An Evolutionary Multitasking Algorithm for Efficient Multiobjective Recommendation

Ye Tian, Yiwei Hu, Haiping Ma, Le Wu, and Xingyi Zhang *Senior Member, IEEE*

Abstract—Represented by evolutionary algorithms and swarm intelligence algorithms, nature-inspired metaheuristics have been successfully applied to recommender systems and amply demonstrated effectiveness in particular for multiobjective recommendation. Owing to the population based search paradigm, these algorithms can produce a number of recommendation lists making diverse trade-offs between multiple metrics, meeting the requirements of accuracy, novelty, diversity, and other user preferences. However, these algorithms are criticized for the low efficiency of the optimization process, especially when the number of users is large. To address this issue, this paper proposes an evolutionary multitasking based recommendation method, where each task corresponds to a user and all the tasks are optimized simultaneously, thus highly improving the efficiency of recommendation. In order to enhance the effectiveness, all the users are divided into multiple populations according to the similarity between their preferences, where each population is evolved with internal knowledge transfer between users, and all the populations are evolved with external knowledge transfer between populations. Experimental results on various datasets verify that the proposed method can better balance between multiple metrics than classical and deep neural network based recommendation methods, and exhibits significantly higher efficiency than evolutionary multiobjective optimization based recommendation methods.

Index Terms—Recommender systems, evolutionary multiobjective optimization, evolutionary multitasking, knowledge transfer

1 Introduction

In today's era of big data, users are inundated with massive information resources in network activities [1]. To alleviate the overload of information, recommender systems have been developed to filter out the most suitable items according to user and item profiles [2]. Based on audiences' ratings on movies, recommender systems can suggest suitable unrated movies in streaming services [3]. According to customers' click records and shopping lists, recommender systems can provide the most relevant goods on e-commerce websites [4]. By predicting the return ratios according to stocks' hierarchical correlation, recommender systems can identify profitable stocks for investors in fintech scenarios [5]. Moreover, recommender systems have covered various types of items, such as driving routes [6], academic articles [7], scenic spots [8], and print services [9].

The techniques used in conventional recommender systems mainly include content based methods, memory based collaborative filtering methods, model based collaborative filtering methods, and many others. The content based methods take a user's experiences as reference content, and suggest items whose content is similar to the reference [10]. The memory based collaborative filtering methods evaluate the interest of a user in a candidate item, on the basis of the similarity between the user and other users preferring the item [11], or the similarity between the item and other items preferred by the user [12]. The model based collaborative filtering methods predict users' ratings on candidate items by training mathematical models, among which the matrix factorization is widely used [13]. There also exist many other techniques employed by recommender systems, such as the demographic filtering methods suggesting items based on personal demographic features [14], the knowledge based methods inferring matches between users and items based on knowledge information [15], and the utility based methods sorting items based on utility functions [16], among many others [17]. While a single recommendation method may suffer from overspecialization, sparseness, cold-start, long-tail, or other issues [18], [19], hybrid recommendation methods are put forward to inherit the merits of different techniques and remedy their shortcomings by each other, such as the ensemble of content based and collaborative filtering methods [20], collaborative filtering and demographic filtering methods [21], and demographic filtering, knowledge based, and utility based methods [22]. Furthermore, the vigorous development of computational intelligence has promoted a new round of recommendation technology revolution, such as the deep learning models estimating the ranking scores of items [23] or directly generating recommendation lists [24], the evolutionary computation techniques tuning the hyperparameters in recommendation models [25] or mining frequent patterns [26], and the fuzzy set theory assisting the inference of matches between users and items [27] or similarity between items [28].

While most existing methods are relentless in the pursuit of high recommendation accuracy [29], it has been recognized that the recommendation lists solely based on accuracy are sometimes not the most suitable ones and may

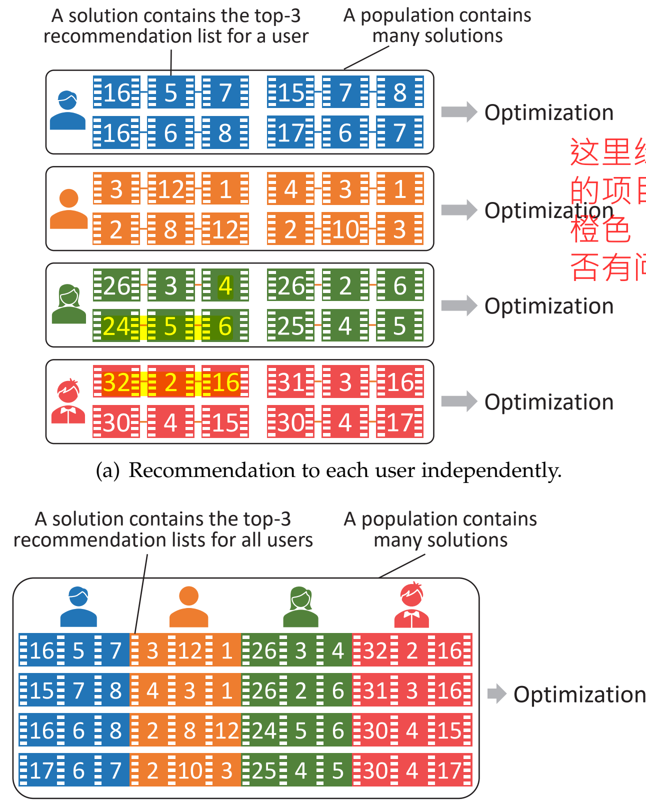
- *Y. Tian, Y. Hu, and H. Ma are with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, Institutes of Physical Science and Information Technology, Anhui University, Hefei 230601, China. Email: field910921@gmail.com, 523797773@qq.com, hpma@ahu.edu.cn.*
- *L. Wu is with the Key Laboratory of Knowledge Engineering with Big Data, Hefei University of Technology, Hefei 230029, China. Email: lewu.ustc@gmail.com.*
- *X. Zhang is with the Key Laboratory of Intelligent Computing and Signal Processing of Ministry of Education, School of Artificial Intelligence, Anhui University, Hefei 230601, China. Email: xyzhanghust@gmail.com.*

Manuscript received –.

even degrade user experience [30], [31]. In addition to accuracy, audiences would like to watch some movies covering a broad range of genres, and websites hope unpopular movies could be paid much attention. Thus, other metrics like diversity [32], novelty [33], fairness [34], and interpretability [35] have been suggested in practice. Besides, some recommender systems consider user preferences in different aspects of items [36] or produce the same recommendation lists for many users [37], in which multiple metrics are also involved. The emerging demand for multiobjective recommendation gave birth to a number of multiobjective methods in recent years [38]. On the one hand, some methods implicitly consider multiple metrics in the recommendation, such as the personalized diversity promoting generative adversarial network considering accuracy and diversity [24], the self-supervised graph learning considering accuracy and robustness [39], and the dominant and frequent itemset mining algorithm considering support and occupancy [40]. On the other hand, some methods explicitly aggregate multiple metrics in the recommendation, such as the linear modular dispersion hybrid algorithm aggregating accuracy and diversity [41], the greedy algorithm aggregating social welfare and fairness [37], and the graph theoretic approach aggregating compatibility and versatility [42].

Due to the conflicting nature of the above metrics, it is not easy to strike a balance between them for multiobjective recommendation [43], [44]. More seriously, the scalarization methods can only obtain extreme recommendation lists (i.e., some metrics are the best but the others are the worst) when the Pareto front is nonconvex [45]. To steer clear of the aggregation of metrics, multiobjective evolutionary algorithms (MOEAs) have been employed to produce a set of recommendation lists making diverse trade-offs between metrics [38]. Moreover, MOEAs search for suitable recommendation lists in a black-box manner, which means that any new metrics can be added without modifying the recommendation method itself [46]. However, the introduction of flexibility is at the expense of efficiency, where MOEAs are criticized for the time-consuming population based optimization process. As illustrated in Fig. 1, some MOEA based recommendation methods search for top-K items for each user independently [43], [47], which are inefficient for handling a large number of users in sequence. By contrast, some other methods search for top-K items for all users simultaneously [48], [49], which encounter the curse of dimensionality [50] and are ineffective in finding high-quality recommendation lists in such a huge search space.

In recent years, evolutionary multitasking has appeared as a novel optimization paradigm, which optimizes multiple tasks simultaneously and transfers useful genetic material between tasks [51]. Due to the similarity between the preferences of many users, it is highly desirable to regard each user as a task and optimize all tasks via evolutionary multitasking, where recommended items can be transferred between tasks to facilitate the optimization process. That is, evolutionary multitasking simultaneously searches for the recommendation lists for all users as shown in Fig. 1(b), but it only encodes the recommendation list for a single user in each solution as shown in Fig. 1(a), thus holding a good balance between efficiency and effectiveness. Nevertheless, evolutionary multitasking has not been applied to recommender systems so far, since existing algorithms focus on handling only several tasks [51], [52], [53] and are ineffective in finding the recommendation lists for a large number of users. To address this dilemma, this paper proposes a clustering based Evolutionary Multitasking algorithm for Multiobjective Recommendation (EMMR), which not only provides an effective and efficient recommendation method, but also extends the application scope of evolutionary multitasking. It should be noted that the proposed method is different from the concepts of group recommendation or multitask recommendation, since group recommendation suggests the same items to multiple users [37] while the proposed method suggests different items to users, and multitask recommendation handles multiple recommendation tasks (e.g., rating prediction and top-K recommendation) simultaneously [54] while the proposed method handles a single recommendation task (i.e., top-K recommendation).



(a) Recommendation to each user independently.

(b) Recommendation to all users simultaneously.

Fig. 1. Two types of MOEA based recommendation methods with different encoding schemes.

这里绿色和红色的项目中间线是橙色，不知道是否有问题

Overall, the contributions of this work are summarized as follows.

1) This work formulates a top-K recommendation task as a multiobjective multitasking optimization problem, where the recommendation to each user is regarded as a task and all the tasks are handled simultaneously. By solving this problem using evolutionary multitasking, the superiority of MOEAs in balancing between multiple conflicting metrics is retained, and the inefficiency of MOEAs in handling a large number of users can be addressed.
2) This work serves as the first attempt to customize

an evolutionary multitasking algorithm for recommender systems. To enable the algorithm to handle a large number of tasks simultaneously, the users are clustered into multiple groups, where each user is associated with several solutions and the solutions of all users in the same group constitute a population. By performing internal knowledge transfer inside each population and external knowledge transfer between populations, the solutions of all users are evolved cooperatively, and thus the convergence speed can be improved considerably.

3) This work compares the performance of the proposed method with state-of-the-art recommendation methods on three datasets. Experimental results demonstrate that the proposed method is superior over classical methods, deep neural network based methods, and MOEA based methods. Furthermore, ablation studies verify the effectiveness of the core components in the proposed method.

This paper is organized as follows. Section 2 introduces the basic concepts of evolutionary multiobjective optimization and evolutionary multitasking, as well as existing MOEA based recommendation methods and evolutionary multitasking algorithms. Section 3 describes the proposed method in detail. Section 4 presents and analyzes the experimental comparisons between the proposed method and the state-of-the-art. Lastly, Section 5 concludes this paper.

2 Related Work and Motivation

This section first gives an introduction to multiobjective optimization and MOEA based recommendation methods, then reviews evolutionary multitasking algorithms, and lastly presents the motivation of this work.

2.1 Multiobjective Evolutionary Algorithm Based Recommendation Methods

A maximization multiobjective optimization problem can be defined as

$$\begin{array}{ll}\text{Maximize} & \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})) \\ \text{Subject to} & \mathbf{x} \in \Omega\end{array}, \tag{1}$$

where $\mathbf{x}$ denotes a solution within the search space Ω and $\mathbf{F}$ contains m objective functions. Instead of aggregating all the objectives by different weights, MOEAs aim to make diverse trade-offs between the objectives based on Pareto dominance relation. Formally, a solution $\mathbf{x}$ is better than (i.e., dominates) a solution $\mathbf{y}$ if and only if

$$\begin{cases} f_i(\mathbf{x}) \geq f_i(\mathbf{y}), & \forall i = 1, 2, \ldots, m \\ f_j(\mathbf{x}) > f_j(\mathbf{y}), & \exists j = 1, 2, \ldots, m \end{cases}, \tag{2}$$

which is denoted as $\mathbf{x} \succ \mathbf{y}$. The goal of MOEAs is to find a set of Pareto optimal solutions S for the problem, where $\mathbf{x}$ does not dominate $\mathbf{y}$ for all $\mathbf{x} \in \Omega$ and $\mathbf{y} \in S$. By comparing solutions according to their dominance relations, MOEAs do not aggregate the objectives or use any weight, and an MOEA can be used to handle any number of objectives. Moreover, MOEAs are able to obtain a set of uniformly distributed solutions on highly irregular Pareto fronts, which are very suitable for solving complex problems [55].

Owing to the high flexibility of MOEAs, they have been applied to handle multiple conflicting objectives in various areas [50]. In terms of recommender systems, MOEAs have also been employed to search for recommendation lists by optimizing multiple metrics. For instance, a non-dominated sorting genetic algorithm [48], a non-dominated neighbor immune algorithm [56], and a probabilistic MOEA [49] were suggested to optimize accuracy and diversity. In [43], a decomposition based MOEA was adopted to optimize accuracy and novelty. In [57], a sparse MOEA was proposed to optimize support, occupancy, and utility. In [46], a reference vector guided evolutionary algorithm was adopted to optimize accuracy, diversity, novelty, and coverage. In [58], four MOEAs equipped with knowledge graph were employed to optimize precision, diversity, and explainability.

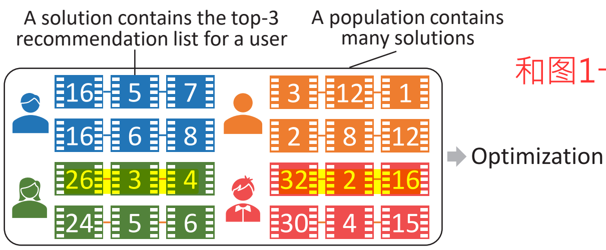
In terms of the encoding schemes (i.e., the meaning of each decision variable in a solution) of the above methods, some of them search for top-K items by using binary encoding [56], [57], some search for top-K items by using integer encoding [43], some search for top-K items for multiple users by using integer encoding [48], [49], and some others optimize the parameters related to recommendation models [46], [58]. As illustrated in Fig. 1, these encoding schemes encounter difficulties in balancing between the efficiency and effectiveness of recommendation, where the methods encoding the recommendation list for a single user are inefficient to handle many users, the methods encoding the recommendation lists for many users are ineffective to find high-quality solutions, and the methods encoding the parameters of recommendation models are unable to explicitly balance between multiple conflicting metrics. Therefore, this work proposes an evolutionary multitasking based recommendation method, which encodes the recommendation list for a single user but can search for the recommendation lists for many users simultaneously, thus ensuring both the efficiency and effectiveness of recommendation. In the next subsection, some basic concepts of evolutionary multitasking are presented.

2.2 Evolutionary Multitasking

A maximization multitasking optimization problem can be defined as

$$\begin{array}{ll}\text{Maximize} & \mathbf{T}(\mathbf{x}_1, \ldots, \mathbf{x}_n) = (\mathbf{F}_1(\mathbf{x}_1), \ldots, \mathbf{F}_n(\mathbf{x}_n)) \\ \text{Subject to} & \mathbf{x}_i \in \Omega_i, \;\; i = 1, \ldots, n\end{array}, \tag{3}$$

where $\mathbf{x}_i$ denotes a solution within the search space Ω_i for the i-th task, and each task $\mathbf{F}_i$ denotes an optimization problem with single or multiple objective functions. The difference between multiobjective optimization and multitasking optimization is that the former considers the trade-off between objectives, and each solution should be evaluated on all objectives. By contrast, the latter only searches for the optimal solutions for each task, and each solution is only evaluated on a single task. Moreover, the solutions for a multiobjective optimization problem share a unique search space while the solutions for different tasks may locate in different search spaces. In order to exploit latent complementarities between essentially separate tasks, a multifactorial evolutionary algorithm was proposed to leverage upon the implicit parallelism of population based



和图1一样

Fig. 2. The proposed evolutionary multitasking based recommendation method with a hybrid encoding scheme.

search, which assigns each solution to a task by adapting its skill factors and decodes it into a task-specific representation [51]. In other words, the algorithm uses a unified encoding scheme to encode each solution $\mathbf{x}$, which can be further mapped into any search space Ω_i for task $\mathbf{F}_i$.

Evolutionary multitasking refers to an optimization paradigm rather than a specific optimizer, and a number of evolutionary multitasking algorithms have been proposed to solve single-objective or multiobjective multitasking optimization problems in recent years [52], [59], [60]. For multiobjective multitasking optimization problems, diverse trade-offs between the objectives of each task can be made while no trade-off between tasks is made, which meets the requirements of many real-world scenarios. Taking a recommender system as an example, it aims to balance between the accuracy, diversity, novelty, and many other metrics in the recommendation list for each user, and provides different recommendation lists for different users for the sake of personalization. As a consequence, evolutionary multitasking has been applied to various applications such as cloud service composition [61], feature selection [62], vehicle routing [63], and scheduling [64], and this work proposes an evolutionary multitasking algorithm for recommender systems for the first time.

2.3 Motivation of This Work

The motivation of applying evolutionary multitasking to multiobjective recommendation is not the simple parallelization of the recommendation to many users (i.e., tasks), but the convergence acceleration achieved by transferring useful genetic material between users. More importantly, the recommendation lists for different users are with the same encoding and in the same search space (i.e., $\Omega_1 = \cdots = \Omega_n$), hence the decoding procedure used in the multifactorial evolutionary algorithm can be omitted. As illuminated in Fig. 2, different solutions in the proposed method may stand for the recommendation lists for different users, and the transfer of useful genetic material (i.e., items) can be naturally achieved by the crossover between the solutions for different users.

The knowledge transfer between users is based on the hypothesis that some of them have similar preferences, which is obvious for a recommendation task containing a large number of users. However, there also exist many users with quite different preferences, the knowledge transfer between which may even hinder the convergence. Hence, existing evolutionary multitasking algorithms mostly focus on handling several tasks to alleviate this issue [51]. To tackle a large number of users, this work adopts a user clustering strategy according to the similarity between users' preferences, so as to promote positive transfer while avoiding negative transfer. The users in the same group are associated with a population, in which each solution stands for the recommendation list for a user. With the assistance of an internal knowledge transfer strategy within each population and an external knowledge transfer strategy between populations, the solutions of all users can be cooperatively evolved with fast convergence speed. After the optimization process terminates, a unique set of solutions (i.e., recommendation lists) with diverse trade-offs between multiple metrics can be provided for each user. In the next section, the procedure of the proposed evolutionary multitasking based recommendation method is elaborated.

3 The Proposed Method

This section first presents the definition of the multiobjective multitasking recommendation problem solved in this work, then details the procedure of the proposed evolutionary multitasking based recommendation method.

3.1 Problem Definition

Given a dataset containing a list of users' ratings on items, which constitutes a user-item interaction matrix R with many elements being unknown, the goal of the recommendation task considered in this paper is to find K unrated items for each user, so as to satisfy multiple requirements quantified by metrics. Formally, the optimization problem can be defined as

$$\begin{aligned} \text{Maximize} \quad & \mathbf{T}(\mathbf{x}_1,\dots,\mathbf{x}_n) = (\mathbf{F}(\mathbf{x}_1,1),\dots,\mathbf{F}(\mathbf{x}_n,n)) \\ & \mathbf{F}(\mathbf{x},u) = (f_1(\mathbf{x},u),\dots,f_m(\mathbf{x},u)),\ u=1,\dots,n \\ \text{Subject to} \quad & \mathbf{x} \in \{1,\dots,l\} \\ & |\mathbf{x}| = K \end{aligned}, \tag{4}$$

where n is the number of users, m is the number of metrics, l is the number of items, solution $\mathbf{x}$ denotes a recommendation list, and objective $f_j(\mathbf{x},u)$ denotes the j-th metric value of recommendation list $\mathbf{x}$ with respect to the u-th user. It is worth noting that no delicate metric is designed in this work, as this work focuses on developing an evolutionary multitasking based recommendation method that is flexible to handle any number of metrics. Here, three representative metrics are chosen to instantiate the proposed method, including accuracy, novelty, and diversity.

Accuracy is certainly the most important metric for a recommendation list, which is generally reflected by a user's ratings on the recommended items. However, since the recommended items must be unrated by the user, the ratings have to be estimated by recommendation models [48], [49] or utility functions [9], [56]. Here the ratings of all users on all unrated items are estimated by recommendation models for collaborative filtering [65], [66], and the first metric f_1 is defined as

$$f_1(\mathbf{x},u) = \frac{1}{|\mathbf{x}|}\sum_{i\in\mathbf{x}} \hat{R}_{ui}, \tag{5}$$

where $\hat{R}_{ui}$ denotes the rating of the u-th user on the i-th item estimated by a recommendation model, and a higher $\hat{R}_{ui}$ indicates a higher preference.

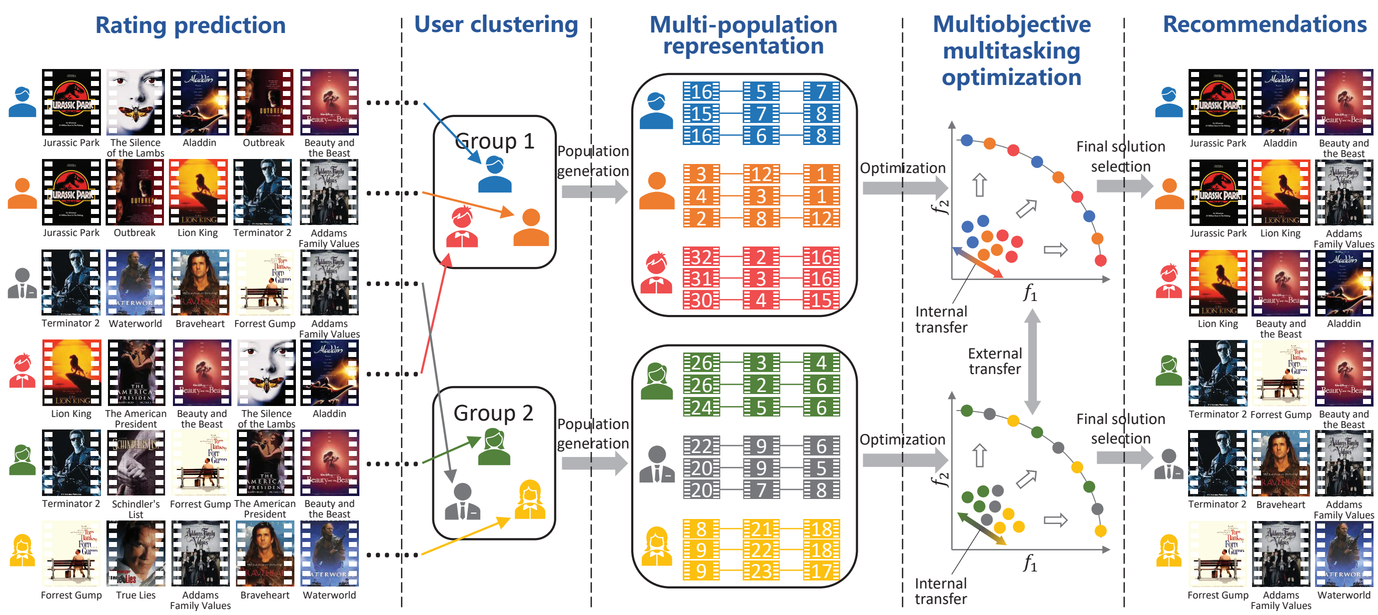


Fig. 3. Procedure of the proposed evolutionary multitasking algorithm for multiobjective recommendation (EMMR).

Novelty reflects the ratio of niche items in the recommendation list, where niche items refer to those preferred by a few users [33]. Hence, novelty is included to better cover the preferences of the minority, and the second metric f_2 is defined as

$$f_2(\mathbf{x}, u) = 1 - \frac{1}{|\mathbf{x}|}\frac{1}{n}\sum_{i\in\mathbf{x}} popular_i, \tag{6}$$

where $popular_i$ denotes the popularity of the i-th item, which is counted by the number of users rating the item in the user-item interaction matrix R.

Diversity measures the ability of recommending dissimilar items to a user, where the similarity between items can be measured by utility functions [56], [58] or additional properties [24], [49]. Since the datasets involved in the experiments of this paper contain category labels of items, the number of unique category labels in a recommendation list is considered, and the third metric f_3 is defined as

$$f_3(\mathbf{x}, u) = \frac{|\bigcup_{i\in\mathbf{x}} label_i|}{|label_{all}|}, \tag{7}$$

where $label_i$ denotes the category labels of the i-th item and $label_{all}$ denotes all the unique category labels in the dataset.

3.2 Solution Representation

The proposed method represents each solution $\mathbf{x} = (x_1, \ldots, x_K)$ with a vector of integers, where each decision variable denotes the index of an item. Besides, an additional scalar is attached to each solution to indicate the user index that it is associated with. As shown in Fig. 2, each solution denotes a recommendation list for a user, and all the solutions for the users in the same group constitute a population. In addition, the items encoded in a solution for a user should be unrated by the user, which is ensured in the generation of solutions.

Algorithm 1: Procedure of EMMR

Input: R (user-item interaction matrix), c (number of groups), s (number of solutions for each user)
Output: P_{final} (set of recommendation lists for each user)

1 $\hat{R} \leftarrow$ Estimate the ratings of all users on all the unrated items based on R;
2 $P \leftarrow Initialization(\hat{R}, c, s)$; `//Algorithm 5`
3 **while** *terminal condition is not fulfilled* **do**
4   **for** *each population* $P_i \in P$ **do**
5         $P'_i \leftarrow MatingSelection(P_i, s)$; `//Algorithm 2`
6         $O \leftarrow GeneticOperators(P'_i, \hat{R})$; `//Algorithm 6`
7         $P_i \leftarrow EnvironmentalSelection(P_i \cup O, s)$; `//Algorithm 3`
8     $P \leftarrow ExternalTransfer(P, \hat{R})$; `//Algorithm 7`
9 $P_{final} \leftarrow FinalSelection(P)$; `//Algorithm 4`
10 **return** P_{final};

3.3 Procedure of the Proposed EMMR

The procedure of the proposed EMMR is illustrated in Fig. 3 and Algorithm 1. To begin with, the ratings of all the unrated items in the user-item interaction matrix are estimated by using a recommendation model (Line 1), which are then used to divide the users into c groups (Line 2). By initializing s solutions for each user, the solutions for all the users in the same group constitute a population and all the c populations are evolved separately. At each generation of the evolution of each population P_i, s repeatable parents are selected from the solutions for each user (Line 5), and s offspring solutions are then generated for each user with internal knowledge transfer (Line 6). Afterwards, the offspring solutions are combined with the original population, and s solutions for each user survive to the next generation (Line 7). At the end of each generation, the solutions in each population may migrate to another population as external knowledge transfer (Line 8). After the optimization process terminates, a solution for each user is picked up from the populations

Algorithm 2: $MatingSelection(P_i, s)$

Input: P_i (a population), s (number of parents for each user)
Output: P_i' (a set of parents)
$P_i' \leftarrow \emptyset$;
for *each user u in population P_i* **do**
  $P_u \leftarrow$ All solutions in P_i for user u;
  for $t = 1, \dots, s$ **do**
    $[\mathbf{x}, \mathbf{y}] \leftarrow$ Randomly select two solutions from P_u;
    if $\mathbf{x}$ *has a smaller non-dominated front number than* $\mathbf{y}$ **then**
      $P_i' \leftarrow P_i' \cup \{\mathbf{x}\}$;
    else if $\mathbf{y}$ *has a smaller non-dominated front number than* $\mathbf{x}$ **then**
      $P_i' \leftarrow P_i' \cup \{\mathbf{y}\}$;
    else if $\mathbf{x}$ *has a larger crowding distance than* $\mathbf{y}$ **then**
      $P_i' \leftarrow P_i' \cup \{\mathbf{x}\}$;
    else
      $P_i' \leftarrow P_i' \cup \{\mathbf{y}\}$;
return P_i';

Algorithm 3: $EnvironmentalSelection(P_i, s)$

Input: P_i (a combined population), s (number of solutions for each user)
Output: P_{next} (truncated population for next generation)
$P_{next} \leftarrow \emptyset$;
for *each user u in population P_i* **do**
  $P_u \leftarrow$ All solutions in P_i for user u;
  $[Front_1, Front_2, \dots] \leftarrow$ Do non-dominated sorting on P_u;
  $Crowd \leftarrow$ Calculate the crowding distances of solutions in P_u;
  $k \leftarrow \mathrm{argmin}_i\, |Front_1 \cup \dots \cup Front_i| \geq s$;
  $Front_k \leftarrow$ Delete $|Front_1 \cup \dots \cup Front_k| - s$ solutions with the smallest crowding distances from $Front_k$;
  $P_{next} \leftarrow P_{next} \cup Front_1 \cup \dots \cup Front_k$;
return P_{next};

Algorithm 4: $FinalSelection(P)$

Input: P (a set of populations)
Output: P_{final} (a set of solutions for each user)
$P_{final} \leftarrow \emptyset$;
for *each population $P_i \in P$* **do**
  for *each user u in population P_i* **do**
    $P_u \leftarrow$ All solutions in P_i for user u;
    $\mathbf{x} \leftarrow$ Select the solution with the smallest angle to vector $(1, 1, \dots)$ in the objective space from P_u;
    $P_{final} \leftarrow P_{final} \cup \{\mathbf{x}\}$;
return P_{final};

Algorithm 5: $Initialization(\hat{R}, c, s)$

Input: $\hat{R}$ (user-item interaction matrix), c (number of groups), s (number of solutions for each user)
Output: P (a set of c populations)
1 $U \leftarrow$ Divide the users into c groups according to $\hat{R}$;
2 $P \leftarrow \emptyset$;
3 **for** *each user group $U_i \in U$* **do**
4   $P_i \leftarrow \emptyset$;
5   **for** *each user $u \in U_i$* **do**
6     $\mathbf{x} \leftarrow$ Create a solution by randomly selecting K items from 100 unrated items with the highest ratings estimated for user u;
7     $P_i \leftarrow P_i \cup \{\mathbf{x}\}$;
8   $P \leftarrow P \cup \{P_i\}$;
9 **return** P;

as the final recommendation list (Line 9).

Since this work focuses on finding high-quality recommendation lists, the solution initialization and offspring generation strategies (Lines 2, 6, 8) are the core contributions of the proposed EMMR, which will be elaborated in the subsequent subsections. By contrast, the selection strategies of the classical NSGA-II [67] are adopted to balance between multiple conflicting metrics (Lines 5, 7). More specifically, Algorithm 2 gives the procedure of mating selection, where s repeatable parents are selected via binary tournament selection according to the non-dominated front numbers and crowding distances of the solutions for each user. Algorithm 3 presents the procedure of environmental selection, where s solutions with the smallest non-dominated front numbers and largest crowding distances survive to the next generation for each user.

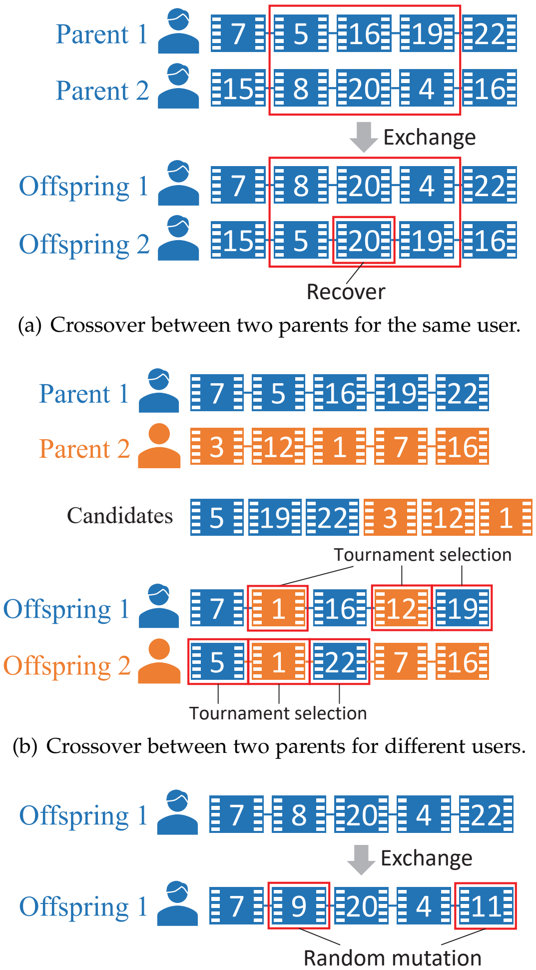
It is worth noting that the proposed EMMR outputs a set of recommendation lists rather than a single one for each user. These recommendation lists make diverse trade-offs between metrics, hence all of them can be retained and one is presented to the user according to user-specific options, e.g., some users may prefer popular items and some others may prefer niche items. Nevertheless, EMMR provides a function for picking up a single recommendation list from all the solutions for each user, so that it can be compared with existing methods recommending a single list more easily. As described in Algorithm 4, EMMR selects the solution with the smallest angle to vector $(1, 1, \dots)$ in the objective space from all the solutions for each user.

3.4 Population Initialization

Before the initialization of solutions, all the users are divided into c groups as presented in Line 1 of Algorithm 5. The clustering is carried out according to the user-item interaction matrix $\hat{R}$, by means of a hierarchical clustering method for large-scale datasets [68]. After that, each solution for each user is generated independently, which consists of K items randomly selected from 100 unrated items with the highest ratings estimated for the user (Line 6). In total, $n \times s$ solutions are initialized and assigned to c populations, where n is the number of users and s is the number of solutions for each user.

3.5 Genetic Operators and Internal Knowledge Transfer

The procedure of the proposed genetic operators with internal knowledge transfer is detailed in Algorithm 6, where two parents are randomly selected (Line 3) to generate two offspring solutions by using the proposed crossover operator (Lines 5–37), and then each offspring solution undergoes the proposed mutation operator (Lines 40–42). As plotted in Fig. 4(a), when the crossover is performed on two parents



(a) Crossover between two parents for the same user.

(b) Crossover between two parents for different users.

(c) Mutation of each offspring solution.

Fig. 4. Genetic operators suggested in the proposed EMMR.

associated with the same user, several successive items are randomly selected from the two parents and exchanged to generate two offspring solutions. If the exchange results in duplicated items in the offspring solutions, the exchanged items will be recovered. On the contrary, when the crossover is performed on two parents associated with different users, it is unreasonable to directly exchange their items since the preferences as well as the optimal items of the two users are different. In this case, internal knowledge transfer between the two users is carried out to generate one offspring solution for each user. In addition, since the mutation operator aims to enhance the exploration ability and escape from local optima, as shown in Fig. 4(c), each item in each offspring solution is replaced with a randomly selected one with a small probability mut_prob.

The procedure of internal knowledge transfer between two parents for different users is detailed in Fig. 4(b) and Algorithm 6. When generating the offspring solution $\mathbf{o}_1$ for the same user as parent $\mathbf{x}$, the items in both the two parents are first put into $\mathbf{o}_1$, and the unique items in each parent are put into a temporary set $\mathbf{d}$ (Line 11). Afterwards, the items in $\mathbf{d}$ are selected and put into $\mathbf{o}_1$ one by one, until $\mathbf{o}_1$ contains K items in total (Line 12). When selecting an item from $\mathbf{d}$, half the items are first selected from $\mathbf{d}$ via binary

Algorithm 6: $GeneticOperators(P'_i, \hat{R})$

Input: P'_i (a set of parents), $\hat{R}$ (user-item interaction matrix)
Output: O (offspring population)

1 $O \leftarrow \emptyset$;
2 **while** $P'_i \neq \emptyset$ **do**
3 $[\mathbf{x}, \mathbf{y}] \leftarrow$ Randomly select two solutions from P'_i;
4 $P'_i \leftarrow P'_i \setminus \{\mathbf{x}, \mathbf{y}\}$;
5 **if** $\mathbf{x}$ *and* $\mathbf{y}$ *are for the same user* **then**
6 $\mathbf{o}_1 \leftarrow \mathbf{x}$; $\mathbf{o}_2 \leftarrow \mathbf{y}$;
7 $[\varphi_1, \varphi_2] \leftarrow$ Randomly select two indexes from $\{1, \ldots, K\}$ with $\varphi_1 \leq \varphi_2$;
8 $[\mathbf{o}_1, \mathbf{o}_2] \leftarrow$ Exchange the items in $[\varphi_1, \varphi_2]$ of $\mathbf{o}_1, \mathbf{o}_2$;
9 $[\mathbf{o}_1, \mathbf{o}_2] \leftarrow$ Recover each exchanged item if it is duplicated in $\mathbf{o}_1$ or $\mathbf{o}_2$;
10 **else**
11 $\mathbf{o}_1 \leftarrow \mathbf{x} \cap \mathbf{y}$; $\mathbf{o}_2 \leftarrow \mathbf{o}_1$; $\mathbf{d} \leftarrow (\mathbf{x} \cup \mathbf{y}) \setminus \mathbf{o}_1$;
12 **while** $|\mathbf{o}_1| < K$ **do**
13 $\mathbf{p} \leftarrow \emptyset$;
14 **while** $|\mathbf{p}| < |\mathbf{d}|/2$ **do**
15 $[i, j] \leftarrow$ Randomly select two items unrated by the user of $\mathbf{x}$ from $\mathbf{d} \setminus \mathbf{p} \setminus \mathbf{o}_1$;
16 **if** *i has a higher rating estimated for the user of* $\mathbf{x}$ *than j* **then**
17 $\mathbf{p} \leftarrow \mathbf{p} \cup \{i\}$;
18 **else**
19 $\mathbf{p} \leftarrow \mathbf{p} \cup \{j\}$;
20 $[i, j] \leftarrow$ Randomly select two items from $\mathbf{p}$;
21 **if** *i has a higher rating estimated for the user of* $\mathbf{y}$ *than j* **then**
22 $\mathbf{o}_1 \leftarrow \mathbf{o}_1 \cup \{i\}$;
23 **else**
24 $\mathbf{o}_1 \leftarrow \mathbf{o}_1 \cup \{j\}$;
25 **while** $|\mathbf{o}_2| < K$ **do**
26 $\mathbf{q} \leftarrow \emptyset$;
27 **while** $|\mathbf{q}| < |\mathbf{d}|/2$ **do**
28 $[i, j] \leftarrow$ Randomly select two items unrated by the user of $\mathbf{y}$ from $\mathbf{d} \setminus \mathbf{q} \setminus \mathbf{o}_2$;
29 **if** *i has a higher rating estimated for the user of* $\mathbf{y}$ *than j* **then**
30 $\mathbf{q} \leftarrow \mathbf{q} \cup \{i\}$;
31 **else**
32 $\mathbf{q} \leftarrow \mathbf{q} \cup \{j\}$;
33 $[i, j] \leftarrow$ Randomly select two items from $\mathbf{q}$;
34 **if** *i has a higher rating estimated for the user of* $\mathbf{x}$ *than j* **then**
35 $\mathbf{o}_2 \leftarrow \mathbf{o}_2 \cup \{i\}$;
36 **else**
37 $\mathbf{o}_2 \leftarrow \mathbf{o}_2 \cup \{j\}$;
38 $O \leftarrow O \cup \{\mathbf{o}_1, \mathbf{o}_2\}$;
39 **for** *each offspring solution* $\mathbf{o} \in O$ **do**
40 **for** *each item* $i \in \mathbf{o}$ **do**
41 **if** $rand() < mut\_prob$ **then**
42 $\mathbf{o} \leftarrow$ Replace item i with a randomly selected item unrated by the user of $\mathbf{o}$;
43 **return** O;

tournament selection according to their ratings estimated for the same user as parent $\mathbf{x}$ (Lines 14–19), and one item is then

Algorithm 7: $ExternalTransfer(P, \hat{R})$

Input: P (a set of populations), $\hat{R}$ (user-item interaction matrix)
Output: P (updated populations)
1 **for** *each population* $P_i \in P$ **do**
2   **for** *each user* u *in population* P_i **do**
3     $P_u \leftarrow$ All solutions in P_i for user u;
4   **if** *more than half the solutions in* P_u *have not been changed for two successive generations* **then**
5       $v \leftarrow$ The user with the highest similarity to u excluding those in P_i;
6       $P_j \leftarrow$ The population which user v belongs to;
7       $P_i \leftarrow P_i \setminus P_u$;
8       $P_j \leftarrow P_j \cup P_u$;
9 **return** P;

identified and put into $\mathbf{o}_1$ via binary tournament selection according to their ratings estimated for the same user as the other parent $\mathbf{y}$ (Lines 20–24). Analogously, when putting an item into the offspring solution $\mathbf{o}_2$ for the same user as parent $\mathbf{y}$, half the items are first selected from $\mathbf{d}$ via binary tournament selection according to their ratings estimated for the same user as $\mathbf{y}$ (Lines 27–32), and one item is then identified via binary tournament selection according to their ratings estimated for the same user as $\mathbf{x}$ (Lines 33–37). This way, the crossover operator not only shares the items in both the two parents, but also considers the preferences of both the two users, thus achieving the acceleration of convergence speed.

3.6 External Knowledge Transfer

With the optimization process goes on, the solutions for a user may get stuck in local optima even with the internal knowledge transfer between the user and others in the same group. In this case, it is desirable to transfer the user to other groups for exploring promising items preferred by other users. Algorithm 7 details the procedure of external knowledge transfer, where all the solutions for a user u migrate to another population if more than half of them have not been changed for two successive generations (Line 4). To determine a suitable population, the similarity between u and each of the other user v is calculated by

$$sim(u, v) = \frac{|\bigcup_{\mathbf{x}\in P_u} \mathbf{x} \cap \bigcup_{\mathbf{y}\in P_v} \mathbf{y}|}{|\bigcup_{\mathbf{x}\in P_u} \mathbf{x} \cup \bigcup_{\mathbf{y}\in P_v} \mathbf{y}|}, \quad (8)$$

where P_u and P_v are the solution sets for users u and v, respectively. That is, the ratio of the same items recommended for two users is regarded as the similarity between them. Except for the users in the same group as u, the user v with the highest similarity to u is then determined (Line 5), and the solutions for u migrate to the population where v belongs to (Lines 6–8). As a consequence, the users are clustered according to the user-item interaction matrix before optimization while transferred according to the found items during optimization. Since the found items constitute quasi-optimal recommendation lists, the external knowledge transfer can move users to more suitable populations for better acceleration of convergence speed.

TABLE 1
Statistics of the experimental datasets.

Datasets	#Users	#Items	#Ratings	#Categories
Anime	25989	6773	810887	82
MovieLens-10m	40600	8625	1411225	20
Amazon-Music	15054	83728	143864	404

4 Experimental Studies

This section first presents the experimental settings, then compares the performance of the proposed method and state-of-the-art multiobjective recommendation methods, and lastly performs ablation studies to verify the effectiveness of the proposed internal knowledge transfer and external knowledge transfer strategies.

Amazon数据集中的名字叫Digital Music，我简写写成了Music，这里不知道是否有问题

4.1 Experimental Settings

4.1.1 Datasets

Three widely used datasets are involved in the experiments. The first one is the *Anime*[1] dataset, which contains a large number of user ratings on anime. The second one is the *MovieLens-10m*[2] dataset, which contains a large number of user ratings on movies collected from the MovieLens web site. The third one is the *Amazon-Music*[3] dataset, which contains the user ratings on music collected from Amazon. The three datasets are characterized by four statistics, including the number of users, the number of items, the number of ratings, and the number of unique category labels, where the detailed information is presented in Table 1.

For all the three datasets, the users with less than five ratings are excluded. For each item rated by a user, it is regarded as a preferred item only if the score is 5 for *MovieLens-10m* and *Amazon-Music* datasets and 10 for *Anime* dataset; otherwise, it is regarded as an unrated item. The training set of each dataset contains 80% randomly selected preferred items, and the test set of each dataset contains the remaining 20% preferred items.

4.1.2 Comparative Methods

The proposed method is compared with the following representative recommendation methods:

- **BPR-MF** [65] is the Bayesian personalized ranking based matrix factorization, which is an enhanced version of a classical collaborative filtering method for maximizing accuracy only.
- **LightGCN** [66] is a graph convolution network based collaborative filtering method, which simplifies the design of graph convolution networks and achieves substantial accuracy improvements.
- **PD-GAN** [24] is a personalized diversity-promoting generative adversarial network for multiobjective recommendation, which takes accuracy and diversity into consideration.

1. https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database
2. https://grouplens.org/datasets/movielens/10m/
3. http://jmcauley.ucsd.edu/data/amazon/

- **SGL** [39] suggests the self-supervised graph learning paradigm and applies it to LightGCN, which relieves the long-tail issue and improves the robustness, enabling LightGCN to consider both accuracy and novelty.
- **PMOEA** [49] is a recommendation method based on a probabilistic MOEA, which searches for top-K items for all users simultaneously by maximizing accuracy and diversity.
- **MOEA-ProbS** [48] is a recommendation method based on NSGA-II [67], which divides all users into multiple groups and searches for top-K items for each group of users by maximizing accuracy and diversity. However, the optimization of each group of users is independent of the others.
- **MORS** [43] is a recommendation method based on a decomposition based MOEA, which searches for top-K items for each user by maximizing accuracy, novelty and diversity.

不是所有的方法都是1000epoch，这里不知道是否有问题

For the sake of fairness, all the compared methods follow the same parameter settings as much as possible. In the deep neural network based recommendation methods, the embedding size is fixed to 64 and the neural networks are trained via Adam with a learning rate of 0.001 and a batch size of 1024 for 1000 epochs. In the MOEA based recommendation methods, the population size is set to 10 for MORS and EMMR encoding a single recommendation list and 100 for PMOEA and MOEA-ProbS encoding multiple recommendation lists, the total number of generated recommendation lists is set to $1000n$ (n is the number of users), the crossover probability is set to 1, and the mutation probability is set to 0.1. Besides, the trade-off parameter in PD-GAN is set to 0.9, and the number of user groups in the proposed EMMR is set to 10.

While the MOEA based recommendation methods can optimize any number of metrics, the deep neural network based recommendation methods can only handle fixed metrics. To achieve the comparison between different types of recommendation methods, the above methods are divided into two groups and compared separately. The first group includes BPR-MF, PD-GAN, MORS, PMOEA, MOEA-ProbS, and EMMR, where BPR-MF is employed by all the other methods to estimate the ratings of unrated items for pursuing high accuracy and novelty. The second group includes LightGCN, SGL, MORS, PMOEA, MOEA-ProbS, and EMMR, where LightGCN is employed by all the other methods to estimate the ratings of unrated items for pursuing high accuracy and diversity. When maximizing accuracy and novelty, all the MOEA based recommendation methods optimize the objectives f_1, f_2 defined in (5)(6). When maximizing accuracy and diversity, all the MOEA based recommendation methods optimize the objectives f_1, f_3 defined in (5)(7).

写反了，上面是diversity，下面是novelty

4.1.3 Evaluation Metrics

To evaluate the accuracy of a top-K recommendation list, the metric *Recall@K* is adopted to calculate the ratio of recommended items appearing in the test set, i.e.,

$$Recall@K = \frac{|\mathbf{x}_u \cap T_u|}{|T_u|}, \tag{9}$$

where $\mathbf{x}_u$ denotes a recommendation list obtained for user u and T_u denotes the set of items preferred by user u in the test set. To evaluate the novelty of a recommendation list, the metric *Novelty* defined in (6) is used. To evaluate the diversity of a recommendation list, the metric *Diversity* defined in (7) is used. Moreover, the metric *F-Score* [69] is adopted to quantitatively evaluate a recommendation list in terms of both accuracy and novelty or accuracy and diversity, which calculates the harmonic mean of two metrics:

$$\textit{F-Score} = \frac{2 \times Recall@K \times Novelty}{Recall@K + Novelty} \tag{10}$$

or

$$\textit{F-Score} = \frac{2 \times Recall@K \times Diversity}{Recall@K + Diversity}, \tag{11}$$

where a higher *F-Score* indicates a better balance between two metrics made by the recommendation list. Besides, since the MOEA based recommendation methods obtain multiple recommendation lists for each user, a single one is selected by Algorithm 4 and evaluated in the experiments.

4.2 Comparisons with State-of-the-Art Methods

Table 2 lists the performance of LightGCN, SGL, PMOEA, MOEA-ProbS, MORS, and the proposed EMMR in terms of accuracy and novelty averaged over all users. For LightGCN, it obtains the best *Recall@10* on *Anime* and *MovieLens-10m* datasets, since it is used to estimate the accuracy of candidate recommendation lists in all the other compared methods. Nevertheless, LightGCN obtains worse *Recall@10* than EMMR on *Amazon-Music* dataset, which implies that it is not accurate enough and the consideration of novelty can improve the accuracy to some extent. For SGL, it obtains the best *Novelty* on *Anime* and *MovieLens-10m* datasets, since it enhances LightGCN with self-supervised graph learning for long-tail recommendation. However, the improvement of novelty is at the expense of accuracy, where SGL cannot strike good balance between these two metrics. For PMOEA, it obtains unsatisfactory *Recall@10* and *Novelty* due to the encoding of all users in a solution, which introduces a huge search space that hinders the population from convergence. For MOEA-ProbS and MORS, they obtain good *F-Score* as well as good balance between accuracy and novelty, mainly due to the use of evolutionary multiobjective optimization. It is worth noting that the proposed EMMR obtains the best *F-Score* on all the datasets, since it can balance between accuracy and novelty as well as MOEA-ProbS and MORS, while it outperforms MOEA-ProbS and MORS with the assistance of evolutionary multitasking for convergence acceleration.

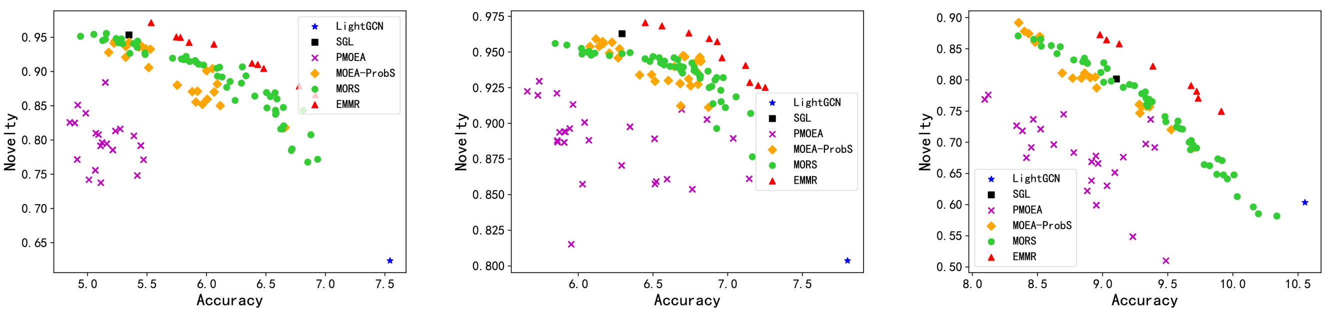
For visual observations, Fig. 5 plots the recommendation lists obtained by the compared methods for a user in each of the three datasets. For the single recommendation list obtained by LightGCN for each user, it has better accuracy but worse novelty than those obtained by the other methods. For the single recommendation list obtained by SGL for each user, it is dominated (i.e., worse in terms of both accuracy and novelty) by those obtained by the proposed EMMR. For the multiple recommendation lists obtained by PMOEA, MOEA-ProbS, MORS, and EMMR for each user, it is obvious that the recommendation lists obtained by PMOEA are dominated by those obtained by MOEA-ProbS and MORS, and the recommendation lists obtained by

TABLE 2
Accuracy and novelty performance of six recommendation methods on three datasets. The best results are shown in bold.

Dataset	*Anime*			*MovieLens-10m*			*Amazon-Music*		
Method	*Recall@10*	*Novelty*	*F-Score*	*Recall@10*	*Novelty*	*F-Score*	*Recall@10*	*Novelty*	*F-Score*
LightGCN	**0.2719**	0.5592	0.3659	**0.2366**	0.5742	0.3351	0.0379	0.7236	0.0720
SGL	0.1894	**0.8695**	0.3110	0.1534	**0.8252**	0.2587	0.0441	0.8752	0.0840
PMOEA	0.1546	0.6456	0.2495	0.1235	0.7423	0.2118	0.0245	0.7911	0.0475
MOEA-ProbS	0.2027	0.7834	0.3221	0.1899	0.7823	0.3056	0.0378	0.8254	0.0642
MORS	0.2003	0.7234	0.3137	0.1853	0.7822	0.2996	0.0447	0.8132	0.0847
The proposed EMMR	0.2525	0.8235	**0.3747**	0.2127	0.8095	**0.3369**	**0.0562**	**0.9023**	**0.1058**

0.7229

0.3865



(a) A user in *Anime* dataset. (b) A user in *MovieLens-10m* dataset. (c) A user in *Amazon-Music* dataset.

Fig. 5. Accuracy and novelty of the recommendation lists obtained by six recommendation methods for the users in three datasets.

TABLE 3
Accuracy and diversity performance of six recommendation methods on three datasets. The best results are shown in bold.

Dataset	*Anime*			*MovieLens-10m*			*Amazon-Music*		
Method	*Recall@10*	*Diversity*	*F-Score*	*Recall@10*	*Diversity*	*F-Score*	*Recall@10*	*Diversity*	*F-Score*
BPR-MF	**0.2701**	0.2500	0.2597	**0.2100**	0.5009	0.2959	0.0299	0.0288	0.0293
PD-GAN	0.2145	0.3635	0.2698	0.1843	0.6754	0.2896	0.0354	0.1060	0.0531
PMOEA	0.1153	0.3002	0.1666	0.0998	0.5742	0.1700	0.0165	0.0541	0.0253
MOEA-ProbS	0.2054	0.2843	0.2385	0.1578	0.6834	0.2564	0.0278	0.0801	0.0413
MORS	0.2234	0.295	0.2543	0.1764	0.6623	0.2786	0.0304	0.0895	0.0454
The proposed EMMR	0.2456	**0.3702**	**0.2953**	0.1927	**0.7017**	**0.3024**	**0.0401**	**0.1237**	**0.0606**

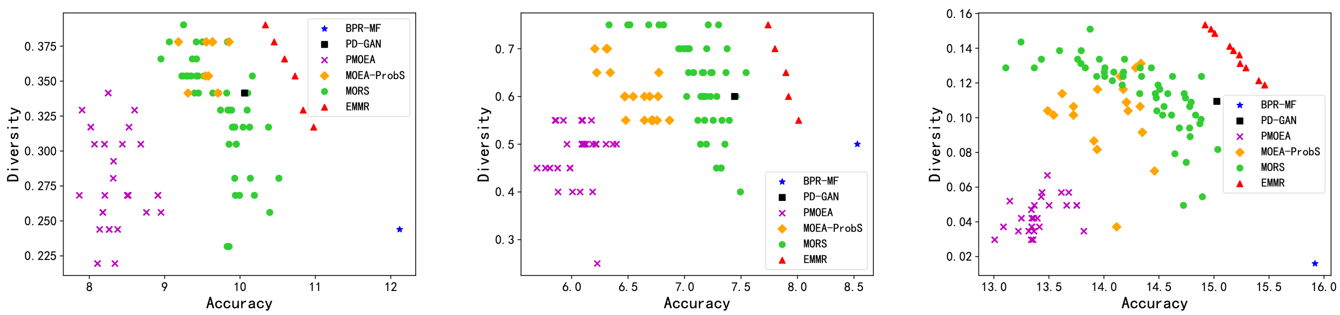
0.2950

MOEA-ProbS and MORS are dominated by those obtained by EMMR. As a consequence, the proposed EMMR holds a better balance between accuracy and novelty than the compared methods for multiobjective recommendation.

Furthermore, Table 3 presents the performance of BPR-MF, PD-GAN, PMOEA, MOEA-ProbS, MORS, and the proposed EMMR in terms of accuracy and diversity averaged over all users. Similar to the results in Table 2, BPR-MF obtains the best *Recall@10* on *Anime* and *MovieLens-10m* datasets since it is used to estimate the accuracy of candidate recommendation lists in all the other compared methods, but it still has worse *Recall@10* than EMMR on *Amazon-Music* dataset. For PD-GAN, it improves the diversity and deteriorates the accuracy; nevertheless, it has worse *Diversity* than EMMR on all the datasets. For PMOEA, MOEA-ProbS, and MORS, they exhibit worse accuracy and diversity performance than PD-GAN. By contrast, the proposed EMMR exhibits better accuracy and diversity performance than PD-GAN, resulting in the best *F-Score* on all the datasets. Besides, Fig. 6 draws the recommendation lists obtained by the compared methods for a user in each of the three datasets. It can be observed that the recommendation lists obtained by BPR-MF have better accuracy but worse diversity than those obtained by the other methods, while the recommendation lists obtained by EMMR dominate those obtained by PD-GAN, PMOEA, MOEA-ProbS, and MORS. Therefore, the superiority of the proposed EMMR over deep neural network based and MOEA based methods for multiobjective recommendation can be confirmed.

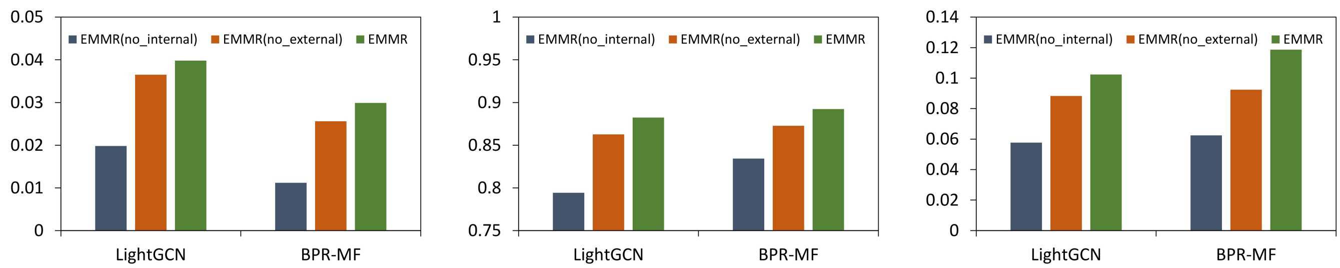
4.3 Ablation Studies

In order to verify the effectiveness of the internal knowledge transfer and external knowledge transfer strategies in EMMR, it is compared with a variant eliminating the



(a) A user in *Anime* dataset. (b) A user in *MovieLens-10m* dataset. (c) A user in *Amazon-Music* dataset.

Fig. 6. Accuracy and diversity of the recommendation lists obtained by six recommendation methods for the users in three datasets.



(a) *Recall@10*. (b) *Novelty*. (c) *Diversity*.

Fig. 7. Accuracy, novelty, and diversity performance of the proposed EMMR and its variants on *Amazon-Music* dataset.

internal knowledge transfer (i.e., crossover is always performed on two parents for the same user in Algorithm 6) and a variant eliminating the external knowledge transfer (i.e., Algorithm 7 is not performed). Fig. 7 shows the performance of EMMR and its two variants on *Amazon-Music* dataset in terms of accuracy, novelty, and diversity. Note that here the three objectives f_1, f_2, f_3 defined in (5)(6)(7) are optimized simultaneously, and both LightGCN and BPR-MF are adopted to estimate the ratings of unrated items. It can be found that the EMMR without internal knowledge transfer exhibits the worst accuracy, novelty, and diversity, which indicates that the evolutionary multitasking optimization paradigm transferring useful genetic material between tasks is efficient for handling many users. Moreover, the EMMR without external knowledge transfer is also underperformed by the original EMMR in terms of accuracy, novelty, and diversity, which means that the dynamic migration of solutions enhances the exploration ability and improves the solution quality. In short, both the internal knowledge transfer and external knowledge transfer can improve the performance of MOEAs in the multiobjective recommendation to a large number of users.

In addition, to analyze the influence of the number of user groups on the performance of EMMR, Fig. 8 plots the *Recall@10* and *F-Score* of EMMR with different numbers of user groups on *Amazon-Music* dataset, where BPR-MF is adopted to estimate the ratings of unrated items. It can be found from the figure that the performance of EMMR improves when the number of user groups increases from 2 to 10, since more groups can better avoid negative transfer happening in existing evolutionary multitasking algorithms.

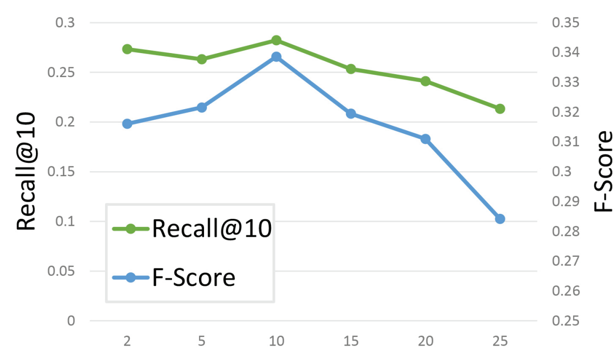


Fig. 8. Performance of the proposed EMMR with different numbers of user groups on *Amazon-Music* dataset.

Anime数据集

On the contrary, the performance of EMMR deteriorates when the number of user groups is larger than 10, which means that too many groups may eliminate positive transfer and make the performance of EMMR similar to the MOEAs handling each user independently. To summarize, it is reasonable to set the number of user groups to 10.

5 Conclusions and Future Work

To address the low efficiency of multiobjective evolutionary algorithms in searching for the recommendation lists for many users, this paper has proposed a novel evolutionary multitasking algorithm for multiobjective recommendation. With the evolutionary multitasking framework, the proposed method is able to search for the recommendation lists for all users simultaneously, in which useful genetic material is transferred between users for the acceleration of convergence speed. Moreover, the users are divided into multiple groups to promote positive transfer and avoid negative transfer, where the users in the same group are handled by evolving a population with internal knowledge transfer, and all the populations are cooperatively evolved with external knowledge transfer. Experimental results on three datasets have demonstrated that the proposed method is superior over existing deep neural network based and evolutionary algorithm based recommendation methods.

This work has revealed the bright prospect of evolutionary multitasking in multiobjective recommendation. Hence, it can be further applied to other recommendation tasks such as multi-criteria recommendation [70] and group recommendation [37]. Besides, since evolutionary algorithms are optimizers that cannot estimate the ratings of unrated items, it is highly desirable to adopt more efficient recommendation models or utility functions for estimating the accuracy of candidate recommendation lists.

Acknowledgments

This work was supported in part by the National Key R&D Program of China under Grant 2018AAA0100100, in part by the National Natural Science Foundation of China under Grant 61906001, Grant 62136008, and Grant U21A20512, and in part by the Collaborative Innovation Program of Universities in Anhui Province under Grant GXXT-2020-013.

References

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez, “Recommender systems survey,” *Knowledge-Based Systems*, vol. 46, pp. 109–132, 2013.

[2] P. Resnick and H. R. Varian, “Recommender systems,” *Communications of the ACM*, vol. 40, no. 3, pp. 56–58, 1997.

[3] P. Winoto and T. Y. Tang, “The role of user mood in movie recommendations,” *Expert Systems with Applications*, vol. 37, no. 8, pp. 6086–6092, 2010.

[4] C. Gao, T.-H. Lin, N. Li, D. Jin, and Y. Li, “Cross-platform item recommendation for online social e-commerce,” *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[5] Y.-L. Hsu, Y.-C. Tsai, and C.-T. Li, “FinGAT: Financial graph attention networks for recommending top-K profitable stocks,” *arXiv preprint arXiv:2106.10159*, 2021.

[6] Y. Ge, H. Li, and A. Tuzhilin, “Route recommendations for intelligent transportation services,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 3, pp. 1169–1182, 2019.

[7] N. Akhtar and D. Agarwal, “An efficient mining for recommendation system for academics,” *International Journal of Recent Technology and Engineering*, vol. 8, no. 5, pp. 1619–1626, 2020.

[8] R. A. Hamid, A. S. Albahri, J. K. Alwan, Z. T. Al-qaysi, O. S. Albahri, A. A. Zaidan, A. Alnoor, A. H. Alamoodi, and B. B. Zaidan, “How smart is e-tourism? a systematic review of smart tourism recommendation system applying data management,” *Computer Science Review*, vol. 39, p. 100337, 2021.

[9] Y. Tian, S. Yang, L. Zhang, F. Duan, and X. Zhang, “Surrogate-assisted multiobjective evolutionary algorithm for large-scale task-oriented pattern mining,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 3, no. 2, pp. 106–116, 2019.

[10] M. J. Pazzani and D. Billsus, “Content-based recommendation systems,” in *The Adaptive Web*, 2007, pp. 325–341.

[11] Z.-D. Zhao and M.-S. Shang, “User-based collaborative-filtering recommendation algorithms on hadoop,” in *Proceedings of the 2010 Third International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 478–481.

[12] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceedings of the 10th International Conference on World Wide Web*, 2001, pp. 285–295.

[13] A. Mnih and R. R. Salakhutdinov, “Probabilistic matrix factorization,” in *Advances in Neural Information Processing Systems*, vol. 20, 2007.

[14] A. K. Pandey and D. S. Rajpoot, “Resolving cold start problem in recommendation system using demographic approach,” in *Proceedings of the 2016 International Conference on Signal Processing and Communication*, 2016.

[15] R. Burke, “Knowledge-based recommender systems,” *Encyclopedia of library and information systems*, vol. 69, no. Supplement 32, pp. 175–186, 2000.

[16] M. Zihayat, A. Ayanso, X. Zhao, H. Davoudi, and A. An, “A utility-based news recommendation system,” *Decision Support Systems*, vol. 117, pp. 14–27, 2019.

[17] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, “Recommender system application developments: A survey,” *Decision Support Systems*, vol. 74, pp. 12–32, 2015.

[18] L. Sharma and A. Gera, “A survey of recommendation system: Research challenges,” *International Journal of Engineering Trends and Technology*, vol. 4, no. 5, pp. 1989–1992, 2013.

[19] S. Wang, K. Zhang, L. Wu, H. Ma, R. Hong, and M. Wang, “Privileged graph distillation for cold start recommendation,” in *Proceedings of the 2021 International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 1187–1196.

[20] H. Wang, P. Zhang, T. Lu, H. Gu, and N. Gu, “Hybrid recommendation model based on incremental collaborative filtering and content-based algorithms,” in *Proceedings of the 2017 IEEE International Conference on Computer Supported Cooperative Work in Design*, 2017, pp. 337–342.

[21] N. Pereira and S. L. Varma, *Financial planning recommendation system using content-based collaborative and demographic filtering*. Springer, Singapore, 2019, ch. Smart Innovations in Communication and Computational Sciences, pp. 141–151.

[22] M. Prangl, R. Bachlechner, and H. Hellwagner, “A hybrid recommender strategy for personalized utility-based cross-modal multimedia adaptation,” in *Proceedings of the 2007 IEEE International Conference on Multimedia and Expo*, 2007.

[23] S. Wu, F. Sun, W. Zhang, X. Xie, and B. Cui, “Graph neural networks in recommender systems: A survey,” *ACM Computing Surveys*, 2022.

[24] Q. Wu, Y. Liu, C. Miao, B. Zhao, Y. Zhao, and L. Guan, “PD-GAN: Adversarial learning for personalized diversity-promoting recommendation,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, vol. 19, 2019, pp. 3870–3876.

[25] X. Wang, J. Wen, F. Luo, W. Zhou, and H. Ren, “Personalized recommendation system based on support vector machine and particle swarm optimization,” in *Proceedings of the 2015 International Conference on Knowledge Science, Engineering and Management*, 2015, pp. 489–495.

[26] X. Zhang, F. Duan, L. Zhang, F. Cheng, Y. Jin, and K. Tang, “Pattern recommendation in task-oriented applications: A multi-objective perspective [application notes],” *IEEE Computational Intelligence Magazine*, vol. 12, no. 3, pp. 43–53, 2017.

[27] R. R. Yager, “Fuzzy logic methods in recommender systems,” *Fuzzy Sets and Systems*, vol. 136, no. 2, pp. 133–149, 2003.

[28] J. Guo, J. Deng, and Y. Wang, “An intuitionistic fuzzy set based hybrid similarity model for recommender system,” *Expert Systems with Applications*, vol. 135, pp. 153–163, 2019.

[29] L. Wu, X. He, X. Wang, K. Zhang, and M. Wang, “A survey on accuracy-oriented neural recommendation: From collaborative filtering to information-rich recommendation,” *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[30] S. M. McNee, J. Riedl, and J. A. Konstan, “Being accurate is not enough: how accuracy metrics have hurt recommender systems,” in *CHI'06 extended abstracts on Human factors in computing systems*, 2006, pp. 1097–1101.

[31] P. Gupta, D. Garg, P. Malhotra, L. Vig, and G. M. Shroff, “NISER: Normalized item and session representations with graph neural networks,” *arXiv preprint arXiv:1909.04276*, 2019.

[32] J. Carbonell and J. Goldstein, “The use of MMR, diversity-based reranking for reordering documents and producing summaries,” in *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1998, pp. 335–336.

[33] L. Zhang, “The definition of novelty in recommendation system,” *Journal of Engineering Science and Technology Review*, vol. 6, no. 3, pp. 141–145, 2013.

[34] L. Wu, L. Chen, P. Shao, R. Hong, X. Wang, and M. Wang, "Learning fair representations for recommendation: A graph based perspective," in *Proceedings of The Web Conference*, 2021, pp. 2198–2208.

[35] Z. Lyu, Y. Wu, J. Lai, M. Yang, C. Li, and W. Zhou, "Knowledge enhanced graph neural networks for explainable recommendation," *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[36] L. Wu, Y. Ge, Q. Liu, E. Chen, R. Hong, J. Du, and M. Wang, "Modeling the evolution of users' preferences and social links in social networking services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 6, pp. 1240–1253, 2017.

[37] X. Lin, M. Zhang, Y. Zhang, Z. Gu, Y. Liu, and S. Ma, "Fairness-aware group recommendation with Pareto-efficiency," in *Proceedings of the Eleventh ACM Conference on Recommender Systems*, 2017, pp. 107–115.

[38] Y. Zheng and D. X. Wang, "A survey of recommender systems with multi-objective optimization," *Neurocomputing*, vol. 474, pp. 141–153, 2022.

[39] J. Wu, X. Wang, F. Feng, X. He, L. Chen, J. Lian, and X. Xie, "Self-supervised graph learning for recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 726–735.

[40] L. Tang, L. Zhang, P. Luo, and M. Wang, "Incorporating occupancy into frequent pattern mining for high quality pattern recommendation," in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2012, pp. 75–84.

[41] Q. Ding, Y. Liu, C. Miao, F. Cheng, and H. Tang, "A hybrid bandit framework for diversified recommendation," *arXiv preprint arXiv:2012.13245*, 2020.

[42] S. Patil, D. Banerjee, and S. Sural, "A graph theoretic approach for multi-objective budget constrained capsule wardrobe recommendation," *ACM Transactions on Information Systems*, vol. 40, no. 1, pp. 1–33, 2022.

[43] S. Wang, M. Gong, H. Li, and J. Yang, "Multi-objective optimization for long tail recommendation," *Knowledge-Based Systems*, vol. 104, pp. 145–155, 2016.

[44] S. Patil, D. Banerjee, and S. Sural, "Recommendation with diversity: Mass diffusion model based on trust network and object reputation," *ACM Transactions on Information Systems*, p. 5527568, 2021.

[45] Y. Tian, H. Chen, H. Ma, X. Zhang, K. C. Tan, and Y. Jin, "Integrating conjugate gradients into evolutionary algorithms for large-scale continuous multi-objective optimization," *IEEE/CAA Journal of Automatica Sinica*, 2022.

[46] X. Cai, Z. Hu, P. Zhao, W. Zhang, and J. Chen, "A hybrid recommendation system with many-objective evolutionary algorithm," *Expert Systems with Applications*, vol. 159, p. 113648, 2020.

[47] C. Xu, A. S. Ding, and S. S. Liao, "A privacy-preserving recommendation method based on multi-objective optimisation for mobile users," *International Journal of Bio-Inspired Computation*, vol. 16, no. 1, pp. 23–32, 2020.

[48] Y. Zuo, M. Gong, J. Zeng, L. Ma, and L. Jiao, "Personalized recommendation based on evolutionary multi-objective optimization [research frontier]," *IEEE Computational Intelligence Magazine*, vol. 10, no. 1, pp. 52–62, 2015.

[49] L. Cui, P. Ou, X. Fu, Z. Wen, and N. Lu, "A novel multi-objective evolutionary algorithm for recommendation systems," *Journal of Parallel and Distributed Computing*, vol. 103, pp. 53–63, 2017.

[50] Y. Tian, L. Si, X. Zhang, R. Cheng, C. He, K. C. Tan, and Y. Jin, "Evolutionary large-scale multi-objective optimization: A survey," *ACM Computing Surveys*, vol. 54, no. 8, p. 174, 2022.

[51] A. Gupta, Y.-S. Ong, and L. Feng, "Multifactorial evolution: Toward evolutionary multitasking," *IEEE Transactions on Evolutionary Computation*, vol. 20, no. 3, pp. 343–357, 2015.

[52] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, "Multiobjective multifactorial optimization in evolutionary multitasking," *IEEE Transactions on Cybernetics*, vol. 47, no. 7, pp. 1652–1665, 2017.

[53] J. Li, H. Li, Y. Liu, and M. Gong, "Multi-fidelity evolutionary multitasking optimization for hyperspectral endmember extraction," *Applied Soft Computing*, vol. 111, p. 107713, 2021s.

[54] J. Hu and P. Li, "Collaborative multi-objective ranking," in *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018, pp. 1363–1372.

[55] Y. Tian, C. He, R. Cheng, and X. Zhang, "A multistage evolutionary algorithm for better diversity preservation in multiobjective optimization," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 9, pp. 5880–5894, 2021.

[56] B. Geng, L. Li, L. Jiao, M. Gong, Q. Cai, and Y. Wu, "NNIA-RS: A multi-objective optimization based recommender system," *Physica A*, vol. 424, pp. 383–397, 2015.

[57] W. Fang, Q. Zhang, J. Sun, and X.-J. Wu, "Mining high quality patterns using multi-objective evolutionary algorithm," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[58] L. Xie, Z. Hu, X. Cai, W. Zhang, and J. Chen, "Explainable recommendation based on knowledge graph and multi-objective optimization," *Complex & Intelligent Systems*, vol. 7, no. 3, pp. 1241–1252, 2021.

[59] K. K. Bali, Y. S. Ong, A. Gupta, and P. S. Tan, "Multifactorial evolutionary algorithm with online transfer parameter estimation: MFEA-II," *IEEE Transactions on Evolutionary Computation*, vol. 24, no. 1, pp. 69–83, 2020.

[60] K. K. Bali, A. Gupta, Y. S. Ong, and P. S. Tan, "Cognizant multitasking in multiobjective multifactorial evolution: MO-MFEA-II," *IEEE Transactions on Cybernetics*, vol. 51, no. 4, pp. 1784–1796, 2021.

[61] L. Bao, Y. Qi, M. Shen, X. Bu, J. Yu, Q. Li, and P. Chen, "An evolutionary multitasking algorithm for cloud computing service composition," in *Proceedings of the World Congress on Services*, 2018, pp. 130–144.

[62] K. Chen, B. Xue, M. Zhang, and F. Zhou, "Evolutionary multitasking for feature selection in high-dimensional classification via particle swarm optimisation," *IEEE Transactions on Evolutionary Computation*, 2021.

[63] L. Feng, Y. Huang, L. Zhou, J. Zhong, A. Gupta, K. Tang, and K. C. Tan, "Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem," *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3143–3156, 2020.

[64] F. Zhang, Y. Mei, S. Nguyen, M. Zhang, and K. C. Tan, "Surrogate-assisted evolutionary multitask genetic programming for dynamic flexible job shop scheduling," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 4, pp. 651–665, 2021.

[65] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," *arXiv preprint arXiv:1205.2618*, 2012.

[66] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 639–648.

[67] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[68] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in *Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, 1996, pp. 103–114.

[69] Y. Liang, T. Qian, Q. Li, and H. Yin, "Enhancing domain-level and user-level adaptivity in diversified recommendation," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 747–756.

[70] D. Monti, G. Rizzo, and M. Morisio, "A systematic literature review of multicriteria recommender systems," *Artificial Intelligence Review*, vol. 54, pp. 427–468, 2021.