, the providers, and the system [3]. Users want to get relevant, personalized, and diverse recommendations that match their needs. Providers supply the items for the system and gain utility from users' choice. The system supports both users and providers and balances their interests. An example of MSRS is Booking.com[1] with three main stakeholders of the travelers (as users), the hotels/airlines (as providers), and the Booking.com website (as the system). Travelers expect to receive relevant and diverse recommendations; hotels and airlines expect to be given a fair exposure to different users so they have enough customers; and the website seeks to be as more satisfied as possible with the bookings' commissions. Booking.com cannot survive without the existence of any of these three stakeholders and therefore it needs to take all of their preferences into account. This example is beyond the traditional accuracy-focused RS in which the needs and preferences of users are the only consideration.

According to [34], accuracy-focused RSs can achieve high user utility, but they may bring in massive unfair disparity in the exposure of the providers and adversely impact the system for its long term run. Providers have to wait for exposure and the exposure determines the revenues for them. For instance, high exposure on Spotify rises the traffic to a music provider's channel, and accordingly help them earn better advertisement revenues. On the other hand, typically a few providers get most of the exposure and the other providers struggle to survive in the system, and hence they may shift to other systems [20]. Provider unfairness may cause the items belong to some providers often appear in the recommendation lists, while the items belong to the other providers do not have a comparable exposure, leading to skew in the emergence of providers in the recommendations [29]. This restricts the choices for the users and reduces their overall satisfaction. Thus, it is important to consider the provider fairness (referred as P-fairness) in MSRSs. P-fairness lies in balancing across various providers rather than only concentrating on certain popular ones [20]. By considering P-fairness in recommendations, the new and less popular providers can have more chances to be explored. P-fairness has several benefits for the stakeholders: (1) it leads to more sale and profitability, (2) it gives the chance to providers and the system to have their brand top in mind with potential and current customers, (3) it increases the word-of-mouth marketing and customer loyalty, (4) it brings in new customers and boost customer retention. Simply recommending K items from the least exposed providers can be one way to achieve p-fairness in recommendation. However, this may result in loss of user utilities and makes the recommendations unfair to the users. Thus, the system should try to fairly

[1] https://www.booking.com

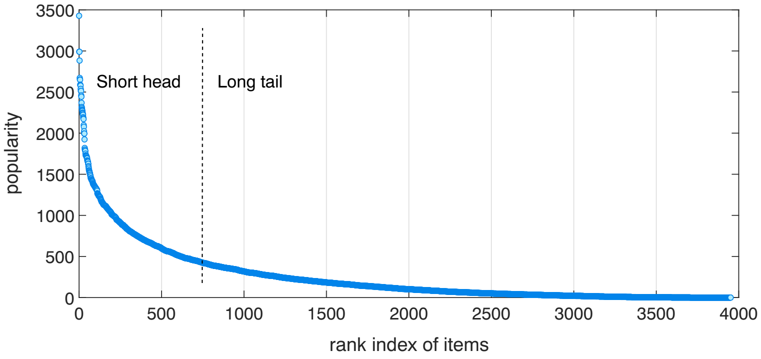


Figure 1 The share of popular (short head) and long-tail items in MovieLens 1M dataset

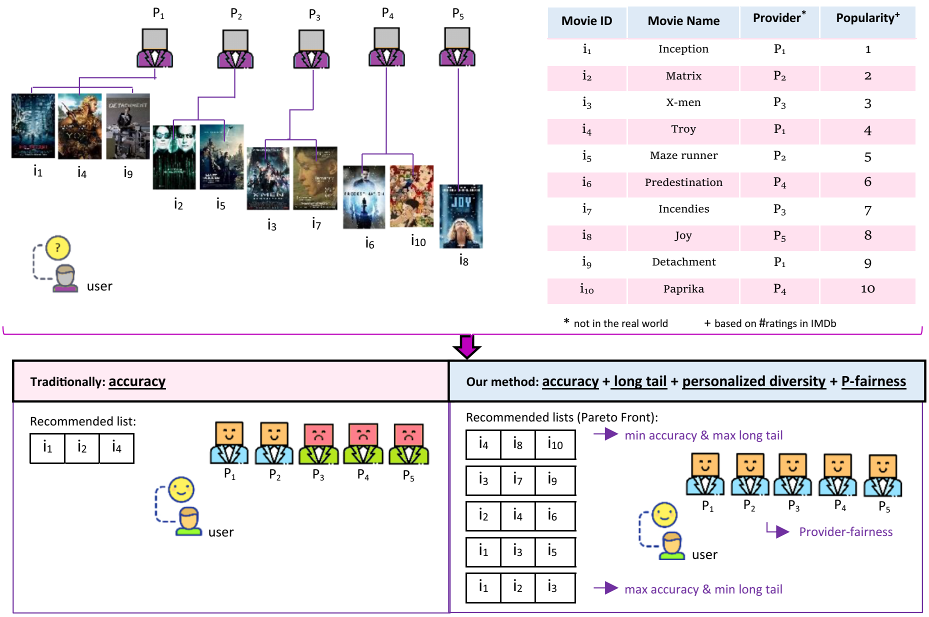
distribute the utility loss among all the users while being fair to both users and providers of the system.

There are two significant limitations with existing state-of-the-art fairness-aware recommendation studies: (1) the majority of existing work focuses on fairness only towards users of the system, meaning that they attempt to only improve the fairness of one stakeholder, ignoring the fairness of other stakeholders in the system [13, 23, 24]; (2) existing work with multi-stakeholder view typically optimizes the recommendation list for the target user by easily including items from different providers [29, 30, 34]. However, we believe that the item recommendations should be optimized considering both users' and providers' objectives. To do so, we propose a personalized diversification strategy to recommend long-tail items from less known providers by considering the users' level of interest to diversity.

Diversity in recommendation has several advantages for all stakeholders of the system. Users will receive more diverse and serendipitous recommendations. More providers will be satisfied as their items got exposure. Finally, the system will gain more profits of matching more users and providers. Diversity in recommendation can be achieved through long-tail recommendation [15]. Long-tail recommendation refers to the strategy of targeting a large number of niche items [5]. As a real example, Figure 1 shows the huge share of long-tail items in MovieLens dataset.[2] In the long-tail multi-stakeholder recommendation studies, the diversity is considered at the user level, and an identical strategy should not be applied to all users [15]. Some users may be interested in specific content while others may prefer to interact with a broader range of contents. Long-tail recommendation makes users surprised if niche items are brought to the interest of the right user. Thus, it is of interest to perform the diversification strategy in a personalized manner based on user's level of interest to diversity.

In this work, we propose a fairness-aware MSRS to increase the personalized diversity and the P-fairness of the recommendation while the accuracy is almost kept. The proposed objective functions are: (1) the recommendation accuracy, (2) the inclusion of long-tail items, (3) the personalized diversity, and (4) the P-fairness. There is no single solution available to optimize all these contradictory objectives at the same time [33]. Therefore, we use Multi-Objective Evolutionary Algorithms (MOEAs) to find a set of possible solutions at one run [43]. Figure 2 is a toy example to show the comparison of accuracy-focused RS

[2] https://grouplens.org/datasets/movielens/1m/



Movie ID	Movie Name	Provider*	Popularity+
i_1	Inception	P_1	1
i_2	Matrix	P_2	2
i_3	X-men	P_3	3
i_4	Troy	P_1	4
i_5	Maze runner	P_2	5
i_6	Predestination	P_4	6
i_7	Incendies	P_3	7
i_8	Joy	P_5	8
i_9	Detachment	P_1	9
i_{10}	Paprika	P_4	10

* not in the real world + based on #ratings in IMDb

Figure 2 A movie recommendation example to compare the traditionally accuracy-focused method, and our method considering long-tail inclusion and P-fairness

and our fairness-aware MSRS in a movie-domain recommendation. In the traditional RS, the recommendation list only focuses on the recommendation accuracy. In our method, the recommendation lists are generated by considering different trade-offs among the objective functions. Note that the recommendation lists are personalized based on the interest level of a user in the long-tail recommendation. With our method, explorer users are recommended with relevant diverse items that may delightfully surprise them. Providers are satisfied as their items get more exposure and they can survive and stay in the system. The system gains many advantages (e.g. growth of the loyal customer and the loyal provider bases) from successfully matching item recommendations with users based on both providers' and users' objectives. To the best knowledge of the authors, empirical study on investigating the best strategy to include fair components for both users and providers by considering diversity in MSRSs is still lacking in the literature. We carry out experiments against two real-world datasets to compare our results with existing studies. In particular, we address the following research questions:

- **RQ1**: How does the proposed term "P-fairness" affect the performance of the recommendation model?
- **RQ2**: Does the proposed diversification strategy -considering the users' level of interest with diversity- achieve state-of-the-art performance compared with baseline methods?
- **RQ3**: How does the multi-objective optimization affect the results comparing with traditional and re-ranking methods?
- **RQ4**: How to test that improvements by the proposed method are statistically significant?

In summary, this work makes the following contributions:

- We propose a fairness-aware multi-stakeholder recommender system by considering the objectives of users and providers in a fair way;
- We propose a personalized diversification method to consider the interest level of the user in long-tail recommendation;
- We develop a P-fairness algorithm by calculating the exposure distribution of providers and making them more satisfied;
- We develop four objective functions that reflect the accuracy, the inclusion of long-tail items, the personalized diversity, and the P-fairness of the recommendation results;
- We propose a multi-objective optimization algorithm to get the optimal solutions with a trade-off among the objective functions;
- We carry out a set of experiments against real-world datasets to show the significant improvement of P-fairness and personalized recommendation diversity while the loss of accuracy is small.

The remainder of this paper is organized as follows. We review some related work in Section 2. Section 3 presents the proposed method. The experimental results are provided in Section 4 followed by the conclusion and future work in Section 5.

2 Background and related work

This work is related with several research topics as “multi-stakeholder recommender system”, “fairness in recommender system”, “long tail recommendations”, and “personalized diversification of the recommendations”.

2.1 Accuracy-focused recommender system

Traditional accuracy-focused RS aims to recommend the most desirable items to a target user. The only objective of traditional RSs is to improve the accuracy of the recommendation [18, 19, 26, 39, 40]. The item which is worth suggesting must be recommended based on the prediction of user’s preference for the item. Collaborative Filtering (CF) is the most popular and effective algorithm of recommender systems. CF approach has been known as a accuracy-focused recommender system with the only objective of improving the recommendation accuracy [19]. However, it has been recognized that the objectives of other stakeholders have to be considered in a fairness-aware recommender system (e.g. whether the list of recommendations is diverse, whether it contains novel items to surprise users, whether the providers are satisfied as most of their items have explored, and so on [17]). Consequently, it is of interest to MSRSs, which have shifted the focus of recommender systems research, to cover a wider range of objectives.

2.2 Multi-stakeholder recommender system

MSRS combines the preferences of different parties of which the user is one. For instance, a system might promote certain items in the interest of fairness towards item providers. In such a system, we do not interpret the output as strictly reflecting the user’s preferences. In fact, MSRS develops a deeper understanding of how an organization might consider the perspectives of different stakeholders in designing a recommender system. Generally, there are three main stakeholders in a MSRS as follows [3]:

- Users: The users are those entities that receive the recommendations. They are the individuals who browse the website to find the items that meet their needs.
- Providers: The providers are those who supply the recommended items, and gain benefits from the user's selection.
- System: The system is the platform that create the RS in order to match users and providers to gain utility from successfully doing so. The system may be an e-commerce website, retailer, broker, or other platform where users look for recommendations

2.3 Fairness in recommendation

The fairness-aware recommender systems have been gaining a lot of attention recently. Zhu et al. [42] proposed a fairness-aware tensor-based recommendation framework to find solutions for overcoming the algorithmic discrimination. The majority of existing work focus on fairness for users of the system. Burke et al. [6] introduced a balanced neighborhood mechanism to preserve personalization in recommendation while enhancing the fairness of recommendation outcomes against users within a specific demographic feature such as gender or age. Li et al. [24] showed that RSs behaved unfairly by classifying users into different groups according to their level of activity. Garcia and Bonchi [14] studied the problem of minimizing the amount of user unfairness introduced when enforcing group-fairness constraints in ranking. There are few existing works with multi-stakeholder view on fairness [29, 30, 34]. Liu et al. [25] presented a fairness-aware re-ranking strategy to balance the ranking quality and provider fairness by trading off between accuracy and the coverage of the providers. Modani et al. [32] proposed a re-ranking algorithm to increase exposure distribution across the providers that results in improving the provider fairness without much affecting the accuracy of recommendations. However, these studies optimize the recommendation list for the target user ignoring both users' and providers' objectives at the same time. A good fairness-aware multi-stakeholder RS should attempt to increase the item exposure for the providers while the accuracy loss is not big. In such a system, providers would receive fair exposure for their items, whether they are popular or long tail [3].

2.4 Long-tail recommendation

RSs suggest items to target users based on their prior feedbacks/ratings. Hence, they tend not to recommend items with limited historical data, even if these items would be rated favorably by the users. Therefore, RSs can create a rich-get-richer effect for popular items while ignoring the long-tail items. The term "long tail" is first presented by Anderson in [5]. According to Anderson's definition, the long-tail items are those items with low popularity in the system. Moreover, RSs mostly recommend items very similar to what the users have already purchased or liked in the past [1]. However, this over-specialization of recommendations is often inconsistent with sale's goals and users' preferences. Thus, several research works have recently focused on long-tail recommendation. Yin et al. [41] proposed a long-tail recommendation solution based on the indirect edge-weighted graph representation, and hitting time to exploit the less popular items. Domingues et al. [12] presented a long-tail recommendation approach for music recommender systems. Authors in [9] carried out extensive experiments to evaluate the performances of various RS on the task of long-tail recommendation, and their experimental results showed that recommending long-tail items causes decrease in the accuracy of RS. To address this problem, researchers has recently focused on multi-objective optimization algorithms to make a trade-off between accuracy and long-tail recommendation [15, 33, 38]. Long-tail recommendation also achieve an increase in the diversity of recommendation lists. Diversification has become one of the leading topics of recommender system research not only as a way to solve the over-fitting problem but also an approach to increasing the quality of the recommendations [22]. Most of the available research work have used an identical diversification strategy for all users, whereas the diversity of recommendation list can be personalized according to users' interests [15].

2.5 Personalized diversity in recommendation

In recommender systems, diversity is one of the most significant matters of concern as recommending more distinct items, avoiding redundancy, and solving the over-fitting problem are brought through diversification [17]. Moreover, it has been of interest for recommender systems to provide the recommendation lists that are adjusted according to users' preferences. Shi et al. [36] proposed a method to diversify the recommendation results for individual users. They made use of the variance of the latent user factors and uncertainty of the user profiles to indicate users' interest for diversity. Chen et al. [8] studied a method to adjust the diversity within the recommendations by incorporating the users' personality. Di Noia et al. [11] conducted a research to work on individual diversity. They use Shannon's entropy to model users' tendency to accept diverse recommendations. They proved that a user who selected many diverse items in the past is more willing to receive diverse recommendation. Afterwards, in a recent study, Hamedani and Kaedi [15] proposed a long-tail recommendation solution while the accuracy is almost kept. They focused on the statement that "different users might prefer different levels of diversity in recommendations".

3 The proposed method

In this paper, the multi-stakeholder recommendation is presented as a 4-objectives optimization problem. Multi-Objective Optimization (MOO) has been introduced to optimize multiple contradictory objective functions at the same time. A significant feature of MOO is that, there is no single solution available that maximizes all the objectives simultaneously. The studies on MOEA are rich and various approaches have been proposed [31] including NSGA-II, SPEA-II, MOSA, and MOEA/D. In this work, Non-dominated Sorting Genetic Algorithm II (NSGA-II) [10], as a sub-category of MOEA, is applied in order to discover the closest solution to the Pareto-optimal solution. It brings in diversity and preserves the best solution of the current population in the next generation. Our recommendation problem can be formulated as [31]:

$$\begin{cases} max & F(L) = (f_1(L), f_2(L), f_3(L), f_4(L))^T \\ s.t. & L \in \Omega \end{cases} \tag{1}$$

where $f_t(L)$ is the tth objective function and L is a recommendation list. In MOO, there exists a (possibly infinite) number of Pareto optimal solutions. A solution is called Pareto efficient, if none of the objective functions can be improved without degrading at least one of the other objectives. For two recommendation lists $L_i, L_j \in \Omega$, it is said that L_i dominates L_j (denoted as $L_i < L_j$) iff

$$\begin{cases} & \forall a = 1, 2, 3, 4 f_a(L_i) \geq f_a(L_j) \\ \wedge & \exists b = 1, 2, 3, 4 f_b(L_i) > f_b(L_j) \end{cases}$$

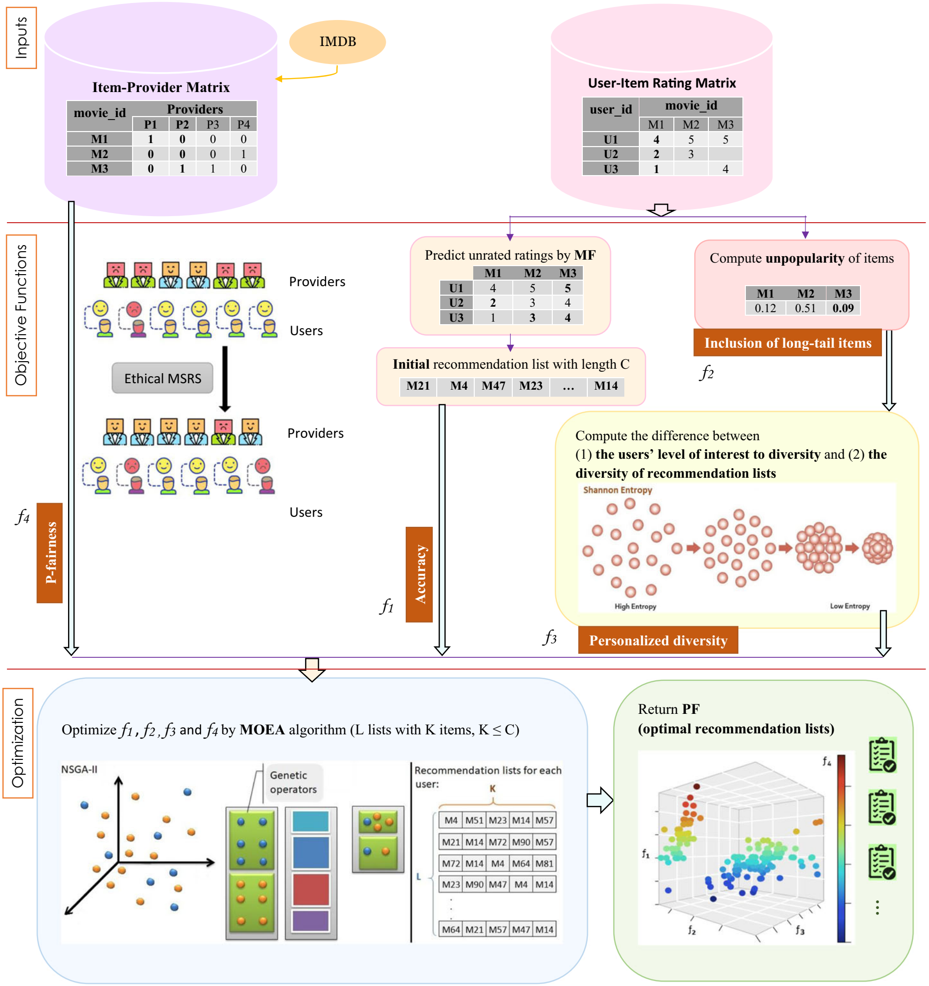
A decision vector $L^* \in \Omega$ is called a Pareto-optimal solution if there is no other solution that dominates it. The set of all the Pareto-optimal solutions is called the Pareto-optimal set (P^*), defined as

$$P^* = \{L \in \Omega | \neg \exists L^* \in \Omega, L > L^*\} \tag{2}$$

The values of the objective function under the objective space corresponding to the feasible solutions in Pareto-optimal set are called the Pareto Front (PF), written as

$$PF = \{F(L^*) = (f_1(L^*), f_2(L^*), f_3(L^*), f_4(L^*))^T | L^* \in P^*\} \tag{3}$$

MOO optimizes the four contradictory objective functions simultaneously to find a set of Pareto-optimal solutions, approximating the true PF, for each user.



Item-Provider Matrix

movie_id	Providers			
	P1	P2	P3	P4
M1	**1**	**0**	0	0
M2	**0**	**0**	0	1
M3	**0**	**1**	1	0

User-Item Rating Matrix

user_id	movie_id		
	M1	M2	M3
U1	**4**	5	5
U2	**2**	3	
U3	**1**		4

Predict unrated ratings by **MF**

	M1	M2	M3
U1	4	5	**5**
U2	**2**	3	4
U3	1	**3**	**4**

Compute **unpopularity** of items

M1	M2	M3
0.12	0.51	**0.09**

Initial recommendation list with length C

M21	M4	M47	M23	…	M14

Compute the difference between (1) **the users' level of interest to diversity** and (2) **the diversity of recommendation lists**

Optimize f_1, f_2, f_3 and f_4 by **MOEA** algorithm (L lists with K items, K ≤ C)

Recommendation lists for each user:

M4	M51	M23	M14	M57
M21	M14	M72	M90	M57
M72	M14	M4	M64	M81
M23	M90	M47	M4	M14
.				
M64	M21	M57	M47	M14

Return **PF**
(optimal recommendation lists)

Figure 3 Main framework of the proposed multi-objective MSRS solution (called *MOMSRS*); step 1. preparing the user-item and item-provider matrices as the inputs; step 2. formulating the four objective functions which are (1) accuracy, (2) inclusion of long-tail items, (3) personalized diversity, and (4) P-fairness; step 3. optimizing the contradictory objective functions to return PF as the output for each user

In this work, the proposed multi-objective MSRS (called MOMSRS) involves three main phases.

- First, in our work, we apply Matrix factorization (MF) [21] to predict the ratings of unrated items using the embeddings of users and items (refer to Algorithm 1). To implement MF, we use user embedding and item embedding matrices and Gradient Descent to get the optimal decomposition. We also incorporate user and item bias terms into the dot product to improve the performance of MF model. Next, we build an initial recommendation of top-C items for each user based on the predicted ratings. The initial recommendation list for user ($user_i$) is denoted as $InitialRecList_i$. Then, a set of recommendation lists with the length of K are generated for user ($user_i$) using the items just within $InitialRecList_i$.
- Second, we apply NSGA-II, as a multi-objective optimization tool, to meet the requirements of multi-stakeholder recommender systems in finding PF (set of top-K recommendation lists) for each user. The objective functions are (1) accuracy (f_1), (2) inclusion of long-tail items (f_2), (3) personalized diversity (f_3), and (4) P-fairness (f_4).
- Finally, we select and recommend some of those lists, which are trading-off among objective functions for $user_i$, called PF_i. The recommendation lists in PF_i consist of more long tail items in a personalized manner and more coverage of providers while the accuracy has almost no change. This procedure is going to be repeated for all users of the system.

In the following subsections, the three main steps of the proposed multi-objective MSRS (called MOMSRS) solution are described in details, as shown in Figure 3 and Algorithm 2.

Algorithm 1 MF using embedding.

```
Inputs: $N$, $M$, and $k$.
procedure RATING PREDICTION
    # embedding layers:
    $emb\_u = Embedding(N, k)$
    $emb\_m = Embedding(M, k)$
    # get a user and an item as inputs:
    $u = Input()$ # user
    $m = Input()$ # item
    # pass $u$ and $m$ through embeddings:
    $eu = emb\_u(u)$ #returns a k-size vector
    $em = emb\_m(m)$ #returns a k-size vector
    # dot them to get prediction without bias:
    $r = dot(eu, em)$
    # add user bias and item bias:
    $bias\_u = Embedding(N, 1)$
    $bias\_m = Embedding(M, 1)$
    # pass $u$ and $m$ through bias embeddings:
    $bu = bias\_u(u)$
    $bm = bias\_m(m)$
    # add bias to prediction:
    $r = add(r, bu, bm)$
return $r$ as the rating prediction for user $u$ on item $m$.
```

Algorithm 2 Our proposed method.

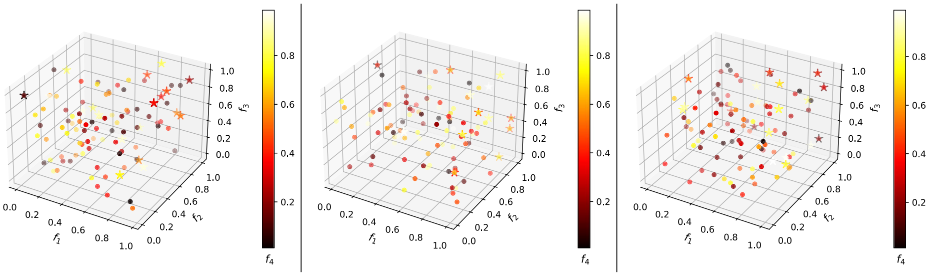
```
Input: $N \times M$ $User\text{-}item$ matrix; $M \times N_p$ $item\text{-}provider$ binary matrix; $N \times C$ $initialRecList$ to hold top-$C$ items based on MF algorithm for all users; number of lists $L$.
procedure $PF(user_i)$
    for each row in $initialRecList$ do
        for #NP do
            # randomly generate $L$ lists with length $K$ from $InitialRecList_i$,
            # fitness calculation (Section 3.2): $\begin{cases} max & f_1 = \sum_{i=1}^{K} \widetilde{r}_{u,i} \\ max & f_2 = \sum_{i=1}^{K} \frac{1}{\mu_i(\sigma_i+1)^2} \\ max & f_3 = -|S_x(u) - div(i_1, i_2, \cdots, i_k)| \\ max & f_4 = \frac{|P_u|}{N_p} \end{cases}$
            # divide the initial population into different PF based on non-domination,
            # rank lists within each PF using crowding distance (refer to Algorithm 3,
            # select based on PF ranking and crowding distance,
            for #gens do
                # crossover (Section 3.3.1 and Figure 6):
                two parent chromosomes $P_1$ and $P_2$ are chosen randomly,
                $Off_1, Off_2 = crossover(P_1, P_2)$
                # eliminate duplicate
                # mutation (Section 3.3.2 and Figure 6):
                $Off_1, Off_2 = mutation(Off_1, Off_2)$
            End for
        End for
    End for
    # $L$ recommendation lists for each user is generated
    # top-$l$ lists ($l \leq L$) are chosen to be recommended to the target user)
return $PF$ for all users
```

3.1 Inputs

The inputs are user-item rating matrix (R) and item-provider binary matrix (P). User-item rating matrix holds the ratings of users U on items I (e.g. $r_{u,i}$ denotes the rating of user u on item i, where $u \in U$ and $i \in I$). The ratings are used to express how users like items. Item-provider binary matrix shows which item belongs to which provider. For example, $P_{i,j} = 1$ means that movie i belongs to provider j, and $P_{i,j} = 0$ means that movie i does not belong to provider j.

3.2 Objective functions

As mentioned, in this study four objective functions are considered for selecting items to be included in the recommendation list. We apply NSGA-II as a multi-objective optimization algorithm to optimize the recommendation lists. The objectives are (1) accuracy, (2) inclusion of long-tail items, (3) personalized diversity, and (4) P-fairness. Our proposed multi-objective optimization solution using NSGA-II has the following steps for the target user ($user_i$).

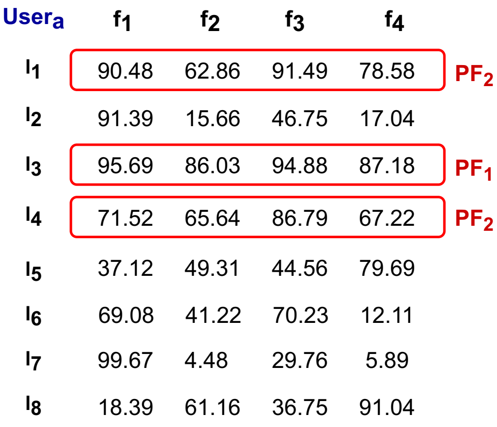


(a) (b) (c)

Figure 4 Pareto Front (PF) of MOMSRS on three random users for MovieLens 1M dataset. PF is indicated by star points and shows a set of recommendation lists with trading-off among the objective functions

- Population initialization: we randomly generate L recommendation lists with length K ($K \leq C$) from $InitialRecList_i$, each list consists of the top-C items for the target user. These L recommendation lists are the initial population.
- Non-dominated sort: by using fast non-dominated sorting, the initial population for each user is divided into different PF as $PF_1, PF_2, \cdots, PF_t$. As an example, Figure 4 shows the PF for three random users considering the objective functions (f_1, f_2, f_3, f_4) in MovieLens 1M dataset.
- Crowding distance: once the non-dominated sort is completed, the crowding distance is assigned. We rank the recommendation lists within each PF using crowding distance in descending order. Algorithm 3 shows the crowding distance calculation.
- Selection: after sorting the recommendation lists based on non-dominated sort and assigning the crowding distance, the selection is carried out using a crowded tournament selection. The crowded tournament selection is based on ranking and distance. The recommendation list with a better PF rank will be selected. If two lists have the same PF rank, the list with bigger crowding distance value will be selected.
- Genetic operators: once selection is done, the offspring population is created from this new population using the modified crossover and mutation operators. More details are provided in Section 3.3. It continues till the population size of new generation exceeds the current population size. The procedure repeats to build next generations.

Figure 5 An example to show the selection of the top-3 recommendation lists ($[l_3, l_1, l_4]$) from seven generated offspring lists for a specific user ($user_a$) based on the proposed method



$User_a$	f_1	f_2	f_3	f_4	
l_1	90.48	62.86	91.49	78.58	PF_2
l_2	91.39	15.66	46.75	17.04	
l_3	95.69	86.03	94.88	87.18	PF_1
l_4	71.52	65.64	86.79	67.22	PF_2
l_5	37.12	49.31	44.56	79.69	
l_6	69.08	41.22	70.23	12.11	
l_7	99.67	4.48	29.76	5.89	
l_8	18.39	61.16	36.75	91.04	

The whole process repeats NP times for each user to obtain an optimum result in the search space. Lastly, top-l recommendation lists containing items with trading-off among all four objective functions are recommended to the target user. Figure 5 is a toy example to show the top-3 recommendation lists to a specific user ($user_a$). As what can be seen, [l_3, l_1, l_4] are the top-3 lists which are in PF_1 and PF_2. Pareto fronts PF_1 and PF_2 are determined based on (3). Note that l_1 and l_4 are in the same PF, so they are ranked based on the crowding distance. The traditional accuracy-focused RSs only consider the accuracy objective function (f_1) and generate [l_7, l_3, l_2] as the top-3 recommendation lists for the target user. According to Figure 5, l_7 and l_2 may have high accuracy but the other objectives are largely ignored. Similarly, a recommender system that only focuses on provider fairness objective (f_4) recommends [l_8, l_3, l_5] to the target user. Again, l_8 and l_5 may have high provider coverage but the other objectives are overlooked. This example shows that the proposed method provides recommendation lists by considering a trade-off among all the objective functions which can make multiple stakeholders satisfied.

Algorithm 3 Crowding distance.

```
Inputs: $PF_i$ is the PF for user $i$ and $N_l$ is the number of recommendation lists in $PF_i$.
procedure $I(d_k)$
    for each $PF_i$ do
        The distance for all the recommendation lists are initialized to be zero i.e. $PF_i(d_j) = 0$, where $j$ is the $j$th recommendation list in $PF_i$.
        for each objective function $m$ do
            Sort the lists in $PF_i$ based on objective $m$ i.e. $I = sort(PF_i, m)$
            Assign infinite distance to boundaries in $PF_i$ i.e. $I(d_1) = \infty$ and $I(d_{N_l}) = \infty$
            for $k = 2$ to $(N_l - 1)$ do
                $I(d_k) = I(d_k) + \frac{I(k+1).m - I(k-1).m}{f_m^{max} - f_m^{min}}$
                # $I(k).m$ is the value of the $m$th objective function of the $k$th list in $I$.
            End for
        End for
    End for
return $I(d_k)$ as distances for the items in each $PF_i$.
```

In the following, the four objective functions are explained in details.

Objective I The accuracy of recommendation is based on the ratings of top-K items predicted by the above-mentioned CF approach. Thus, the first objective function to measure accuracy is:

$$f_1 = \sum_{i=1}^{K} \widetilde{r}_{u,i} \tag{4}$$

where $\widetilde{r}_{u,i}$ is the prediction of user u on item i, and K is the length of list. The larger the value is, the more accurate the items in the list are.

Objective II We also want to provide more long-tail items in recommendation lists. Considering long-tail items based on the number of ratings is not used in this work because: (1) popular items receive a large number of ratings whereas long-tail items receive few ratings which makes it hard to design a normalized objective function; (2) many items have the same number of ratings which makes it hard to differentiate them. According to [16], popular items are with low variance. Therefore, it is better to sum unpopularity of items in a list based on the mean and the variance of the ratings [16]. The second objective function to measure unpopularity of recommendation is:

$$f_2 = \sum_{i=1}^{K} \frac{1}{\mu_i(\sigma_i + 1)^2} \tag{5}$$

where μ_i and σ_i are the mean and the variance of ratings of item i rated by all users. The larger the value is, the more diverse and long-tail items the list has.

Objective III To further improve our method, we aim to increase the inclusion of long-tail items in recommendation by considering the users' level of interests to diversity. To do so, we compute both the user's level of interest to diversity of recommendations and the diversity of recommendation list, separately. Then, we calculate the difference between user interest to diversity and diversity of recommendation list. The low value of the difference shows that the more personalized the diversity of recommendations is. In the following, we explain the personalization process of the diversity in recommendations with more details. Part (a) measures the user's level of interest to diversity of recommendations, part (b) calculates the diversity of recommendation list, and part (c) measures the difference between user interest to diversity and diversity of recommendation list.

a. We employ the Shannon's entropy [35] to estimate the user's level of interests to diversity of the recommendations. The entropy of user u for feature x is computed as follows:

$$S_x(u) = -\sum_{i=1}^{k} q_i \cdot log_k q_i \tag{6}$$

where k is the number of possible values for the feature x and q_i is the ratio of the number of ratings given by user u to the items where the x of which has the value i, to the total number of user's ratings (refer to (7)).

$$q_i = |r_{u,x}|/|r_u|\Big|_{x=i} \tag{7}$$

b. There are two ways to calculate the diversity of a recommendation list, which are overall diversity and feature-based diversity. To measure the user's level of interest to diversity, it is more precise to compute the feature-based diversity [15]. For instance, a user may prefer more diversity in a specific feature of movies such as genre, director, actor, story, and so on. In this work, the feature-based diversity is used as follows:

$$div(i_1, i_2, \cdots, i_k) = \frac{\sum_{a=1}^{M} \sum_{b=1}^{M} (Sim(i_a, i_b))}{\binom{k}{2}} \tag{8}$$

where k shows the number of movies in the recommendation list, $i_1, i_2, \cdots, i_k$ are the recommended movies, and $Sim(i_a, i_b)$ is the feature-based similarity between two items i_a and i_b. The similarity is calculated based on the idea in [18], as shown in (9). For a feature-based similarity with n types, each item has a binary vector ($T_m = (t_{m,1}, t_{m,2}, \cdots, t_{m,n})$), where $t_{m,g} = 1$ if item m has the feature type g and $t_{m,g} = 0$ if item m does not have the feature type g ($g = 1, \cdots, n$).

$$Sim(i_a, i_b) = \frac{O_{11}}{O_{01} + O_{10} + O_{11}} \tag{9}$$

where O_{11} , O_{01} and O_{10} are the numbers of feature types when ($t_{i_a,g}$= 1 and $t_{i_b,g}$ = 1), ($t_{i_a,g}$ = 0 and $t_{i_b,g}$ = 1) and ($t_{i_a,g}$ = 1 and $t_{i_b,g}$ = 0) respectively.
c. We measure the difference between user's level of interest to diversity ($Sx(u)$) and diversity of recommendation list ($div(i_1, i_2, \cdots, i_k)$). The third objective function to measure personalized diversity is:

$$f_3 = |S_x(u) - div(i_1, i_2, \cdots, i_k)| \tag{10}$$

The lower the value of the difference is, the more personalized diversification the recommendations list has.

Objective IV The other objective is to provide P-fairness by covering more providers in the recommendation.

PF_u is a set of recommendation lists ($PF_u = L_{u,1}, L_{u,2}, \cdots, L_{u,l}$) for the target user u. The P-fairness is measured by the providers' coverage of the PF. The set of all items recommended to user u is:

$$I_u = \bigcup_{i=1}^{l} I_i \tag{11}$$

where I_i is a set with all elements in $L_{u,i}$. The set of all providers for items in I_u referred to as P_u. f_4 measures the provider coverage of recommended items to user u:

$$f_4 = \frac{|P_u|}{N_p} \tag{12}$$

where $|P_u|$ is the number of providers in P_u and N_p is the total number of providers in the system. The larger the value is, the more providers are covered.

As a simple example, assume that there are six items ($i_1, i_2, \cdots, i_6$) and four providers (P_1, P_2, P_3, P_4) in the system, and the IP matrix for this system is as follows:

$$\begin{array}{cccccc} & P_1 & P_2 & P_3 & P_4 & \\ \left(\begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\right. & \begin{array}{c} 1 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{array} & \begin{array}{c} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{array} & \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{array} & \left.\begin{array}{c} \\ \\ \\ \\ \\ \\ \end{array}\right) \begin{array}{c} i_1 \\ i_2 \\ i_3 \\ i_4 \\ i_5 \\ i_6 \end{array} \end{array}$$

where the 0 and 1 numbers are assigned following the relationship between each item and the providers (e.g. the provider of i_1 is P_1). As a recommendation, user u receives three lists of items as $PF_u = \{L_{u,1} = \{i_1, i_4, i_6\}, L_{u,2} = \{i_4, i_5, i_6\}, L_{u,3} = \{i_1, i_4, i_5\}\}$. So, I_u and P_u for user u are $I_u = \{i_1, i_4, i_5, i_6\}$ and $P_u = \{P_1, P_3, P_4\}$ respectively. Finally, $f_3 = \frac{|P_u|}{N_p} = \frac{3}{4} = 0.75$ means that the provider coverage for user u is %75.

These objective functions should be maximized simultaneously as follows. Note that in order to formulate the corresponding recommendation problem of this work as a maximum optimization problem, we revise the third objective function.

$$\begin{cases} max & f_1 = \sum_{i=1}^{K} \widetilde{r}_{u,i} \\ max & f_2 = \sum_{i=1}^{K} \frac{1}{\mu_i(\sigma_i+1)^2} \\ max & f_3 = -|S_x(u) - div(i_1, i_2, \cdots, i_k)| \\ max & f_4 = \frac{|P_u|}{N_p} \end{cases} \tag{13}$$

3.3 Optimization

In this section, we will explain the modified crossover and mutation operators as the optimization process of our method MOMSRS.

3.3.1 Crossover

A modified single-point crossover is applied in our method, as shown in Figure 6. According to [28], single-point crossover algorithm is a simple often-used method which shows better results than other crossover algorithms. The procedure is as follows: (1) two parent chromosomes P_1 and P_2 are chosen randomly in order to increase the diversity of population and discover a larger sets of items; (2) position i along the two of P_1 and P_2 recommendation lists is selected randomly ($1 < i < K$); (3) two new child lists are generated by swapping all the genes between $i + 1$ and K ($P_{1,v} \leftrightarrow P_{2,v}, \forall v \in \{i + 1 \leq v \leq K\}$). This exchange procedure may cause some duplicate items in the offspring lists, for instance the 2nd and 4th items of P_1 (item 11) in Figure 6. As each recommendation list must consist of exactly K items, a procedure is needed to eliminate and replace the duplicate. To do so, we improve the standard single-point crossover operation. The repeated items are substituted by items at the same position in the parent lists, which is different from the repeated item. This procedure is repeated till there are no duplicate left, as shown in Figure 6.

3.3.2 Mutation

Single-point mutation is implemented in our method (refer to Figure 6). Single-point mutation can be easily performed as it has a lower computation complexity rather than the others [38]. The item at the mutation point is substituted by a randomly-selected item from the initial recommendation list. To make sure that items in the recommendation list are different, the item should be selected from the items that do not belong to the parent.

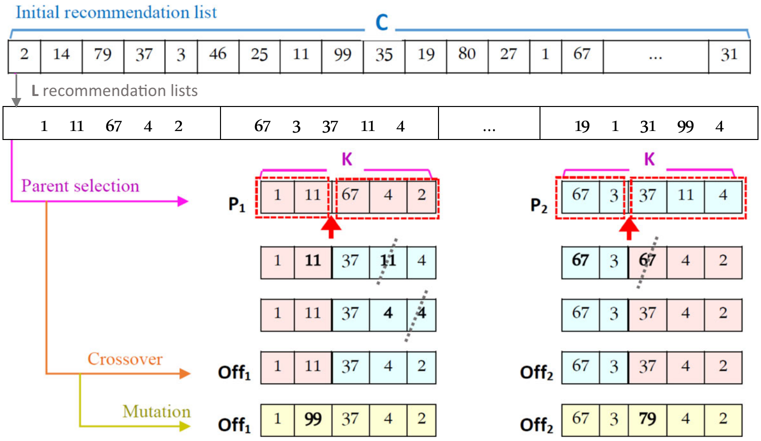


Figure 6 The crossover and mutation operations in our method; the red pointer shows the randomly-chosen position of single-point crossover algorithm. The duplicates are eliminated in our modified crossover operator

Table 1 The statistics of the experimental datasets

Dataset	MovieLens 100K	MovieLens 1M
Users	943	6040
Movies	1,682	3952
Ratings	100,000	1,000,209

4 Experimental result

In order to validate the proposed solution under different users' rating behavior, we conducted a set of experiments on two real datasets. These datasets are MovieLens 100K and MovieLens 1M.

MovieLens 100K dataset This dataset[3] is a well-known movie dataset that has been widely used for the evaluation of CF recommender systems. This dataset consists of 100,000 ratings from 943 users on 1682 movies. The data is provided by the University of Minnesota and are associated with their online movie-recommendation system. Each user has given scores to at least 20 movies on a 5-star scale.

MovieLens 1M dataset This dataset[4] is a bigger dataset from GroupLens which consists of 1,000,209 ratings from 6040 users on 3952 movies. The users have given scores to at least 20 movies on a 5-star scale.

These two datasets are summarized in Table 1. To evaluate our solution, we sampled 80% of each dataset for training, and the remaining 20% are used for the test with the 5-fold cross-validation method. We considered the movie companies as the providers and one of the stakeholders. We crawled the name of these companies from IMDb. We also considered genre of movies as the feature for feature-based diversity measurements.

4.1 Evaluation metrics

In order to evaluate the effectiveness of our solution and compare with the existing works, we use the following metrics.

Mean Absolute Error (MAE) We use the MAE metric to measure the recommendation accuracy of the recommendation lists for all users:

$$MAE = \frac{1}{N}\sum_{u=1}^{N}\sum_{i \in PF_u} |\widetilde{r}_{PF_u} - r_{PF_u}| \tag{14}$$

where $\widetilde{r}_{PF_u}$ and r_{PF_u} are the predicted and the actual ratings of PF_u respectively, and N is the total number of users. Note that $PF_u = \{L_{u,1}, L_{u,2}, \cdots, L_{u,l}\}$ where $L_{u,i}$ is the ith list with top-K items for user u. The smaller the MAE value is, the more accurate the recommendation is.

[3] https://grouplens.org/datasets/movielens/100k/

[4] https://grouplens.org/datasets/movielens/1m/

Diversity of recommendation **a.** Novelty: This metric [2] measures how much exposure the unpopular items have been included in the recommendations.

$$Novelty = \frac{1}{N}\frac{1}{|LT|}\sum_{u=1}^{N}\sum_{i \in PF_u} \mathbb{1}(i \in LT) \tag{15}$$

where LT is the list of the items in long-tail in the system, $|LT|$ is the total number of long-tail items, and $\mathbb{1}(i \in L)$ is an indicator function that returns 1 when i is in LT and 0 otherwise. Note that, the unpopularity value of item i is calculated based on $\frac{1}{\mu_i(\sigma_i+1)^2}$. The larger the $Novelty$ value is, more items in the long tail (novel items) the recommendation lists have covered.

b. Attribute-based Diversity: we measure the diversity of the recommendation list based on attribute "genre" of movies as follows [15]:

$$Div_g(i_1, ..., i_k) = \frac{1}{N}\sum_{i \in PF_u} \frac{\sum_{i=1}^{k}\sum_{j=i}^{k}(1 - Sim(i_a, i_b))}{\frac{k}{2} * (k-1)} \tag{16}$$

where k is the number of items in the list and $(i_1, i_2, \cdots, i_k)$ are the recommended items. $Sim(i_a, i_b)$ indicates the similarity between two items i_a and i_b based on their genre attribute (see more details in [18] for the genre-based similarity calculation). The larger the Div_g value is, the more diverse the recommendation is.

Coverage of providers We introduce this metric to measure the provider coverage in PF.

$$P\_Cov = \frac{1}{N} * \frac{1}{N_p}\sum_{u=1}^{N}\sum_{i \in PF_u} \mathbb{1}(i \in uncov\_P_{PF_u}) \tag{17}$$

where $\mathbb{1}(i \in uncov\_P_{PF_u})$ is an indicator function and it equals to 1 when i belongs to an uncovered provider in PF_u and 0 otherwise, and N_p is the total number of providers in the system. The larger the P_Cov value is, the more providers the recommendation lists have covered.

Table 2 Parameter setting of our method

Parameter	Meaning	Value
C	length of initial recommendation list	50
K	length of recommendation list	5
$epoch$	number of iterations for MF algorithm	15
reg	regularization penalty for MF algorithm	0.05
d	latent dimensionality for MF algorithm	10
L	number of recommendation lists for initial population	50
l	number of top-l recommendation lists to be recommended	5
NP	size of population	100
$gens$	number of generations	30
n	number of neighbors	10
pc	crossover probability	0.9
pm	mutation probability	0.1

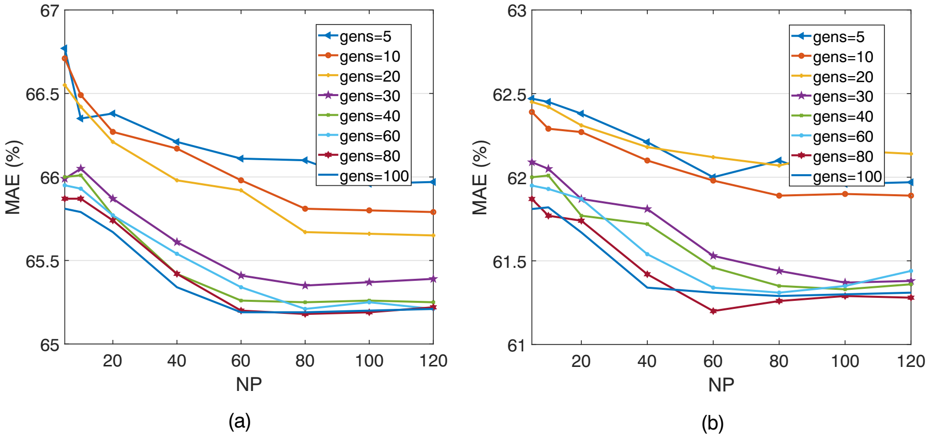


Figure 7 Selection of *gens* and *NP* for datasets **a** MovieLens 100K and **b** MovieLens 1M

4.2 Experimental settings

There are several parameters used in our work, as displayed in Table 2. All the algorithms are coded in Python, and the experiments have been run on an Intel(R) Core(TM) i5 machine with 2.60 GHz CPU and 16.0 G memory. We do not consider the effect of changing *epoch*, *reg*, *d*, and *n* here, as MF results is not our concern in this work. *NP* and *gens* are important to accelerate premature convergence. According to Figure 7, *gens* and *NP* are set as 30 and 80 respectively, because choosing large values causes an increase in computational time without a significant decrease in MAE. *pc* and *pm* are also selected by test and trial. *n* is set as ten because choosing a small size of neighborhood causes lack of ability to explore new search space, while choosing a large one causes an increase in computational time.

4.3 Baseline methods

We carry out experiments against datasets ML100K and ML1M and compare the results of our method *MOMSRS* with existing baselines. The comparison between our method and the baselines is presented in Table 3.

Table 3 Comparison between our *MOMSRS* method and five baseline methods

Method	Rs/MSRS	Objectives				MOEA
		ObjI	ObjII	ObjIII	objIV	
CF-RS	RS	✓				✗
re-ranking-MSRS	MSRS	✓			✓	✗
MOEA/D-RS	RS	✓	✓			MOEA/D
NSGA-II-RS	RS	✓	✓			NSGA-II
NSGA-II-MSRS	MSRS	✓	✓		✓	NSGA-II
MOMSRS*	MSRS	✓	✓	✓	✓	NSGA-II

ObjI, ObjII, ObjIII, and objIV denote *accuracy*, *Long-tail inclusion*, *personalized diversity*, and *P-fairness* objectives, respectively

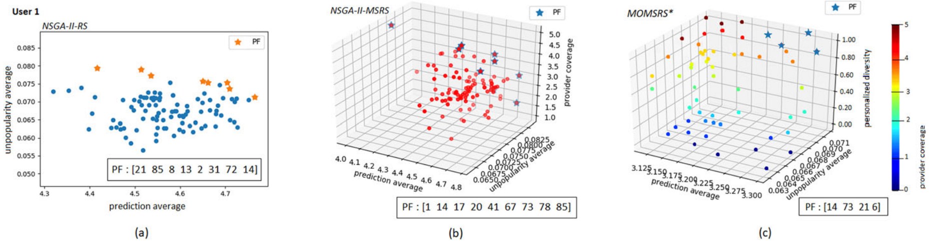


Figure 8 The PF of **a** *NSGA-II-RS*, and **b** *NSGA-II-MSRS*, and **c** our *MOMSRS* method, on a specific user of ML100K data; the star points represent the recommendation lists in the PF

1. *CF-RS* [21] is a single-objective CF method with MF. This method is an accuracy-focused recommendation method.
2. *re-ranking-MSRS* [7] is a re-ranking approach which considers the provider coverage.
3. *MOEA/D-RS* [38] is a two-objectives recommendation method, and solutions are implemented by MOEA/D. Accuracy and diversity are taken as the objective functions at the same time.
4. *NSGA-II-RS* [33] is a two-objectives recommendation method, and solutions are implemented by NSGA-II. Accuracy and long tail inclusion are taken as the objective functions simultaneously.
5. *NSGA-II-MSRS* [20] is a three-objectives recommendation method, and solutions are implemented by NSGA-II. Accuracy, long tail inclusion, and provider fairness are taken as the objective functions at the same time.

4.4 Experimental results and discussion

Multi-objective recommender system suggests multiple recommendation lists to each user. As an example, Figure 8 shows the PF in *NSGA-II-RS*, *NSGA-II-MSRS*, and *MOMSRS* for a specific user in ML100K dataset, respectively. Figure 8a shows the relation between the unpopularity average and the prediction average with *NSGA-II-RS*. A higher prediction average in a list reflects a higher accuracy, and the higher unpopularity average reflects a higher level of long-tail inclusion (diversity) in the recommendation. Figure 8b shows the relation among the unpopularity average, the prediction average, and the provider coverage with *NSGA-II-MSRS*. Figure 8c shows the relation among the unpopularity average, the prediction average, the personalized diversity, and the provider coverage (P-fairness) with *MOMSRS*. Personalized diversity and P-fairness are objectives which have not been considered along with accuracy and long-tail inclusion in existing work. *MOMSRS* recommends four lists, *NSGA-II-RS* suggests eight lists, and *NSGA-II-MSRS* recommends nine lists to this user ($user_1$). Each recommendation lists that *MOMSRS* recommends is the trading-off when considering all the four objective functions.

4.4.1 Performance analysis considering P-fairness (RQ1)

Table 4 and Figure 9 show MAE, $Novelty$, Div_g, and P_Cov against ML100K and ML1M datasets. In our method, a set of recommendation lists as PF is generated in one run for each target user. Based on the concept of PF, we will not say which list in PF is better than the others. Thus, we compute the mean values of the evaluation metrics. The fourth objective function of the proposed method aims to have a fair manner for providers by giving

Table 4 MAE, $Novelty$, Div_g, and P_{-Cov} of our method comparing with baselines against ML100K and ML1M datasets

Dataset	Metric	Method					
		CF-RS	re-ranking-MSRS	MOEA/D-RS	NSGA-II-RS	NSGA-II-MSRS	MOMSRS*
ML100K	MAE	64.77	69.42	66.01	64.54	65.35	64.81
	$Novelty$	23.45	36.12	42.13	49.41	48.14	53.53
	Div_g	19.23	25.43	33.82	49.31	49.29	53.16
	P_{-Cov}	40.66	46.06	42.11	46.93	59.68	62.87
ML1M	MAE	59.46	63.21	61.78	60.18	61.44	60.54
	$Novelty$	35.7	44.04	55.85	58.86	57.19	62.02
	Div_g	28.47	39.21	48.95	61.29	61.35	63.11
	P_{-Cov}	38.54	45.92	44.92	46.39	61.12	64.10

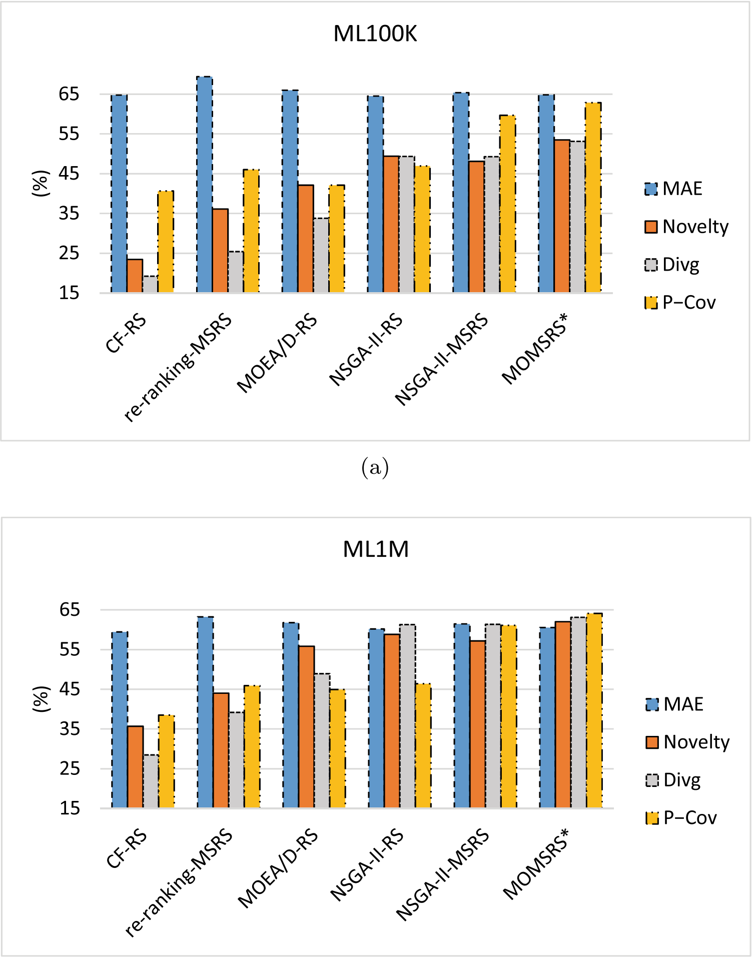
a balanced chance to their items to be explored. For question RQ1, we analyze the effects of P-fairness on performance. Considering P-fairness, our *MOMSRS* method achieves 15.94% and 17.71% higher values for P_Cov, 3.85% and 1.82% higher values for Div_g, and 4.12% and 3.16% higher values for $Novelty$, with only 0.27% and 0.36% loss of accuracy in comparison with *NSGA-II-RS* for ML100K and ML1M datasets respectively. Our *MOMSRS* method covers more providers with a minor gain in MAE. Beyond the baseline methods, our method has covered a larger number of providers so that it increases the utilities and satisfaction of more providers. On Table 4, it can be seen that 61.46%, 54.08%, 55.08%, and 53.61% of providers have not been covered using *CF-RS*, *re-ranking-MSRS*, *MOEA/D-RS*, and *NSGA-II-RS* methods respectively (with ML1M dataset). our *MOMSRS* method reduces this percentage to 37.13% for ML100K dataset and 35.9% for ML1M dataset. This a significant fairness improvement for providers.

4.4.2 Performance analysis considering personalized diversification (RQ2)

We demonstrate our *MOMSRS* method's superiority using personalized diversification over the baseline methods. The contribution of our method over the main baseline *NSGA-II-MSRS* is considering the users' level of interest with diversity. Comparing with *NSGA-II-MSRS*, the P_Cov of our method achieves 3.19% and 2.98% higher for ML100K and ML1M datasets respectively, with 0.54% and 0.9% gain of accuracy (refer to Table 4 and Figure 9). In addition, it can be seen that our method improves the Novelty and diversity. These results show that our method covers more providers with even higher value in accuracy. The proposed method has covered a more relevant diverse and novel items with the use of personalized diversity. This means that recommending diverse items pleasantly surprises users if long-tail items are brought to right users.

4.4.3 Performance analysis using multi-objective optimization (RQ3)

For question RQ3, we compare the performance of our *MOMSRS* method with the baseline *re-ranking-MSRS*. The *re-ranking-MSRS* method re-ranked the accuracy-focused recommendation list using a coverage-oriented algorithm at the provider level while ignoring the importance of long-tail items, novelty and diversity in recommendation. The *re-ranking-MSRS* method could not engage a set of solutions at the same time with respect to multiple



(a)

(b)

Figure 9 The results of MAE, $Novelty$, and P_{-Cov} of our method comparing with baselines against **a** ML100K, and **b** ML1M datasets

objectives. To address these issues, our *MOMSRS* method employs a multi-objective evolutionary algorithm. According to Table 4 and Figure 9, the P_Cov value of our method is 16.81% and 18.18% higher than that in the *re-ranking-MSRS* method against ML100K and ML1M datasets respectively. The $Novelty$ value of our method for ML100K and ML1M datasets is 17.41% and 17.98% larger respectively than those in the *re-ranking-MSRS* method. The Div_g value of our method for ML100K and ML1M datasets is 27.73% and 23.9% larger respectively than those in the *re-ranking-MSRS* method. The accuracy in our *MOMSRS* method is 4.61% and 2.67% higher than *re-ranking-MSRS* method against ML100K and ML1M datasets respectively.

In addition, comparing with traditional accuracy-focused *CF-RS* method, our *MOMSRS* method enhances the novelty (30.08% in ML100K and 26.32% in ML1M), diversity (33.93% in ML100K and 34.64% in ML1M), and provider coverage (22.21% in ML100K

Table 5 Pairwise t-Test results of our *MOMSRS* method against baselines

Dataset	Metric	Methods				
		A-B	A-C	A-D	A-E	A-F
ML100K	MAE	5.3E-3	4.6E-3	3.4E-3	1.9E-3	2.3E-3
	$Novelty$	1.6E-2	1.8E-2	1.9E-2	1.8E-2	1.9E-2
	Div_g	1.4E-2	1.5E-2	1.7E-2	1.8E-2	1.7E-2
	P_{-Cov}	2.1E-2	2.3E-2	2.3E-2	2.4E-2	2.0E-3
ML1M	MAE	6.8E-3	4.5E-3	3.9E-3	4.1E-3	3.4E-2
	$Novelty$	1.0E-2	1.4E-2	2.1E-2	2.1E-2	2.1E-2
	Div_g	1.1E-2	1.2E-2	1.4E-2	1.6E-2	1.5E-2
	P_{-Cov}	1.9E-2	1.9E-2	2.2E-2	2.2E-2	1.8E-2

A, B, C, D, E, and F denote *MOMSRS**, *NSGA-II-MSRS*, *NSGA-II-RS*, *MOEA/D-RS*, *re-ranking-MSRS*, and *CF-RS* respectively

and 25.56% in ML1M) with a small sacrifice in the accuracy (0.04% in ML100K and 1.08% in ML1M). The experimental results show that the accuracy of some recommendation lists in MOEA algorithms is even higher than the list recommended by CF (refer to the 5th list in the example shown in Figure 2).

4.4.4 P-value results of our *MOMSRS* method with baselines (RQ4)

Lastly, to answer RQ4, we provide the pairwise t-test with a confident level $\alpha = 0.05$ in Table 5, where each value shows the p-value for a t-test between the proposed method and each baseline, to compare the performance of our *MOMSRS* method with baselines. A p-value less than α indicates that the difference is statistically significant. The results in Table 5 show that *NSGA-II-MSRS* outperforms these baseline methods for both datasets.

5 Conclusion

In this work, we propose a fairness-aware multi-stakeholder recommendation approach based on MOEA. We specify four objective functions for the recommendation accuracy, long tail inclusion, personalized diversity, and P-fairness. We develop the algorithm to find the optimal solutions with a trade-off among the objective functions. In particular, we propose a personalized diversification method to consider the interest level of the user in long-tail recommendation and develop a P-fairness algorithm to calculate the exposure distribution of providers. The output of our method is a set of Pareto-optimal solutions for each user. Experimental results show that the proposed method is effective to provide more long-tail items in a personalized manner and better fairness for providers with a small loss of the recommendation accuracy.

Consider the strong capacity of graph neural networks in applications [27, 37], in the future work, we are also going to investigate the possibility of employing graph neural networks in multi-stakeholder recommender systems.

References

1. Abbassi, Z., Amer-Yahia, S., Lakshmanan, L.V., Vassilvitskii, S., Yu, C.: Getting recommender systems to think outside the box. In: Proceedings of the Third ACM Conference on Recommender Systems, pp. 285–288 (2009)
2. Abdollahpouri, H.: Incorporating system-level objectives into recommender systems. In: Companion Proceedings of the 2019 World Wide Web Conference, pp. 2–6. ACM (2019)
3. Abdollahpouri, H., Adomavicius, G., Burke, R., Guy, I., Jannach, D., Kamishima, T., Krasnodebski, J., Pizzato, L.: Beyond personalization: research directions in multistakeholder recommendation. arXiv:1905.01986 (2019)
4. Adomavicius, G., Tuzhilin, A.: Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans Knowl Data Eng **17**(6), 734–749 (2005)
5. Anderson, C.: The Long Tail: why the Future of Business is Selling Less of More. Hachette Books, New York (2006). https://en.wikipedia.org/wiki/Hachette_Books
6. Burke, R., Sonboli, N., Mansoury, M., Ordoñez-gauger, A.: Balanced neighborhoods for fairness-aware collaborative recommendation (2017)
7. Burke, R.D., Abdollahpouri, H., Mobasher, B., Gupta, T.: Towards multi-stakeholder utility evaluation of recommender systems. In: UMAP (Extended Proceedings) (2016)
8. Chen, L., Wu, W., He, L.: Personality and recommendation diversity. In: Emotions and Personality in Personalized Services, pp. 201–225. Springer (2016)
9. Cremonesi, P., Koren, Y., Turrin, R.: Performance of recommender algorithms on top-n recommendation tasks. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 39–46. ACM (2010)
10. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: Nsga-ii. IEEE Transactions on Evolutionary Computation **6**(2), 182–197 (2002)
11. Di Noia, T., Ostuni, V.C., Rosati, J., Tomeo, P., Di Sciascio, E.: An analysis of users' propensity toward diversity in recommendations. In: Proceedings of the 8th ACM Conference on Recommender systems, pp. 285–288. ACM (2014)
12. Domingues, M.A., Gouyon, F., Jorge, A.M., Leal, J.P., Vinagre, J., Lemos, L., Sordo, M.: Combining usage and content in an online music recommendation system for music in the long-tail. In: Proceedings of the 21st International Conference on World Wide Web, pp. 925–930. ACM (2012)
13. Edizel, B., Bonchi, F., Hajian, S., Panisson, A., Tassa, T.: Fairecsys: mitigating algorithmic bias in recommender systems. Int. J. Data Sci. Anal. **9**(2), 197–213 (2020)
14. Garcia-Soriano, D., Bonchi, F.: Maxmin-fair ranking: individual fairness under group-fairness constraints. arXiv:2106.08652 (2021)
15. Hamedani, E.M., Kaedi, M.: Recommending the long tail items through personalized diversification. Knowl. Based Syst. **164**, 348–357 (2019)
16. Jambor, T., Wang, J.: Optimizing multiple objectives in collaborative filtering. In: Proceedings of the Fourth ACM Conference on Recommender Systems, pp. 55–62. ACM (2010)
17. Kaminskas, M., Bridge, D.: Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. ACM Transactions on Interactive Intelligent Systems (TiiS) **7**(1), 2 (2017)
18. Kermany, N.R., Alizadeh, S.H.: A hybrid multi-criteria recommender system using ontology and neuro-fuzzy techniques. Electron. Commer. Res. Appl. **21**, 50–64 (2017)
19. Kermany, N.R., Zhao, W., Yang, J., Wu, J.: Reincre: enhancing collaborative filtering recommendations by incorporating user rating credibility. In: International Conference on Web Information Systems Engineering, pp. 64–72. Springer (2020)
20. Kermany, N.R., Zhao, W., Yang, J., Wu, J., Pizzato, L.: An ethical multi-stakeholder recommender system based on evolutionary multi-objective optimization. In: 2020 IEEE International Conference on Services Computing (SCC), pp. 478–480. IEEE (2020)
21. Koren, Y.: The bellkor solution to the netflix grand prize. Netflix Prize Documentation **81**(2009), 1–10 (2009)
22. Kunaver, M., Požrl, T.: Diversity in recommender systems–a survey. Knowl. Based Syst. **123**, 154–162 (2017)
23. Leonhardt, J., Anand, A., Khosla, M.: User fairness in recommender systems. In: Companion Proceedings of the The Web Conference 2018, pp. 101–102 (2018)
24. Li, Y., Chen, H., Fu, Z., Ge, Y., Zhang, Y.: User-oriented fairness in recommendation. In: Proceedings of the Web Conference 2021, pp. 624–632 (2021)
25. Liu, W., Burke, R.: Personalizing fairness-aware re-ranking. arXiv:1809.02921 (2018)

26. Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: a survey. Decis. Support. Syst. **74**, 12–32 (2015)
27. Ma, X., Wu, J., Xue, S., Yang, J., Sheng, Q.Z., Xiong, H.: A comprehensive survey on graph anomaly detection with deep learning. arXiv:2106.07178 (2021)
28. Magalhaes-Mendes, J.: A comparative study of crossover operators for genetic algorithms to solve the job shop scheduling problem. WSEAS Trans. Comput. **12**(4), 164–173 (2013)
29. Mansoury, M.: Fairness-aware recommendation in multi-sided platforms. In: Proceedings of the 14th ACM International Conference on Web Search and Data Mining, pp. 1117–1118 (2021)
30. Mehrotra, R., McInerney, J., Bouchard, H., Lalmas, M., Diaz, F.: Towards a fair marketplace: counterfactual evaluation of the trade-off between relevance, fairness & satisfaction in recommendation systems. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 2243–2251 (2018)
31. Miettinen, K.: Nonlinear Multiobjective Optimization, vol. 12. Springer Science & Business Media, Berlin (2012)
32. Modani, N., Jain, D., Soni, U., Gupta, G.K., Agarwal, P.: Fairness aware recommendations on behance. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 144–155. Springer (2017)
33. Pang, J., Guo, J., Zhang, W.: Using multi-objective optimization to solve the long tail problem in recommender system. In: Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 302–313. Springer (2019)
34. Patro, G.K., Biswas, A., Ganguly, N., Gummadi, K.P., Chakraborty, A.: Fairrec: two-sided fairness for personalized recommendations in two-sided platforms. In: Proceedings of The Web Conference 2020, pp. 1194–1204 (2020)
35. Shannon, C.E.: A mathematical theory of communication. ACM SIGMOBILE Mobile Computing and Communications Review **5**(1), 3–55 (2001)
36. Shi, Y., Zhao, X., Wang, J., Larson, M., Hanjalic, A.: Adaptive diversification of recommendation results via latent factor portfolio. In: Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 175–184. ACM (2012)
37. Su, X., Xue, S., Liu, F., Wu, J., Yang, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Jin, D., et al.: A comprehensive survey on community detection with deep learning. arXiv:2105.12584 (2021)
38. Wang, S., Gong, M., Li, H., Yang, J.: Multi-objective optimization for long tail recommendation. Knowl. Based Syst. **104**, 145–155 (2016)
39. Xu, G., Zhang, Y., Yi, X.: Modelling User Behaviour for Web Recommendation Using Lda Model. In: 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, vol. 3, pp. 529–532. IEEE (2008)
40. Yao, W., He, J., Huang, G., Zhang, Y.: Modeling dual role preferences for trust-aware recommendation. In: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, pp. 975–978 (2014)
41. Yin, H., Cui, B., Li, J., Yao, J., Chen, C.: Challenging the long tail recommendation. Proceedings of the VLDB Endowment **5**(9), 896–907 (2012)
42. Zhu, Z., Hu, X., Caverlee, J.: Fairness-aware tensor-based recommendation. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, pp. 1153–1162. ACM (2018)
43. Zuo, Y., Gong, M., Zeng, J., Ma, L., Jiao, L.: Personalized recommendation based on evolutionary multiobjective optimization [research frontier]. IEEE Comput. Intell. Mag. **10**(1), 52–62 (2015)