J. Parallel Distrib. Comput. 103 (2017) 53–63

Contents lists available at ScienceDirect

J. Parallel Distrib. Comput.

journal homepage: www.elsevier.com/locate/jpdc

# A novel multi-objective evolutionary algorithm for recommendation systems

Laizhong Cui *, Peng Ou, Xianghua Fu, Zhenkun Wen, Nan Lu

*College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, PR China*

## HIGHLIGHTS

- A new topic diversity indicator is introduced, which can be used to measure various kinds of items in a recommendation list.
- A new probabilistic multi-objective evolutionary algorithm (PMOEA) is presented, which is suitable for the recommendation systems.
- A new crossover operator is proposed to generate new solution, called the multi-parent probability genetic operator.
- The experimental results show that PMOEA can achieve a good balance between precision and diversity.

## ARTICLE INFO

*Article history:*
Received 1 July 2016
Received in revised form
7 September 2016
Accepted 18 October 2016
Available online 29 October 2016

*Keywords:*
Recommendation algorithm
Multi-objective optimization
Topic diversity
Genetic operator

## ABSTRACT

Nowadays, the recommendation algorithm has been used in lots of information systems and Internet applications. The recommendation algorithm can pick out the information that users are interested in. However, most traditional recommendation algorithms only consider the precision as the evaluation metric of the performance. Actually, the metrics of diversity and novelty are also very important for recommendation. Unfortunately, there is a conflict between precision and diversity in most cases. To balance these two metrics, some multi-objective evolutionary algorithms are applied to the recommendation algorithm. In this paper, we firstly put forward a kind of topic diversity metric. Then, we propose a novel multi-objective evolutionary algorithm for recommendation systems, called PMOEA. In PMOEA, we present a new probabilistic genetic operator. Through the extensive experiments, the results demonstrate that the combination of PMOEA and the recommendation algorithm can achieve a good balance between precision and diversity.



## 1. Introduction

With the arrival of the information age, people are faced with a large number of information resources on Internet. This is the problem of information overload, and it causes people cannot quickly find the information that they are interested in. In order to make the user quickly get the interested information, the recommendation system appears in people's eyes. Ricci et al. [24] introduced the basic ideas, concepts and some applications of the recommendation system. The recommendation system can pick out the interested information resources to the user based on the user's history experience. It can achieve the purpose of saving time and cost for people. The recommendation system has been successfully applied in all kinds of network activities [2]. Traditional recommendation algorithm can be roughly divided into content-based recommendation algorithm, collaborative filtering recommendation algorithm and hybrid recommendation algorithm.

The traditional recommendation system evaluates the ratings of items based on the user's experiences, and then the top-*n* high rating items are selected to recommend. In order to improve the accuracy of the recommendation, the recommendation algorithms have been continuously developed. Some researchers make improvements on the methods of item rating evaluation. Liu et al. [15] analyzed a variety of calculation methods on the similarity, and proposed a new similarity method to improve the traditional collaborative filtering algorithm. Fan et al. [8] integrated content-based recommendation algorithm and user activity level to predict the empty values in user–item matrix when calculating user similarity. Zhou et al. [36] utilized the bipartite network to improve the recommendation system. Koren et al. [11] decomposed user–item

* Corresponding author.
*E-mail address:* cuilz@szu.edu.cn (L. Cui).

http://dx.doi.org/10.1016/j.jpdc.2016.10.014
0743-7315/

matrix to evaluate the rating of unknown items to the user. In addition, many researchers consider the trust factor to improve the accuracy of recommendation algorithm. O'Donovan et al. [19] presented two trust models and incorporated them into standard collaborative filtering algorithm. Wanita et al. [31] described the characteristics of social network, trust, and the existing trust models in social networks. The trust of neighbor nodes is used to improve the item rating evaluation for the social recommendation [33,30,10,12]. Wang et al. [29] presented a joint social and content recommendation for user-generated videos in online social network. More and more recommendation systems are deployed on cloud environment. To optimize the resource scheduling in cloud systems, Qiu et al. [21] put forward a genetic-based optimization algorithm and Li et al. [14] presented an online optimization method about scheduling preemptable tasks on cloud for real-time systems. In order to improve the user experience of watching videos on mobile devices and social TV, Niu et al. [18] introduced a wireless interface scheduling algorithm to select proper wireless interfaces and Wang et al. [28] proposed a group recommendation algorithm based on external followee.

However, in the above recommendation algorithms, accuracy is regarded as a single evaluation metric. As shown in [9], we know that the traditional recommendation algorithms are faced with four challenges: the data sparse problem, the cold start problem, the single evaluation metric, and the problem of false data. The first two challenges have been slowed down in the existing recommendation algorithms. For the issue of the single evaluation metric, a good recommendation algorithm not only relies on the accuracy to measure the performance, but also takes some other metrics into account. McNee et al. [17] indicated that accuracy metrics have hurt recommendation systems. When a person likes to eat bread for breakfast, we not only recommend the bread to the user, but also need to recommend other types of breakfast. Some non-accuracy indicators of the recommendation system are proposed [2,34,13,3,16,1,37,20]. Bobadilla et al. [2] made a summary about the accuracy indicators and non-accuracy indicators. Accuracy indicators contain precision, recall and F1-measure. Non-accuracy indicators contain novelty, diversity, stability, and reliability. Zhang et al. [34] defined diversity according to the differences between items. Lathia et al. [13] showed that temporal diversity is an important facet of the recommendation systems. Castells et al. [3] demonstrated that novelty is the difference between present and past experience, and diversity is the internal differences within parts of an experience. Ma et al. [16] proposed an algorithm to solve the dilemma between accuracy, diversity and novelty based on bidirectional transfer. They defined diversity as the mean value of Hamming distance. Belem et al. [1] took a tag diversity into account for the topic diversity. Ziegler et al. [37] used dissimilarity rank to calculate the diversity. Panniello et al. [20] used the Simpson's diversity index, the Shannon's entropy and the Tidemann & Hall's index to represent recommendation diversity. In our reality life, the diversity should be used to describe various types of items. But sometimes, an item belongs to several types. In this situation, all above diversity indicators cannot effectively describe the diversity of items in a recommendation list. In this paper, we introduce a new indicator for the topic diversity.

Most traditional recommendation algorithms concentrate on item rating evaluation, which means the items are sorted by item rating, and the top-$n$ items are selected to recommend for the user. These algorithms can try to ensure the accuracy since the high rating items are recommended. However, they could not guarantee the non-accuracy indicators of the recommendation list. In order to solve the conflict between the accuracy indicator and non-accuracy indicator, some multi-objective algorithms are introduced into the recommendation system [9,39,26,22,23,27]. Geng et al. [9] proposed a recommendation algorithm, which utilizes non-dominated Neighbor Immune Algorithm. Zuo et al. [39] presented a MOEA-based recommendation method to solve conflicting between accuracy and diversity. Wang et al. [26] designed a multi-objective framework for long tail items recommendation. Ribeiro et al. [22] proposed a hybrid recommendation approach and utilized multi-objective approach to find several hybridization parameters of different algorithms. Ribeiro et al. [23] introduced the Pareto efficiency concept and presented a way to combine several recommendation algorithms that the multi-objective can be maximized simultaneously. Wang et al. [27] proposed a multi-objective evolutionary algorithm based on the decomposition to optimize the rating and the popularity of items simultaneously.

Although the multi-objective evolutionary algorithms can improve the performance of the recommendation algorithms, they are not satisfying enough. In this paper, we will try to carry on the improvement based on the application features of the recommendation system. Actually, in the recommendation system, the more times an item is recommended, the more likely it is to be accepted. In other words, when many friends recommend the same item to a user, this user would pay more attention to this item. According to this fact, we present a new multi-objective evolutionary algorithm (called PMOEA) for the recommendation systems. Our proposed multi-objective evolution algorithm is based on NSGA-II. The main contributions of this paper are as follows:

(1) We propose a new topic diversity indicator, which can be used to measure various kinds of items in a recommendation list.
(2) We propose a probabilistic multi-objective evolutionary algorithm, called PMOEA, which is suitable for the recommendation systems. PMOEA utilizes the probability to estimate whether the gene is inherited to the offspring solution. In PMOEA, we also design a new solution generation method, called the multi-parent probability genetic operator.
(3) Compared with some known recommendation algorithms, the experimental results show that the combination of PMOEA and the recommendation algorithm can improve the metrics of diversity and novelty, sacrificing a certain degree of precision.

The remainder of this paper is organized as follows. Section 2 introduces the research background and related works. In Section 3, we present the multi-objective recommendation framework and propose a novel multi-objective optimization for recommendation algorithms. The experimental results are discussed in Section 4. Finally, Section 5 concludes this paper.

## 2. Background

In this section, we will introduce the background of item rating evaluation, the typical multi-objective evolutionary algorithm and a classical multi-objective evolutionary recommendation algorithm.

### 2.1. Item rating evaluation

The traditional recommendation algorithm is based on the evaluation about the ratings of unknown items. And then, it selects the top-$n$ items to recommend. The item rating evaluation is the basic step for a recommendation algorithm. We briefly introduce User-based Collaborative Filtering algorithm, Item-based Collaborative Filtering algorithm [25] and ProbS method [36]. Collaborative Filtering algorithm utilizes the similarity between users or items to make item rating evaluation. ProbS method uses the concept of bipartite network and the allocating resource to make item rating evaluation. We will briefly introduce these three methods as follows.

User-based Collaborative Filtering (CF_User in short) algorithm relies on the homogeneity between users, which is based on the fact that the similar users have the same behaviors or interests. According to the user's preference for items, some similarity neighbors are found, and then the item ratings to those users are predicted based on these found neighbors. The cosine similarity is always used as the calculation method of similarity, which can be described as follows,

$$sim\,(u, v) = \frac{r_u \cdot r_v}{|r_u| \cdot |r_v|} \tag{1}$$

where $r_u$ is the item rating vector of user $u$, and $r_v$ is the item rating vector of user $v$. $sim(u, v)$ is the similarity between user $u$ and user $v$.

The rating of an item can be calculated as follows,

$$PR\,(u, i) = \frac{\sum\limits_{v \in S(u,K)} sim\,(u, v)\, r\,(v, i)}{\sum\limits_{v \in S(u,K)} sim\,(u, v)} \tag{2}$$

where $r(v, i)$ is the rating of item $i$ given by user $v$. $PR(u, i)$ is the predicted rating for user $u$ to item $i$. $K$ is the number of neighbors for user $u$, and $S(u, K)$ is the set of user $u$'s $K$ neighbors.

Item-based Collaborative Filtering (CF_Item in short) algorithm is similar to CF_User algorithm. The only difference is that CF_Item algorithm calculates the similarity between items instead of users. It utilizes user's preferences on items to find similar items, and then according to the preferences of the user's history, it evaluates unknown item rating to the user. The cosine similarity is always used as the calculation method of similarity, which is described as follows,

$$sim\,(i, j) = \frac{r_i \cdot r_j}{|r_i| \cdot |r_j|} \tag{3}$$

where $r_i$ is the rating vector of item $i$, and $r_j$ is the rating vector of item $j$. $sim(i, j)$ is the similarity between item $i$ and item $j$.

The rating of an item can be calculated as follows,

$$PR\,(u, i) = \frac{\sum\limits_{j \in S(i,K)} sim\,(i, j)\, r\,(u, j)}{\sum\limits_{j \in S(i,K)} sim\,(i, j)} \tag{4}$$

where $sim(i, j)$ is the similarity between item $i$ and item $j$, $r(u, j)$ is the rating of item $j$ given by user $u$. $PR(u, i)$ is the predicted rating for user $u$ to item $i$. $K$ is the number of neighbors for item $i$, and $S(i, K)$ is the set of item $i$'s $K$ neighbors.

ProbS method [36] utilizes bipartite network to evaluate item rating. It will divide the network into user set and item set. The item set is $X = [x_1, x_2 \ldots x_n]$, and the user set is $Y = [y_1, y_2 \ldots y_m]$. $n$ is the number of items, and $m$ is the number of users. In ProbS method, the allocation of resources between users and items is considered, and the top-$n$ items are selected to recommend. ProbS method uses the user's ratings on known items to predict the user's ratings on unknown items. It first calculates the transition resources from the items to the users. The resources distributed from items to users can be calculated as follows,

$$f\,(y_i) = \sum_{\beta=1}^{n} \frac{r_{i,\beta}}{k_\beta} \tag{5}$$

where $r_{i,\beta}$ is the rating of item $\beta$ given by user $i$, and $k_\beta$ is the number of users that have rating on item $\beta$. $f(y_i)$ denotes the resources that user $i$ has obtained.

Then, the users have obtained the resources and they will distribute the resources to the items that they have rated. The ratings of items are described by the resources of items. The resources distributed from the users to the items can be calculated as follows,

$$f'(x_\alpha) = \sum_{i=1}^{m} \frac{r_{i,\alpha} f\,(y_i)}{k_i} = \sum_{i=1}^{m} \frac{r_{i,\alpha}}{k_i} \sum_{\beta=1}^{n} \frac{r_{i,\beta}}{k_\beta} \tag{6}$$

where $r_{i,\alpha}$ is the rating of item $\alpha$ given by user $i$, and $k_i$ is the number of items with rating given by user $i$.

### 2.2. Multi objective evolutionary algorithm

Generally, the evolutionary algorithm is divided into single objective optimization algorithm and multi-objective optimization algorithm according to the number of objective functions. MOEA/D [35], NSGA-II [7] and SPEA2 [38] are the three classical multi-objective evolutionary algorithms. Multi-objective optimization algorithm is mainly used to solve the optimization problem of two or more conflicting objective functions. However, we know that there is a certain degree of conflict between the accuracy indicator and non-accuracy indicator in the recommendation system. Therefore, some researchers began to study the multi-objective evolutionary algorithm for the recommendation system. Multi-objective optimization problem is to optimize a set of conflicting functions. As described in [23], its formula is given as follows,

$$maximize\ F\,(x) = (f_1(x), f_2(x) \cdots f_m(x))^T \tag{7}$$

where $\Omega \in R^n$ is the decision space, and $x = [x_1, x_2 \cdots x_n]$, $x \in \Omega$. $x$ is a candidate solution. $n$ is the dimension of the solution, and $m$ is the number of the objective functions. $F : \Omega \to R^m$, and $R^m$ is the objective space.

If there are $u_i \geq v_i$ for each $u_i, v_i (u, v \in R^m)$, and there is at least one index $j$, making $u_j > v_j$, we can say that the solution $u$ dominates solution $v$, $i, j \in (1, \ldots, m)$. For a solution $x^*$, if there is no other solution $x \in \Omega$ such that $F\,(x)$ dominates $F\,(x^*)$, we call the solution $x^*$ the Pareto optimal solution. The set of all the Pareto optimal solutions is called Pareto set ($PS$). $PF = \{F\,(x) \in R^m | x \in PS\}$ is the Pareto front ($PF$).

### 2.3. MOEA-ProbS

Nowadays, some researchers have utilized the multi-objective evolutionary algorithm for recommendation system. They firstly evaluate rating of the item, and then according to the objective function to guide and select the appropriate recommended list for users. Zuo et al. [39] proposed a MOEA-Probs recommendation algorithm, which is mainly used to solve the balance between accuracy and diversity. In MOEA-Probs, the ProbS method [36] is used to evaluate the ratings of the items. And then, NSGA-II is used to select the recommendation list. Its two objective functions are described as Eqs. (8) and (9).

$$PR = \frac{\sum\limits_{i \in S} \sum\limits_{\alpha=1}^{L} pr_{i,\alpha}}{|S| \cdot L}. \tag{8}$$

In Eq. (8), $|S|$ is the number of users to recommend, and $L$ is the length of a recommendation list for every user. $pr_{i,\alpha}$ is the rating for user $i$ on item $\alpha$.

$$CV = \frac{N_{dif}}{N}. \tag{9}$$

In Eq. (9), $N_{dif}$ is the number of different items in a recommendation list, and $N$ is the total number of items in the recommendation system.

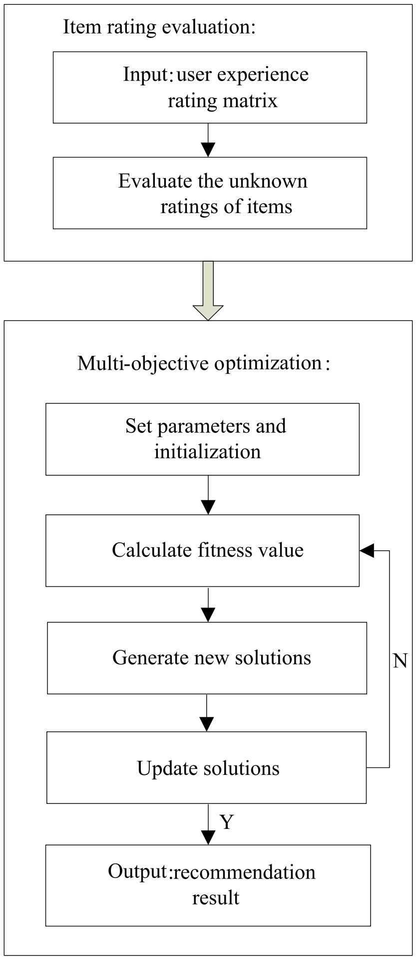



**Fig. 1.** The working process of a typical multi-objective optimization recommendation algorithm.

Regarding to the above two objective functions, *PR* can be well used to guide accuracy, but *CV* cannot describe the diversity of the recommendation. *CV* is the coverage rate. Actually, coverage rate cannot be a good indicator to recommend different types of items. It can describe the different items, but cannot describe the different types of items.

There is another problem that the standard crossover operator of MOEA-ProbS [39] is used to generate the new solution. Although there is no problem in theory, we have no corresponding actual physical meaning to describe the actual recommendation algorithm.

## 3. The proposed multi-objective evolutionary algorithm

In this section, we first introduce the framework of a multi-objective recommendation algorithm. Second, we formulate two objective functions for recommendation. One objective function is the accuracy function, which represents the accuracy of a recommendation list. It is used to evaluate whether the recommendation list satisfies the user's interests. Another objective function is the diversity function. Generally, diversity denotes the different types of items in the recommendation list. To obtain a more reasonable recommendation list, we introduce a new topic diversity indicator. It denotes the number of topics in the recommendation list. Finally, in order to solve the conflict between accuracy and diversity of the recommendation, we design a probabilistic multi-objective evolutionary algorithm (called PMOEA), which is suitable for the recommendation systems. The details are depicted in the following subsections.

### 3.1. The framework of a multi-objective recommendation algorithm

Recently, multi-objective optimization algorithms have been well applied to recommendation systems. The motivation of the multi-objective optimization for recommendation is to formulate the recommendation to be a multi-objective optimization problem and solve it. In this paper, we propose a new multi-objective evolutionary algorithm for recommendation systems based on NSGA-II [7], which combines the non-dominated relation and the genetic algorithm (GA).

The working process of a typical multi-objective optimization recommendation algorithm is shown in Fig. 1. The working process consists of two stages: item rating evaluation and multi-objective optimization. In the stage of item rating evaluation, according to the historical user–item rating matrix, the unknown rating of items for each input user will be predicted by some item rating evaluation algorithms. In the stage of multi-objective optimization, the initial solution and parameters for multi-objective optimization algorithm are initialized to enter a loop process. In the loop process, the fitness value is calculated first, which is the value calculated by the objective function. And then, new solutions are produced through the operations of crossover and mutation on the original solution, and each solution for the next generation is updated by the comparison of those new solutions and the old solutions. This loop process will be repeated until the termination condition is satisfied. The final solution is the recommendation result for the input users.

### 3.2. Objective functions

In the multi-objective optimization recommendation algorithm, the quality of a recommendation list depends on the objective functions. Here, we use the accuracy function and diversity function as the objective functions. The accuracy function is related to the sum of item rating in the recommendation list. It can guide the recommendation list to have larger acceptance probability for the user. The diversity function is related to the topics of items in the recommendation list. It can guide the recommendation list tending to contain more topics.

#### 3.2.1. Objective function of accuracy

Most traditional recommendation algorithms are based on the prediction rating and select the items with high rating as their recommendation result for the user. A user gives a higher rating to an item, which means this user prefers this item. According to the accuracy functions in [39,26,27], our objective function of accuracy is also based on the evaluated rating of items. The objection function of accuracy is described as follows,

$$PR = \frac{\sum_{u \in U} \sum_{i \in L} r_{u,i}}{|U| \cdot |L|} \tag{10}$$

where $|U|$ is the number of users to recommend, $|L|$ is the length of a recommendation list for every user, and $r_{u,i}$ is the rating of item $i$ given by user $u$.

#### 3.2.2. Objective function of diversity

In recent years, many researchers have studied non-accuracy indicators about the recommendation list for the recommend system. The diversity of a recommendation list is used to indicate the difference between items in a recommendation list. Furthermore, the best way to describe the difference between items is using the various kinds of items. Our objective function of diversity is based on the variety of topics or types. In a recommendation list $L_u = [x_1, x_2 \dots x_n]$ for user $u$, if the item $x_1$ belongs to topic $z_a$ and $z_b$, the item $x_1$'s topics can be depicted as a set $t_{x_1} = [z_a, z_b]$. Actually, an item may belong to several topics. The recommendation list $L_u$ contains topic set $z_{L_u} = \left[t_{x_1}, t_{x_2} \dots t_{x_n}\right] = [z_a, z_b \dots z_m, z_c \dots z_r, \dots, z_m]$. There may be some common topics in the set $z_{L_u}$. Therefore, we remove the same topics and keep only one for these same topics in the set $z_{L_u}$. The topic set $z_{L_u}$ can be described as $s_{L_u} = [z_a, z_b \dots z_r]$.

Panniello et al. [20] measured the recommendation diversity by several classical measurements, including the Simpson's diversity index, the Shannon's entropy and the Tidemann & Hall's index. In those methods, Shannon's entropy measures the situation of the individual distribution in the overall specimen. Inspired by Shannon's entropy, we propose a concept for the topic distribution of a recommendation list. The topic distribution describes the equilibrium status of the topics in a recommendation list. $H(L_u)$ denotes the topic distribution of recommendation list $L_u$, and it is calculated as follows,

$$H(L_u) = -\sum_{j \in s_{L_u}} q_j \log(q_j) \tag{11}$$

where $L_u$ is the recommendation list for the user $u$. $j$ is a topic in recommendation list $L_u$, and $q_j$ is the probability of the occurrence of topic $j$ in recommendation list $L_u$. The probability of occurrence of topic $j$ is calculated as follows,

$$q_j = \frac{\left|s_{L_u,j} \cap z_{L_u}\right|}{\left|z_{L_u}\right|} \tag{12}$$

where $\left|z_{L_u}\right|$ is the number of total topics in recommendation list $L_u$, and $\left|s_{L_u,j}\right|$ is the number of topic $j$ in the set $s_{L_u}$.

However, the topic distribution is not good enough to describe the topic diversity of a recommendation list. This is because every recommendation list may have different total numbers of topics. Therefore, we take the number of different topics in a recommendation list into account and introduce a new diversity indicate. We refer to this diversity indicate as the various numbers for topic distribution. Regarding two recommendation lists that they have the same topic distribution of their topic sets, we think the list with more number of different topics should have higher diversity. Based on this point, our diversity indicate consists of the number of different topics and the distribution of every topic in the recommendation list. $Div(L_u)$ denotes the various numbers for topic distribution of recommendation list $L_u$, and it is calculated as follows,

$$Div(L_u) = \frac{\left|s_{L_u}\right|}{N_t} \cdot H(L_u) \tag{13}$$

where $\left|s_{L_u}\right|$ is the number of different topics in recommendation list $L_u$, and $N_t$ is the number of the kinds of topics in the whole recommendation system.

Actually, although $Div(L_u)$ considers the number of topics and the distribution of each topic in a recommendation list, it is also not good enough to describe the topic diversity for a recommendation list. We give an example to explain its limitation in Fig. 2. For an item, it may contain several topics. There are two recommendation lists, $L1 = [A, B, C]$ and $L2 = [E, F, G]$. If their topic sets are $z_{L1} = [1, 2, 2, 3, 1, 4]$, $s_{L1} = [1, 2, 3, 4]$, $z_{L2} = [1, 2, 3, 4, 2, 1]$, $s_{L1} = [1, 2, 3, 4]$, they will have the same number of topics, and the same distribution of each topic. Therefore, $Div(L_u)$ cannot distinguish the topic diversity between $L1$ and $L2$. Actually, when a user does not like item $E$ in $L2$, the remaining part of $L2$ only contains two topics after item $E$ is removed. It means $L2$ has a sharp falling for the topic diversity. Regarding $L1$, even if a user does not like anyone of the items, the topic diversity of the remaining items changes a little after any item is removed. Therefore, we think the recommendation list $L1$ is better than the recommendation list $L2$. Accordingly, we take the distribution of a topic in every item for the recommendation list into account. If there are two recommendation lists with the same number kinds of topics and the values of topic distribution, we should consider the number of topics that each item includes.

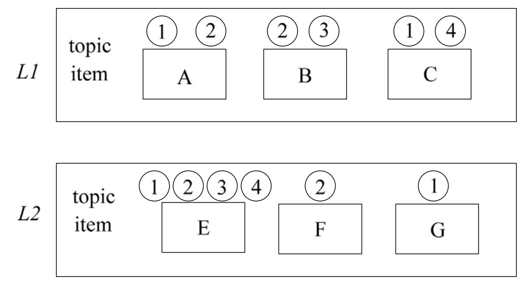



**Fig. 2.** An example about the limitation of $Div(L_u)$.

Based on the above analysis, we take three aspects for our topic diversity into account, including the topic distribution in a recommendation list, the number of different topics in a recommendation list and the distribution of a topic for each item in a recommendation list. Finally, our topic diversity is calculated as follows,

$$diversity(L_u) = -\left(\sum_{i \in L_u} \frac{\left|t_{x_i}\right|}{\left|z_{L_u}\right|} \cdot \log \frac{\left|t_{x_i}\right|}{\left|z_{L_u}\right|}\right) \cdot Div(L_u) \tag{14}$$

where $\left|t_{x_i}\right|$ is the number of topics included in item $x_i$.

Consequently, our objective function of diversity can be represented as follows.

$$D = \frac{\sum_{u \in U} diversity(L_u)}{|U|}. \tag{15}$$

These two objective functions can effectively lead to a tendency for accuracy and diversity in a recommendation list. The objective function of accuracy will make users select items that have high rating. The objective function of diversity will make users select items that have large value of topic diversity. Obviously, there is the conflict between accuracy and diversity. It means an item may do not have high rating and large value of diversity at the same time. Therefore, we utilize the multi-objective optimization algorithm to solve this conflict. The design of the objective function is described as follows.

$$\begin{cases} \max f_1 = PR \\ \max f_2 = D. \end{cases} \tag{16}$$

### 3.3. Design of proposed algorithm PMOEA

Generally, a typical multi-objective evolutionary algorithm consists of the following steps.

Step 1: The setup of parameters and initialization of solution.

Step 2: The calculation of the fitness value.

Step 3: The generation of new solutions.

Step 4: The selection of better solution according to the relations of non-domination.

Step 5: The examination of the termination condition. If the loop is not terminated, go to Step 2. Otherwise, Step 6 is executed at the end of the cycle.

Step 6: The output of the final solution and it is the last step of this algorithm.

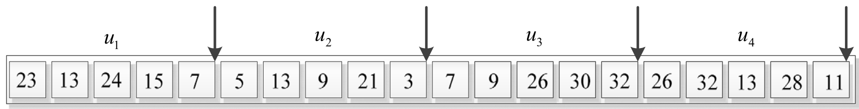



**Fig. 3.** The structure of a solution with chromosome encoding.

Actually, there are some intelligent algorithms that can be used to generate new solutions for the multi-objective optimization algorithm. The commonly intelligent algorithms include genetic algorithm (GA) [32], particle swarm optimization (PSO) [4], artificial bee colony (ABC) algorithm [6], differential evolution (DE) algorithm [5] and so on. In this paper, we make use of GA to generate new solutions. GA mainly includes the following procedures: population initialization, individual encoding, selection, crossover and mutation, solution update, individual decoding (fitness), termination rules. In the last subsection, we propose two objective functions, which can be used to calculate the fitness value. In this subsection, we design a novel multi-objective evolutionary algorithm. First, we set up the solution space and the way of chromosome encoding, and initialize the solution. Second, we introduce a method to generate the new solutions. According to the relations of non-domination, we select the good solution. We consider the features of the recommendation for the multi-objective optimization algorithm, and we propose a probabilistic multi-objective evolutionary algorithm, called PMOEA.

### 3.3.1. Representation and initialization

The most common methods of individual encoding include binary code, natural number coding, gray code coding, floating point numbers (real) encoding and so on. In the recommendation algorithms, the item can be described by the natural number according to its item ID. Therefore, our algorithm also makes use of the natural number coding to solution space as the description in [1]. In this case, according to the value of the solution, we can quickly find out the corresponding items. A solution represents a chromosome in GA. In other words, each chromosome represents a recommendation result for all users. We use $X = \{x_1, x_2 \cdots x_n\}$ to depict a chromosome and $x_i$ is the gene in the chromosome. The length of the chromosome is equal to the number of users multiplied by the length of a recommendation list. Therefore, each solution includes several recommendation lists, each of which is recommended to a user. Each gene represents an item and thus an item can only appear once in the recommendation list for a user. To more clearly explain the features of chromosome encoding, we give an example of the structure of a solution with chromosome encoding in Fig. 3. As shown in Fig. 3, this solution consists of 4 recommendation lists, each of which is used for a user. For instance, item 23, item 13, item 24, item 15 and item 7 would be recommended to user $u_1$.

When the chromosome coding of the solution is finished, we enter the process of solution initialization. We take the initialization process into two parts. Only one solution is initialized in the first part and other solutions are initialized in the second part. With regard to the solution in the first part, we select top-$n$ items according to the evaluated rating to initialize the recommendation list in the solution. For the initialization of other solutions in the second part, we randomly select different items in the recommendation list to initialize the solutions. It is worth noting that the initialization, and crossover are operated for each corresponding recommendation list in the solutions.

### 3.3.2. Genetic operator

After the determination of the initialization and encoding, we design a new genetic operator. Genetic operator plays an important role in the algorithm of GA, since the new solutions are generated by genetic operator. It contains the crossover operator and mutation operator.

A. Crossover operator

Crossover operator is used to inherit gene to their offspring. The traditional crossover operators include single-point crossover, two-point crossover, multi-point crossover, fusion crossover, uniform crossover and so on. However, these crossover operators do not consider the features of users in a recommendation system. Therefore, we propose a new crossover operator which takes the habits of a user into account. If an item is recommended to a user for several times repeatedly, this user may pay more attention to the item. So, we can compute the frequency of each item appearing in several corresponding recommendation lists. And we utilize the ratio of the current item's frequency to the maximal frequency of items as the probability that current item appears in the new recommendation lists. This probability can better describe a possibility that the user will like the item.

Regarding to the genetic operator of our multi-objective evolutionary algorithm, we calculate the probability of the gene appearing in several parent solutions. And then, we utilize this probability to estimate whether the gene is inherited to the offspring solution. This generation method of the new solution is called the multi-parent probability genetic operator.

For the multi-parent probability genetic operator, we randomly select several original solutions as the parent solutions, and we count the number of each gene appearing in the parent solutions. Then, we could calculate the probability of each gene appearing in the several parent solutions. $K = (k_1, k_2, \ldots, k_m)$ is the set of the times that all the genes appear in the parent solutions. $k_i$ is the number of times that gene $i$ appears in the parent solutions, $i \in (1 \cdots m)$. $m$ is the number of different genes in the parent solutions. The probability of gene $i$ appearing in the parent solutions is $p_i$. The computing formula of $p_i$ is described as follows.

$$p_i = \frac{k_i - 1}{\max(K)}. \tag{17}$$

As we calculate the probability in the parent solutions, we utilize the probability to ensure the gene is inherited in the new solution. When the length of a new solution is less than the length of the solution space, we can randomly select some genes that are not selected in the parent solutions, and add these selected genes to the new solution. Fig. 4 illustrates an example of our proposed multi-parent probability genetic operator. In Fig. 4, three original solutions Parent 1, Parent 2 and Parent 3 are randomly selected as the parent solutions. There are seven different genes in those original solutions. We can calculate the probability of each gene appearing in these original solutions. The probabilities of gene 2, gene 5, and gene 1 are 2/3, 1/3 and 1/3, respectively. For gene 2, we generate a random number $\beta \in 0, 1$. Without loss of generalization, we let $\beta = 0.6$. If $\beta$ is smaller than $p_2$, the new solution will accept the gene 2. Otherwise, gene 2 is added in the candidate set. The value of $p_2$ can be computed as Eq. (17), namely $p_2 = 2/3 = 0.667 > \beta$. Therefore, the new solution will accept the gene 2. For gene 5 and the gene 1, the same calculation and determination process would be conducted. In this case, both gene 2 and gene 1 are included in the Child. Gene 5 is in the candidate set. Since the probabilities of gene 3, gene 4, gene 6 and gene 8 are 0, these genes are also added in the candidate set. Moreover, some genes in the candidate set are randomly selected

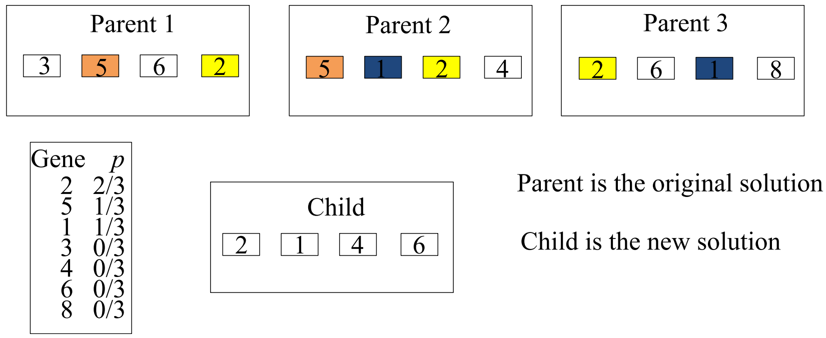



**Fig. 4.** The multi-parent probability genetic operator.

to add in the Child, because the length of the Child is less than the recommendation list.

B. Mutation operator

Regarding to the mutation operator, PMOEA utilizes the same standard single-point mutation operator, like other multi-objective optimization algorithms. For each solution, we randomly select a gene, and generate a random number. If the generated random number is larger than 0.5, we will give the gene a mutation operator. Otherwise, the gene remains unchanged. Through the mutation operator, the mutational gene must be different with the genes in the same recommendation list.

## 4. Experiments

### 4.1. Experimental data

The data set Movielens is used to evaluate the performance and effectiveness of PMOEA, which is a famous publicly available data set for the studies on recommendation. This data set consists of 100 000 ratings (1–5) from 943 users on 1682 movies and each user has rated at least 20 movies. There are 462 users in the test data set. The topic information contains at least 19 categories.

### 4.2. Parameter settings

There are many parameters in PMOEA and MOEA, and the main parameters in the experiments are shown in Table 1.

**Table 1**
The parameter settings of PMOEA and MOEA.

| Parameter | Meaning | Value |
|---|---|---|
| $L$ | The length of the recommendation list | 10 |
| $NP$ | The number of solutions | 100 |
| $pn$ | The number of parents to crossover | 5 |
| $pm$ | The probability for mutation | 0.5 |
| $gmax$ | The number of iterations | 500 |
| $n$ | The number of neighbors | 20 |

### 4.3. Evaluation metrics

To evaluate the quality of a recommendation algorithm, lots of metrics are introduced and utilized. To evaluate the performance and effectiveness of PMOEA, we use precision, diversity and novelty as the evaluation metrics in the experiments.

Precision is to measure the accuracy of the recommendation results. If a user is more satisfied with a recommendation result, this recommendation algorithm has greater precision.

Precision denotes the proportion of accepted recommendation items from the total number of the items in the whole recommendation list. In our experiments, precision is defined as follows,

$$precision = \frac{1}{|U|} \cdot \sum_{u \in U} \frac{|L_u \cap T_u|}{L} \tag{18}$$

where $U$ is the user set in test data set and $L$ is the length of recommendation list. $|U|$ is the number of users in the test data set. $L_u$ is the recommendation list for user $u$, and $T_u$ is the set of items for user $u$ that user $u$ has rated high rating in test data set.

The diversity of a recommendation list is used to indicate the difference between items in recommendation list. Moreover, the best way to describe the difference between items is the various kinds of items.

Diversity includes the topic distribution in recommendation list, the number of different topics in the recommendation list and the distribution of a topic for each item in the recommendation list. As mentioned above, diversity is defined as follows

$$diversity(L_u) = -\left(\sum_{i \in L_u} \frac{|t_{x_i}|}{|z_{L_u}|} \cdot \log \frac{|t_{x_i}|}{|z_{L_u}|}\right) \cdot Div(L_u). \tag{19}$$

Novelty refers to the popularity of the items in a recommendation list. If the items of a recommendation list are more popular, the novelty of this recommendation list is lower.

Novelty denotes the proportion of users from the number of rated item. According to [39], the novelty can be defined as follows,

$$novelty = \frac{1}{M \cdot L} \sum_{u=1}^{M} \sum_{\alpha \in L_u} \log_2 \left(\frac{M}{N_\alpha}\right) \tag{20}$$

where $M$ is the total number of users in the recommendation system, and $L$ is the length of the recommendation list. $N_\alpha$ is the number for rated of item $\alpha$.

### 4.4. Comparison algorithms

In order to validate the performance of PMOEA, we combine MOEA and PMOEA with three traditional recommendation algorithms, including ProbS [36], CF_User, CF_Item [25] and we get the multi-objective optimization recommendation algorithms, including MOEA + ProbS [39], MOEA + CF_User, MOEA + CF_Item, PMOEA + ProbS, PMOEA + CF_User and PMOEA + CF_Item. Our experiments are conducted on these comparison algorithms.

ProbS method [36] utilizes bipartite network to evaluate item rating. It uses a user's rating of known items to predict the user's ratings of the unknown items. In the experiments, we select the top-$n$ higher rating items to recommend them for the user.

User-based Collaborative Filtering (CF_User in short) algorithm is used to get the idea of homogeneity between users. That is to say, similar users have the same behaviors or interests. Based on the user's preferences for items, we find some similarity neighbors, and then predict the item ratings for those users according to the found neighbors. We select the top-$n$ higher rating items to recommend them for the user.

Item-based Collaborative Filtering (CF_Item in short) algorithm is similar to CF_User algorithm. The only difference is that CF_Item algorithm calculates the similarity between items. It utilizes user's preferences on items to find similar items. Then, according to the preferences of the user's history, it evaluates the unknown item's rating to user. We select the top-$n$ higher rating items to recommend them for the user.

MOEA + ProbS uses the ProbS method to evaluate the item rating. Then, the standard multi-objective optimize algorithm (MOEA) is used to select the recommendation list for the user.

MOEA + CF_User uses CF_User algorithm to evaluate the item rating. Then, the standard multi-objective optimize algorithm (MOEA) is used to select the recommendation list for the user.

MOEA + CF_Item uses CF_Item algorithm to evaluate the item rating. Then, the standard multi-objective optimize algorithm (MOEA) is used to select the recommendation list for the user.

PMOEA + ProbS uses ProbS method to evaluate the item rating. Then, our probabilistic multi-objective optimize algorithm (PMOEA) is used to select the recommendation list for the user.

PMOEA + CF_User uses CF_User algorithm to evaluate the item rating. Then, our probabilistic multi-objective optimize algorithm (PMOEA) is used to select the recommendation list for the user.

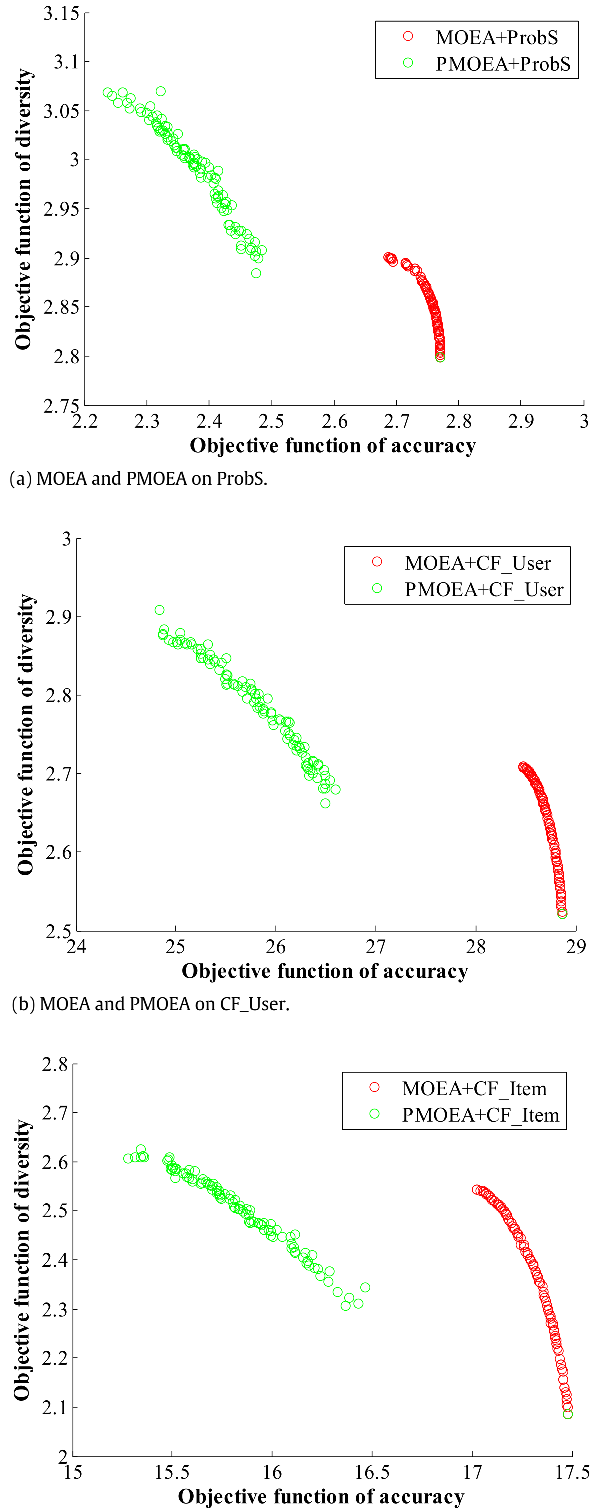
PMOEA + CF_Item uses CF_Item algorithm to evaluate the item rating. Then, our probabilistic multi-objective optimize algorithm (PMOEA) is used to select the recommendation list for the user.

### 4.5. Experimental results

In this subsection, we first analyze the Pareto frontier of MOEA and PMOEA combined with ProbS, CF-User, and CF-Item. Next, we will compare the precision, the diversity and the novelty for the first 10 users in the test data on different algorithms [9,27]. Finally, we analyze the final average results of each algorithm on all users. To obtain statistical results, 30 independent runs are performed for the data set.

Fig. 5 describes the Pareto frontier for the final solutions obtained by several algorithms. Each point represents a pair of fitness values. In each sub-figure, there are one hundred points for each algorithm. MOEA and PMOEA are combined with the three recommendation algorithms (ProbS, CF_User, and CF_Item) respectively, and the results are shown in Fig. 5(a), (b) and (c), respectively. From the comparison results in Fig. 5, we can see that PMOEA can cover more solution space than MOEA on three recommendation algorithms. Therefore, we can say that PMOEA is better than MOEA. For the objective function of accuracy, the solution of MOEA is better than that of PMOEA, while for the objective function of diversity, the solution of PMOEA is better than that of MOEA. Overall, PMOEA can get better balance between precision and diversity than MOEA. In addition, in terms of the objective function of accuracy, the solutions of Fig. 5(a) are significantly smaller than those of Fig. 5(b) and (c). The reason is that ProbS method is an implicit rating of prediction, and the value of item's rating is predicted between 0 to 1. However, CF_User and CF_Item are an obvious item evaluation, and the rating results are in [0, 5].

It is well known that most multi-objective evolutionary recommendation algorithms aim to achieve a balance between precision and diversity. Table 2 describes the evaluation results of these compared algorithms in terms of precision. In Tables 2–4, the term 'mean' means the average value of the metric for all recommendation lists. The term 'min' means the minimum value of the metric for all recommendation lists. The term 'max' means the maximum value of the metric for all recommendation lists. We can see that for most users, the precision value of PMOEA + ProbS,



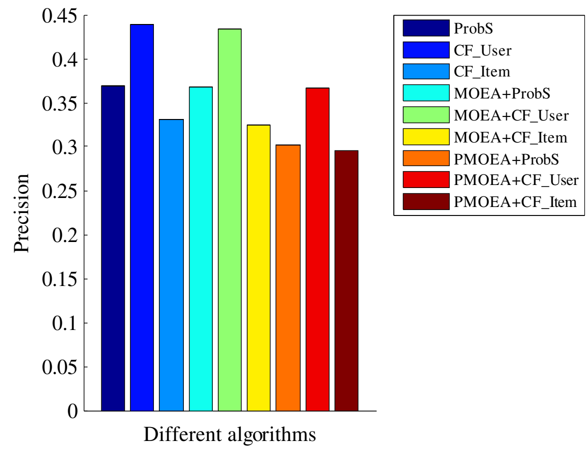

(a) MOEA and PMOEA on ProbS.

(b) MOEA and PMOEA on CF_User.

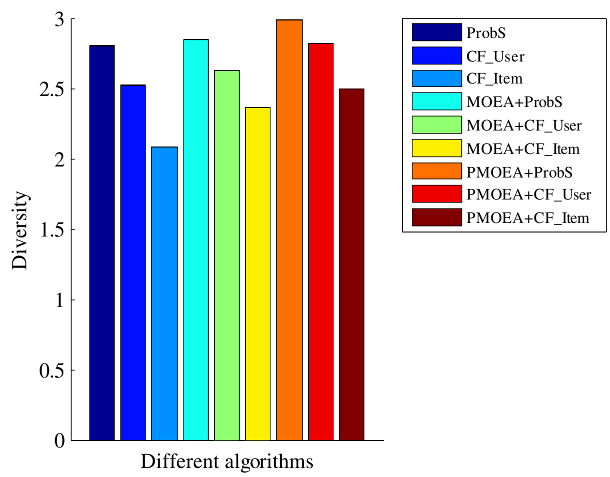
(c) MOEA and PMOEA on CF_Item.

**Fig. 5.** Pareto front of different algorithms.

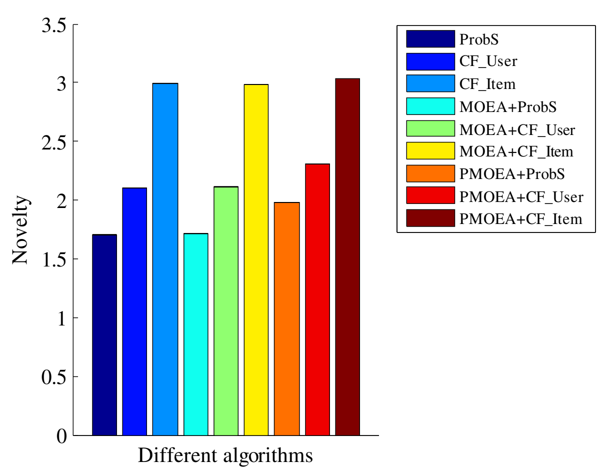
PMOEA + CF_User and PMOEA + CF_Item is smaller than that of the corresponding original algorithms. However, for user 3 and user 7, the mean precision value of PMOEA + CF_User algorithm is higher than that of CF_User algorithm. For user 2, user 3, user 4, user 7 and user 8, the mean precision of PMOEA + CF_Item algorithm is larger than that of CF_Item algorithm. Moreover, the




(a) Precision of different algorithms.




(b) Diversity of different algorithms.




(c) Novelty of different algorithms.

**Fig. 6.** The average precision, diversity and novelty of different algorithms.

maximum precision value of PMOEA + ProbS, PMOEA + CF_User and PMOEA + CF_Item is larger than that of the corresponding original algorithm for most parts of ten users. Therefore, PMOEA is effective to combine with the recommendation algorithm for a fraction of users in terms of precision.

In Table 3, we evaluate the diversity of the compared algorithms. For most users, PMOEA + ProbS, PMOEA + CF_User and PMOEA + CF_Item outperform the corresponding original algorithms. However, in a few cases, the original algorithm is better. For user 4, user 7 and user 10, the mean diversity of PMOEA+ ProbS algorithm is smaller than that of ProbS algorithm. For user 1, user 2, user 3, user 4, and user 7, the mean diversity of PMOEA + CF_User algorithm is smaller than that of CF_User algorithm. But they are very close to each other. For all ten users, the mean diversity of PMOEA + CF_Item is larger than that of the CF_Item algorithm. On the whole, the combination of PMOEA and the recommendation algorithm has better optimization effect on diversity.

Table 4 shows the performance of these compared algorithms on the metric of novelty. Compared with corresponding original algorithms, PMOEA + ProbS and PMOEA + CF_User have obvious advantages. Although the novelty of PMOEA + CF_Item algorithm is less than that of CF_Item for five users, the maximum novelty value of PMOEA_CF_Item algorithm is larger than that of CF_Item algorithm for all ten users. Moreover, the minimum novelty value of PMOEA + ProbS algorithm is equal to that of ProbS on user 1, user 6, user 7, user 8, and user 9. Therefore, the combination of PMOEA with the recommendation algorithm can work well in terms of novelty.

Fig. 6 describes average precision, diversity, novelty of the compared algorithms on all users of the test set. Regarding to precision in Fig. 6(a), the combination of MOEA/PMOEA and the recommendation algorithm cannot be efficient. However, with respect to diversity and novelty in Fig. 6(b) and (c), the combination of PMOEA and the recommendation algorithm is better than the combination of MOEA and the recommendation algorithm. Overall, the combination of PMOEA and the recommendation algorithm can achieve better diversity and novelty by sacrificing a certain degree of precision.

## 5. Conclusion

In this paper, we study how to improve the performance and effectiveness of recommendation algorithm by making use of the multi-objective optimization algorithm. We introduce two objective functions, and put forward a new indicator of topic diversity. It is mainly used to measure the ability of the recommendation algorithm to recommend different topic types of items. And then, we propose a probabilistic multi-objective optimization algorithm (called PMOEA). In PMOEA, we present a new genetic operator. Finally, through the comparison with some known recommendation algorithms, the experimental results show that our PMOEA algorithm can improve the effectiveness, especially in terms of diversity and novelty.

## Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grants 61472258, 61402294, and 61572328. Guangdong Natural Science Foundation under Grant S2013040012895, Foundation for Distinguished Young Talents in Higher Education of Guangdong, China under Grant 2013LYM_0076, Major Fundamental Research Project in the Science and Technology Plan of Shenzhen under Grants JCYJ20140509172609162, JCYJ20140828163633977, JCYJ20140418181958501, JCYJ20150630105452814, JCYJ20160310095523765 and JCYJ20160307111232895. The Open Research Fund of China-UK Visual Information Processing Lab.

**Table 2**
Precision of ten sample users.

| User-ID | ProbS | CF_User | CF_Item | PMOEA + ProbS | | | PMOEA + CF_User | | | PMOEA + CF_Item | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Mean | Min | Max | Mean | Min | Max | Mean | Min | Max |
| 1 | 0.7 | 0.9 | 0.6 | 0.690 | 0.500 | 0.900 | 0.767 | 0.600 | 1 | 0.556 | 0.200 | 0.800 |
| 2 | 0.3 | 0.3 | 0.2 | 0.235 | 0 | 0.300 | 0.246 | 0 | 0.400 | 0.207 | 0 | 0.400 |
| 3 | 0.2 | 0.2 | 0.2 | 0.173 | 0 | 0.300 | 0.211 | 0.100 | 0.400 | 0.216 | 0 | 0.400 |
| 4 | 0.3 | 0.2 | 0 | 0.273 | 0.100 | 0.400 | 0.155 | 0 | 0.200 | 0.023 | 0 | 0.100 |
| 5 | 0.6 | 0.7 | 0.7 | 0.481 | 0.300 | 0.700 | 0.634 | 0.400 | 0.900 | 0.587 | 0.300 | 0.800 |
| 6 | 0.6 | 0.7 | 0.5 | 0.545 | 0.300 | 0.800 | 0.579 | 0.400 | 0.700 | 0.491 | 0.300 | 0.800 |
| 7 | 1 | 0.8 | 0.6 | 0.799 | 0.600 | 1 | 0.912 | 0.800 | 1 | 0.653 | 0.400 | 0.900 |
| 8 | 0.7 | 0.8 | 0.2 | 0.509 | 0.300 | 0.700 | 0.600 | 0.400 | 0.800 | 0.290 | 0.100 | 0.500 |
| 9 | 0.1 | 0.1 | 0 | 0.081 | 0 | 0.100 | 0.071 | 0 | 0.100 | 0 | 0 | 0 |
| 10 | 0.5 | 0.9 | 0.7 | 0.397 | 0.200 | 0.600 | 0.822 | 0.600 | 1 | 0.658 | 0.400 | 0.900 |

**Table 3**
Diversity of ten sample users.

| User-ID | ProbS | CF_User | CF_Item | PMOEA + ProbS | | | PMOEA + CF_User | | | PMOEA + CF_Item | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Mean | Min | Max | Mean | Min | Max | Mean | Min | Max |
| 1 | 1.880 | 3.193 | 2.226 | 2.531 | 1.797 | 3.809 | 3.098 | 2.090 | 4.232 | 3.200 | 1.828 | 4.178 |
| 2 | 2.131 | 2.042 | 1.068 | 2.256 | 1.786 | 2.848 | 2.016 | 1.108 | 2.873 | 1.347 | 0.510 | 2.419 |
| 3 | 2.912 | 2.110 | 1.709 | 3.307 | 2.177 | 4.494 | 2.028 | 1.358 | 2.545 | 1.852 | 1.031 | 2.846 |
| 4 | 3.211 | 2.837 | 3.220 | 3.175 | 1.519 | 4.048 | 2.805 | 2.102 | 3.622 | 3.256 | 2.429 | 4.126 |
| 5 | 2.571 | 2.742 | 2.826 | 2.903 | 1.806 | 3.770 | 2.987 | 1.797 | 4.528 | 3.011 | 2.133 | 4.164 |
| 6 | 2.150 | 2.427 | 1.795 | 3.002 | 2.125 | 3.642 | 3.056 | 2.051 | 4.551 | 2.657 | 1.427 | 3.823 |
| 7 | 2.799 | 3.156 | 1.335 | 2.765 | 1.779 | 3.532 | 2.916 | 2.151 | 3.568 | 1.906 | 0.666 | 3.647 |
| 8 | 2.875 | 2.099 | 1.414 | 3.055 | 2.184 | 3.754 | 3.053 | 1.735 | 3.986 | 2.253 | 1.414 | 3.110 |
| 9 | 3.334 | 2.908 | 1.729 | 3.399 | 2.542 | 4.240 | 2.917 | 1.796 | 3.784 | 2.667 | 1.529 | 3.656 |
| 10 | 2.940 | 2.996 | 2.111 | 2.857 | 1.724 | 3.703 | 3.450 | 2.597 | 4.645 | 2.117 | 1.352 | 2.964 |

**Table 4**
Novelty of ten sample users.

| User-ID | ProbS | CF_User | CF_Item | PMOEA + ProbS | | | PMOEA + CF_User | | | PMOEA + CF_Item | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | Mean | Min | Max | Mean | Min | Max | Mean | Min | Max |
| 1 | 1.684 | 1.918 | 2.306 | 1.915 | 1.684 | 2.252 | 2.087 | 1.795 | 2.394 | 2.618 | 2.292 | 3.240 |
| 2 | 1.798 | 2.383 | 3.175 | 1.989 | 1.761 | 2.411 | 2.616 | 2.236 | 2.973 | 3.193 | 2.812 | 3.679 |
| 3 | 1.876 | 2.454 | 3.536 | 2.100 | 1.838 | 2.675 | 2.587 | 2.259 | 2.934 | 3.483 | 3.081 | 4.014 |
| 4 | 1.913 | 2.271 | 2.995 | 2.111 | 1.891 | 2.441 | 2.424 | 2.151 | 3.038 | 3.175 | 2.631 | 4.008 |
| 5 | 1.692 | 2.058 | 2.791 | 1.873 | 1.669 | 2.148 | 2.210 | 1.981 | 2.459 | 2.759 | 2.341 | 3.188 |
| 6 | 1.672 | 2.264 | 3.144 | 1.943 | 1.672 | 2.289 | 2.347 | 2.114 | 2.751 | 3.001 | 2.410 | 3.511 |
| 7 | 1.700 | 1.985 | 3.514 | 1.999 | 1.700 | 2.525 | 2.208 | 1.985 | 2.669 | 3.503 | 3.151 | 3.852 |
| 8 | 1.576 | 2.074 | 3.395 | 1.956 | 1.576 | 2.739 | 2.304 | 2.032 | 2.648 | 3.242 | 2.727 | 3.754 |
| 9 | 1.499 | 1.633 | 2.318 | 1.778 | 1.499 | 2.199 | 1.780 | 1.633 | 2.038 | 2.489 | 2.126 | 2.899 |
| 10 | 1.601 | 2.393 | 3.517 | 1.901 | 1.601 | 2.177 | 2.509 | 2.324 | 2.907 | 3.581 | 3.288 | 3.873 |

## References


[1] F. Belém, R. Santo, J. Almeida, M. Gonçalves, Topic diversity in tag recommendation, in: Proc. of ACM Conference on Recommender Systems 2013, pp. 141–148.

[2] J. Bobadilla, F. Ortega, A. Hernando, A. Gutiérrez, Recommender systems survey, Knowl.-Based Syst. 46 (2013) 109–132.

[3] P. Castells, N.J. Hurley, S. Vargas, Novelty and Diversity in Recommender Systems, Recommender Systems Handbook, Springer, US, 2015, pp. 881–918.

[4] C.A.C. Coello, M.S. Lechuga, MOPSO: A proposal for multiple objective particle swarm optimization, in: Proc. of Congress on Evolutionary Computation 2002, pp. 1051–1056.

[5] L. Cui, G. Li, Q. Lin, J. Chen, N. Lu, Adaptive differential evolution algorithm with novel mutation strategies in multiple sub-populations, Comput. Oper. Res. 67 (2016) 155–173.

[6] L. Cui, G. Li, Q. Lin, Z. Du, W. Gao, J. Chen, N. Lu, A novel artificial bee colony algorithm with depth-first search framework and elite-guided search equation, Inform. Sci. 367–368 (2016) 1012–1044.

[7] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evol. Comput. 6 (2) (2002) 182–197.

[8] J. Fan, W. Pan, L. Jiang, An improved collaborative filtering algorithm combining content-based algorithm and user activity, in: Proc. of IEEE International Conference on Big Data and Smart Computing 2014, pp. 88–91.

[9] B. Geng, L. Li, L. Jiao, M. Gong, Q. Cai, Y. Wu, NNIA-RS: A multi-objective optimization based recommender system, Physica A 424 (2015) 383–397.

[10] J. Golbeck, FilmTrust: movie recommendations from semantic web-based social networks, in: Proc. of Consumer Communications and Networking Conference 2006, pp. 1314–1315.

[11] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.

[12] U. Kuter, J. Golbeck, SUNNY: a new algorithm for trust inference in social networks using probabilistic confidence models, in: Proc. of AAAI 2007, pp. 1377–1382.

[13] N. Lathia, S. Hailes, L. Capra, X. Amatriain, Temporal diversity in recommender systems, in: Proc. of the 33rd International ACM SIGIR conference on research and development in information retrieval 2010, pp. 210–217.

[14] J. Li, M. Qiu, Z. Ming, G. Quan, X. Qin, Z. Gu, Online optimization for scheduling preemptable tasks on IaaS cloud systems, J. Parallel Distrib. Comput. 72 (4) (2012) 666–677.

[15] H. Liu, Z. Hu, A. Mian, H. Tian, X. Zhu, A new user similarity model to improve the accuracy of collaborative filtering, Knowl. Based Syst. 56 (2014) 156–166.

[16] W. Ma, X. Feng, S. Wang, M. Gong, Personalized recommendation based on heat bidirectional transfer, Physical A 444 (2016) 713–721.

[17] Sean M. McNee, John Riedl, Joseph A. Konstan, Accurate is not always good: How accuracy metrics have hurt recommender system, in: Extended abstracts on Human factors in computing systems, 2006, pp. 1097–1101.

[18] J. Niu, Y. Gao, M. Qiu, Z. Ming, Selecting proper wireless network interfaces for user experience enhancement with guaranteed probability, J. Parallel Distrib. Comput. 72 (10) (2012) 1565–1575.

[19] J. O'Donovan, B. Smyth, Trust in recommender systems, in: Proc. of the 10th International Conference on Intelligent User Interfaces 2005, pp. 167–174.

[20] U. Panniello, A. Tuzhilin, M. Gorgoglione, Comparing context-aware recommender systems in terms of accuracy and diversity, User Model. User-Adapt. Interact. 24 (1) (2014) 35–65.

[21] M. Qiu, Z. Ming, J. Li, K. Gai, Z. Zong, Phase-change memory optimization for green cloud with genetic algorithm, IEEE Trans. Comput. 64 (10) (2015) 3528–3540.

[22] M.T. Ribeiro, A. Lacerda, A. Veloso, N. Ziviani, Pareto-efficient hybridization for multi-objective recommender systems, in: Proc. of ACM conference on Recommender systems 2012, pp. 19–26.

[23] M.T. Ribeiro, N. Ziviani, E.S.D. Moura, I. Hata, A. Lacerdaet, A. Veloso, Multi objective pareto-efficient approaches for recommender systems, ACM Trans. Intell. Syst. Technol. (TIST) 5 (4) (2015) 1–20.
[24] F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook, in: Recommender Systems Handbook, Springer, US, 2010, pp. 1–35.
[25] B. Sarwar, G. Karypis, J. Konstan, J. Riedl, Item-based collaborative filtering recommendation algorithms, in: Proc. of ACM WWW 2001, pp. 285–295.
[26] S. Wang, M. Gong, H. Li, J. Yang, Multi-objective optimization for long tail recommendation, Knowl.-Based Syst. 104 (2016) 145–155.
[27] S. Wang, M. Gong, L. Ma, Decomposition based multiobjective evolutionary algorithm for collaborative filtering recommender systems, in: Proc. of IEEE Congress on Evolutionary Computation, CEC 2014, pp. 672–679.
[28] X. Wang, L. Sun, Z. Wang, D. Meng, Group recommendation using external followee for social TV, in: Proc. of IEEE ICME, 2012, pp. 37–42.
[29] Z. Wang, L. Sun, W. Zhu, S. Yang, H. Li, D. Wu, Joint social and content recommendation for user-generated videos in online social network, IEEE Trans. Multimedia 15 (3) (2013) 698–709.
[30] Y. Wang, G. Yina, Z. Cai, Y. Dong, H. Donga, A trust-based probabilistic recommendation model for social networks, J. Netw. Comput. Appl. 55 (2015) 59–67.
[31] S. Wanita, S. Nepal, C. Paris, A survey of trust in social networks, ACM Comput. Surv. 45 (3) (2013) 47–33.
[32] D. Whitley, A genetic algorithm tutorial, Stat. Comput. 4 (2) (1994) 65–85.
[33] T. Yuan, J. Cheng, X. Zhang, Q. Liu, H. Lu, How friends affect user behaviors? An exploration of social relation analysis for recommendation, Knowl.-Based Syst. 88 (2015) 70–84.
[34] M. Zhang, N. Hurley, Avoiding monotony: improving the diversity of recommendation lists, in: Proc. of ACM Conference on Recommender Systems 2008, pp. 123–130.
[35] Q. Zhang, H. Li, MOEA/D: A multiobjective evolutionary algorithm based on decomposition, IEEE Trans. Evol. Comput. 11 (6) (2007) 712–731.
[36] T. Zhou, J. Ren, M. Medo, Y. Zhang, Bipartite network projection and personal recommendation, Phys. Rev. E 76 (4) (2007) 1–7.
[37] C.N. Ziegler, S.M. McNee, J.A. Konstan, G. Lausen, Improving recommendation lists through topic diversification, in: Proc. of WWW 2005, pp. 22–32.
[38] E. Zitzler, M. Laumanns, L. Thiele, SPEA2: Improving the Strength Pareto Evolutionary Algorithm, Techn. Rep. TIK-Report 103, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.
[39] Y. Zuo, M. Gong, J. Zeng, Personalized recommendation based on evolutionary multi-objective optimization, IEEE Comput. Intell. Mag. 10 (1) (2015) 52–62.

**Laizhong Cui** received the B.S. degree from Jilin University, Changchun, China, in 2007 and Ph.D. degree in computer science and technology from Tsinghua University, Beijing, China, in 2012. He is currently a lecturer in the College of Computer Science and Software Engineering at Shenzhen University, China. He led a project of the National Natural Science Foundation, and several projects of Guangdong Province and Shenzhen City. His research interests include recommendation system, content distribution, cloud computing, software-defined network and social network.

**Peng Ou** is working toward the M.Sc. degree at College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. His research interests include recommendation system, data mining, social network, and Multi-Objective Optimization.

**Xianghua Fu** received the M.Sc. degree from the Northwest A&F University, Yangling, China, in 2002 and Ph.D. degree in computer science and technology from Xi'an Jiaotong University, Xi'an, China, in 2005. He is a professor and postgraduate director at College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, China. He led a project of the National Natural Science Foundation, and several projects of the Science and Technology Foundation of Shenzhen City. His research interests include machine learning, data mining, information retrieval, and natural language processing.

**Zhenkun Wen** received the M.E. degree in computer science and engineering from Tsinghua University, Beijing, China, in 1999. He is currently a Professor in College of Computer Science and Software Engineering at Shenzhen University, Guangdong, China. His research interests include computer networks, machine learning, video information security, etc.

**Nan Lu** is a professor in the College of Computer Science and Software Engineering at Shenzhen University, China. He received the Ph.D. degree from Jilin University, Changchun, China. He led several projects of Guangdong Province and Shenzhen City. His research interests include commerce intelligence, machine learning, complex network community structure, and trust mining of social network.