2014 IEEE Congress on Evolutionary Computation (CEC)
July 6-11, 2014, Beijing, China

Decomposition Based Multiobjective Evolutionary Algorithm for Collaborative Filtering Recommender Systems

Shanfeng Wang, Maoguo Gong, *Member, IEEE*, Lijia Ma, Qing Cai, Licheng Jiao, *Senior Member, IEEE*

***Abstract*— With the rapid expansion of the information on the Internet, recommender systems play an important role in filtering insignificant information and recommend satisfactory items to users. Accurately predicting the preference of users is the first priority of recommendation. Diversity is also an important objective in recommendation, which is achieved by recommending items from the so-called long tail of goods. Traditional recommendation techniques lay more emphasis on accuracy and overlook diversity. Simultaneously optimizing the accuracy and diversity is a multiobjective optimization problem, in which the two objectives are contradictory. In this paper, a multiobjective evolutionary algorithm based on decomposition is proposed for recommendation, which maximizes the predicted score and the popularity of items simultaneously. This algorithm returns lots of non-dominated solutions and each solution is a trade-off between the accuracy and diversity. The experiment shows that our algorithm can provide a series of recommendation results with different precision and diversity to a user.**

I. Introduction

RECOMMENDER systems are tools or techniques to filter the abundant information and recommend users items that are interesting and satisfactory to users based on their different types of information. Demographics of users, like age, gender, can be used in some techniques. Some techniques take advantage of the behavior of users (like books read, items collected, web sites visited) [1]. Social information and ratings on items also are utilized to recommend. Recommender systems have been widely used in many online fields like music, books and movies. Amazon.com recommend books by recommender systems [2]. Recommender systems are used to suggest movies at Netflix.com [3]. Recommender systems have received increasing attention from many researchers because of the significance of their application in e-commerce [4].

There are several challenges and tasks in the fields of recommender systems. The major chanllenge is data sparsity. For in e-commence web sites, there are millions of users and items, but users can just reach and rate a few items, so that lots of items are unrated and the available ratings used in recommendation are rare. Then effective recommendation techniques must deal with the problem of sparsity [5]. Another important challenge in recommender system is cold-start [6, 7]. When new users join in the system or new items are loaded in the system, few information can be used to recommend. To overcome cold start problem, hybrid algorithm is employed [8, 9]. In the traditional recommendation techniques, accuracy is the most focused task. And traditional recommender systems are designed to increase the accuracy of the recommendation results [10, 11], which maximize the metrics accuracy and RMSE such that the prediction is satisfactory to the users. While in [12], McNee indicates that accuracy-focused recommendation may not be the best, for it provides exceptionally similar recommendations, which is detrimental to recommender systems. And recommendation techniques are more accurate if popular items gain high prediction [13]. Then accuracy-focused recommendation algorithms are easy to recommend popular items, which are likely less useful to users. To measure the ability of recommender systems to recommend the unpopular items, the evaluation metrics diversity and novelty are introduced [14–16]. Then how to maximize the accuracy, diversity and novelty simultaneously is a challenge, which is a multiobjective optimization problem. To increase the value of one objective may result in the decrease of other objectives, then it is necessary to find solutions that have a tradeoff between these objectives.

Recently, many recommendation algorithms have been proposed to find a tradeoff among accuracy, diversity and novelty. In [17], a hybrid algorithm is proposed by combining the accuracy-focused and diversity-focused algorithms to solve the dilemma of these metrics. In [13], a framework dealing with the ratings and other additional recommendation goals is proposed. Rodriguez *et al* [18] frame the multiple objectives optimization problem in recommender systems, and this model can be incorporated into other model as an additional stage. In [19], Ribeiro *et al* introduce Pareto-efficient approaches for recommender systems, in which accuracy, diversity and novelty are maximized simultaneously. An algorithm based on evolutionary algorithm is proposed, which combines exiting algorithms with different accuracy, diversity and novelty [20].

Multiobjective evolutionary algorithms (MOEAs) deal with the problem with two or more contradictory objectives. The goal of multiobjective evolutionary algorithms is to achieve a set of Pareto optimal solutions which approximate the Pareto-optimal front. Each of those solutions is a trade-off among these contradictory objectives. Recent years, most multiobjective evolutionary algorithms have been proposed, such as decomposition-based MOEA (MOEA/D) [21], NSGA-II [22], SPEA2 [23], PAES [24] and NNIA [25].

In this paper, we propose a recommendation framework

This work was supported by the National Natural Science Foundation of China (Grant no. 61273317), the National Top Youth Talents Program of China, the Specialized Research Fund for the Doctoral Program of Higher Education (Grant no. 20130203110011) and the Fundamental Research Fund for the Central Universities (Grant no. K5051202053).

The authors are with Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education of China, Xidian University, Xi'an, 710071, China. (Email: gong@ieee.org).

based on multiobjective evolutionary algorithm (MOEA/D-RS). In our framework, the accuracy and diversity of the recommendation are considered simultaneously. The evolutionary algorithm that we use is the Multiobjective Evolutionary Algorithm Based on Decomposition (MOEA/D) [21]. Our motivations can be simplified as follows: a framework is proposed that not only takes into account the predicted preference scores that can show the accuracy of the recommendation, but also deals with values that differentiate whether the items are popular or not to show the diversity of recommendation. To find a good trade-off between these two contradictory objectives, multiobjective evolutionary algorithm is used to optimize the problem. In our algorithm, every nondominated solution along the Pareto front represents a kind of recommendation results, such that every user is given lots of different recommendation lists.

The rest of this paper is organized as follows: Section II gives the related knowledge, including the concept of recommendation, the neighborhood-based collaborative filtering and introduction to multiobjective optimization problem (MOP) and MOEA/D. Detailed description of the proposed framework is put forward in Section III. In Section IV, the performance of our algorithm is given. In Section V, we conclude our works.

II. Related work

A. Recommender systems

Recommender systems aim to suggest items that are likely to be appealing to users according to their preferences. In [26], the recommendation is modeled as the target of predicting ratings for users on unrated items. Adomavicius and Tuzhilin [26] formulate the recommendation as follows. Let M be the set of all the users, and let N be the set of various items. F is the utility function to compute the ratings of user m on unrated item n, where $m \in M$ and $n \in N$:

$$F : M \times N \to R \tag{1}$$

where R is a totally ordered set (e.g. non-negative integer of real numbers). In general, there are two kinds of ratings. One is used to measure the extent to which users like the items, which is called as *explicit ratings*. *Explicit ratings* usually consist of integer of real numbers like the five stars in Amazon.com. Another kind of ratings is referred as implicit ratings, which reflects the collection or consumption of users on items. In the latter form, binary ratings are used to reflect whether the user has collected the items.

In recent years, several types of recommendation methods have been proposed [1, 26, 27]. In [28], Burke divides existing recommendation methods into four types on the basis of their source information: *demographic* techniques [29, 30], which figure out recommendation from the demographic profiles of the users, *content-based* (CB) techniques [30, 31], which recommend the items that are similar to what they have rated, *collaborative filtering* (CF) techniques [32], which generate the recommendation based on the ratings of users who have similar preference with the target user, *knowledge-based* techniques [33, 34], which provide recommendation based on the knowledge about assumptions of users. But all the four techniques above have their strengths and weaknesses, respectively, so that in real application, hybrid algorithms which combine two or more of the techniques above are widely used [35, 36]. One important hybrid algorithm is the combination of CB with CF. Some different ways of combination are given in Fig. 1 [1, 26]. In Fig. 1, CB and CF are combined in four ways. The first one is combining the recommendation results of CB and CF. The second one is incorporating some CB features into CF. The third one is constructing a unified model containing CB and CF, and employing the model to recommend. And the last one is incorporating some CF features into CB.

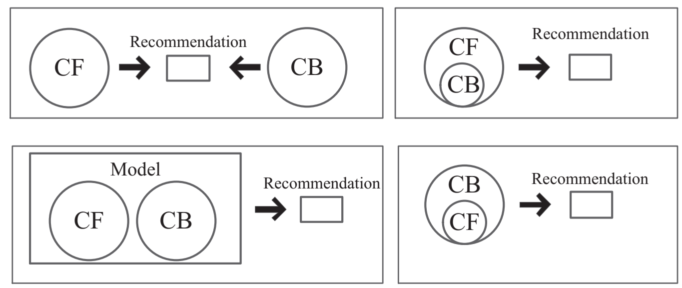


Fig. 1. An example of combination of CB and CF.

Usually, recommender systems give the prediction ratings of users for unknown items-*the rating prediction task*, while in commercial systems, they recommend a few of the best items-*the top-N recommendation task* [32, 37]. The latter also can be considered as prediction ratings, in which we rank all the ratings of items and recommend the top ranked ones.

B. Items-based collaborative filtering

The widely used collaborative filtering technique is *the k Nearest Neighbors (kNN)* recommendation algorithm [1, 38, 39]. The CF based on *kNN* is based on the following steps.

1) Computing items similarity: Similarity is used to measure the extent to which the items or users are similar. In CF, the similarity is computed by the ratings of items co-rated by two users or the ratings of co-users who have rated two items. The former is called user-user similarity, and the latter item-item similarity. There are lots of methods to compute the items similarity. Here, we give some popular methods. The first one is *Pearson correlation*. For the items based CF, the *Pearson correlation* between items i and j is

$$sim(i,j) = \frac{\sum_{u \in U}(R_{u,i} - \bar{R}_i)(R_{u,j} - \bar{R}_j)}{\sqrt{\sum_{u \in U}(R_{u,i} - \bar{R}_i)^2}\sqrt{\sum_{u \in U}(R_{u,j} - \bar{R}_j)^2}} \tag{2}$$

Here, U is the set of users who have both rated item i and j. $R_{u,i}$ and $R_{u,j}$ are ratings of user u on items i and j. $\bar{R}_i$ and $\bar{R}_j$ are the average ratings of items i and j, respectively.

Another popular similarity is *cosine similarity*. The *cosine similarity* between two items i and j is

$$S_{i,j} = \cos(\vec{i}, \vec{j}) = \frac{\vec{i} \bullet \vec{j}}{\|\vec{i}\| * \|\vec{j}\|} \quad (3)$$

where $\vec{i} \bullet \vec{j}$ denote the dot-product of the $\vec{i}$ and $\vec{j}$.

2) Giving the prediction: After getting similarity of all the items, the prediction ratings can be gained by the set of most similar items. And the prediction is given by the following technique, where N is the set of items that are similar to the target item.

$$\hat{r}_{u,i} = b_{u,i} + \frac{\sum_{j \in N} s_{i,j}(r_{u,j} - b_{u,j})}{\sum_{j \in N} |s_{i,j}|} \quad (4)$$

where $b_{u,i}$ is the value of baseline predictor defined as

$$b_{u,i} = \mu + b_u + b_i \quad (5)$$

where μ denotes the overall average rating and parameters b_u and b_i are the observed deviations of user u and item i from the average rating [40].

3) Obtaining the top-k list: For every users, a list consisting of the top ranked k items based on their prediction scores is recommended.

C. Introduction to Multiobjective Optimization

Multiobjective optimization problem can be described as [25],

$$\min F(x) = (f_1(x), f_2(x), \ldots, f_k(x))^T \quad (6)$$

subject to $x = (x_1, x_2, \ldots, x_m) \in \Omega$, where x is the decision vector, Ω is the objective space and k is the number of objectives.

Let x_A and x_B be two decision vectors in objective space Ω. x_A dominates x_B (denoted as $x_A \succ x_B$) iff

$$\begin{aligned} & \forall i = 1, 2, \ldots, k \quad f_i(x_A) \geq f_i(x_B) \\ \wedge \quad & \exists j = 1, 2, \ldots, k \quad f_i(x_A) > f_i(x_B) \end{aligned} \quad (7)$$

A decision vector $x^* \in \Omega$ is called a nondominated solution if there is not another $x \in \Omega$ denominating x^*.

The set of Pareto-optimal set is called the Pareto-optimal set, which is written as

$$P^* \triangleq \{x^* \in \Omega | \neg \exists x \in \Omega, x \succ x^*\} \quad (8)$$

Pareto-optimal front is the corresponding image of the Pareto-optimal set under the objective function space

$$PF^* \triangleq \{F(x^*) = (f_1(x^*), f_2(x^*), \ldots, f_k(x^*))^T | x^* \in P^*\} \quad (9)$$

MOEA aims to find a set of Pareto-optimal solutions that approximate the Pareto-optimal front.

D. Introduction to MOEA/D

In traditional multiobjective evolutionary algorithm, the multiobjective optimization problem is dealt with as a whole, like NSGA-II [22]. MOEA/D is proposed by Zhang and Li [21]. In this algorithm, a multiobjective optimization problem is decomposed into a number of scalar optimization subproblems and these subproblems are optimized simultaneously by evolutionary algorithm. Each subproblem corresponds to an individual solution in the population, and neighborhood relations among these subproblems are defined based on the distances between their aggregation weight vectors. Because of similarities of optimal solutions to two neighborhood subproblems, the current information of neighborhood subproblems are used in the optimization of the subproblem. As indicated in [21], computational complexity at each generation of MOEA/D is lower than that of NSGA-II [22].

There are many decomposition approaches that can decompose the problem of approximation of the Pareto front into a number of scalar optimization subproblems, like weighted sum approach, Tchebycheff approach and penalty-based boundary intersection approach. In our algorithm, Tchebycheff approach is adopted, for recommendation is not a continuous problem. And in the Tchebycheff approach, the scalar optimization problem is:

$$\begin{aligned} min \quad & g^{te}(x|\lambda, z^*) = \max_{1 \leq j \leq m} \{\lambda_j |f_j(x) - z_j^*\} \\ s.t. \quad & x \in \Omega \end{aligned} \quad (10)$$

where, $z^* = (z_1^*, \ldots, z_m^*)^T$ is the reference point, i.e., $z_j^* = min\{f_j(x) | x \in \Omega\}$ for each $j = 1, \ldots, m$. For each Pareto optimal point x^* there is a weight vector λ such that x^* is the optimal solution of (10) and each optimal solution of (10) is a Pareto optimal solution of the MOP (6). And different Pareto optimal solutions can be obtained by tuning the weight vector.

III. The Proposed Algorithm

In this section, we will give a detailed description of the algorithm we proposed named as MOEA/D-RS. First, the framework of MOEA/D-RS is given. And then further explanations of the algorithm are introduced, including the objective functions, initialization and genetic operators and so on.

A. The framework of MOEA/D-RS

The goal of MOEA/D-RS is to give a top-k recommendation list to every user. Our algorithm can mainly be divided into two steps. First, the items-based CF is utilized to generate the score of items that have been unrated, which is used to be an objective function. And a top ranked list with the length of L named as cf-L can be obtained on the basis of ratings, where $L > k$. Subsequently, MOEA/D is used to optimize the MOP. In MOEA/D, the two objectives optimization problem is decomposed into a number of s-calar optimization subproblems. And every solution of these subproblems is a trade-off between accuracy and diversity.

MOEA/D gives many different solutions finally, and every solution represents a recommendation list to the target user.

B. Objective function

The goal of our algorithm is to optimize the accuracy and diversity simultaneously. In our algorithm, the accuracy of recommendation is on the basis of ratings and the diversity on the basis of values that indicate whether the items are popular or not.

The larger the rating of a user on an item is, the more possible the suggestion of the item to the user is. In MOEA/D-RS, the larger the sum of all the ratings in the list, the more accurate the recommendation is. Then, one objective function is defined as:

$$F_1 = \sum_{i=1}^{k} \hat{r}_{u,i} \tag{11}$$

where, u is the target user, $i \in I$ is the item in the recommendation list and k is the length of the list. $\hat{r}_{u,i}$ presents the prediction rating of user u on item i obtained by item-based CF. The summation runs over all the items in the list.

In [13], a function is proposed to measure whether the item is popular or not. It is shown mathematically as:

$$p_i = \frac{1}{\mu_i(\sigma_i + 1)^2} \tag{12}$$

where the value of an item is the reciprocal of the mean (μ_i) and the variance (σ_i). To avoid division by zero, one is added to the variance. The more popular an item is, the less the value is.

To evaluate the popularity of all the items in recommendation list, we sum values of all the items in the list obtained by the approach above as our another objective:

$$F_2 = \sum_{i=1}^{k} \frac{1}{\mu_i(\sigma_i + 1)^2} \tag{13}$$

To let the recommendation be accurate, we maximum the objective function F_1, while to increase the diversity of the recommendation list, the objective function F_2 may be also maximized. In order to formulate recommendation problem as a minimum optimization problem, we revise both of objective functions, and then the problem is described as:

$$\begin{cases} \min \quad F_1 = -\sum_{i=1}^{L} \hat{r}_{u,i} \\ \min \quad F_2 = -\sum_{i=1}^{k} \frac{1}{\mu_i(\sigma_i + 1)^2} \end{cases} \tag{14}$$

C. Representation and initialization

To get the prediction ratings of user on unrated items, the item-based collaborative filtering is adopted. Through the item-based collaborative filtering, all the items unrated can be given a rating. The detailed description of the item-based CF is in Section II, where the *Pearson correlation* is used to compute the item-item similarity. In MOEA/D-RS, we just select the recommendation items from the items in cf-L.

In this algorithm, every chromosome is encoded as a string $X = \{x_1, x_2, \ldots, x_k\}$, where $x_i \in [1, L]$ is an integer that represents the corresponding item in cf-L and k is the length of the final recommendation list. Every chromosome is generated randomly, and every individual in the chromosome is different, for in a list a item could not be recommended more than once. An example of the representation with the length of 10 is given in Fig. 2.

1	2	3	4	5	6	7	8	9	10
6	10	4	20	7	1	23	11	14	16

Fig. 2. An example of individual representation of MOEA/D-RS. The length of the list is 10, and every element is not the same.

D. Genetic operation

1) Crossover: Since the length of the final recommendation list is not very large, in our algorithm, single point crossover in favor of uniform crossover is adopted. Given two parent chromosome X and Y, we randomly select a point i ($1 \le i \le k$), and all the genes at the right of point i between X and Y are swapped (i.e. $X_a \leftrightarrow Y_a$, $i \le a \le k$). But in the process of crossover, some genes may repeat in the child chromosomes, and then a mechanic is necessary to avoid it. In MOEA/D-RS, if there are genes repeating in a child chromosome, the right gene of them is substituted by any item that does not belong to this chromosome. An example of this kind of crossover operation is displayed in Fig. 3.

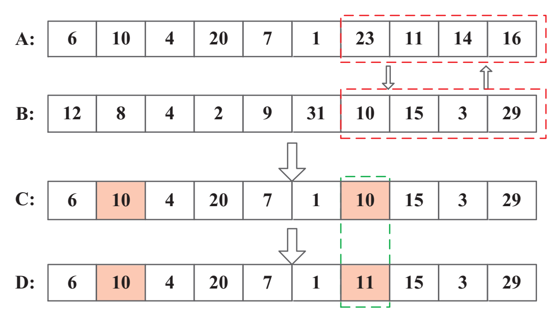


Fig. 3. An example of description of the crossover operation. A and B are the selected parent chromosomes. The length of list is 10, i.e. $k = 10$. The crossover point is the 7th gene. C is the child chromosome of parent chromosome A. In C, the 2th and 7th genes are same, and D is achieved, when 7th gene-10 in C is substituted by 11, where 11 is randomly obtained from items that are not belong to C.

2) Mutation: In this algorithm, one point mutation is adopted. First, a chromosome X is randomly selected to be mutated. Then we randomly select a point i in the chromosome X, and this gene is substituted by any item that does not belong to this chromosome to avoid repeating.

IV. EXPERIMENTS

In this section, we will test the performance of our algorithm. To show the effectiveness of MOEA/D-RS, we compare our algorithm with traditional items-based CF.

A. Experimental settings

There are many parameters in our algorithm, and parameters used in our algorithm are displayed in TABLE I. Besides, all the algorithms are written in MATLAB, and the experiments have been run on a Inter(R) Core(TM) i5 machine with 3.20GHZ CPU and 4.00GB Memory.

TABLE I

THE EXPERIMENTAL PARAMETERS USED IN MOEA/D-RS

Parameters	Meaning	values
L	the length of cf-L	50
k	the length of final recommendation list	10
pop	the size of subproblems	100
gen	the number of generations	200
ns	the size of neighborhood	15
pc	the probability of crossover	0.9
pm	the probability of mutation	0.06
us	the update size	3

B. Evaluation metrics

To show the performance of MOEA/D-RS, we employ some different metrics to measure performance of MOEA/D-RS.

The most widely used metrics to show the accuracy are precision and recall, which denote the the proportion of a user's relevant recommendation items in the top-k list. For a target user u, precision $P_u(k)$ and recall $R_u(k)$ are defined as

$$P_u(k) = \frac{d_u(k)}{k} \tag{15}$$

$$R_u(k) = \frac{d_u(k)}{D_u} \tag{16}$$

where k is the length of recommendation list, $d_u(k)$ denotes the number of relevant recommendation items of user u (items rated by user u that are present in the probe set) in the top-k recommendation list and D_u is the total number of relevant items of user u. The larger the precision is, the more accurate the recommendation is.

Diversity is to show the degree of difference among recommendation items. There are two kinds of diversity metrics: the *Inter-user diversity* - to indicate the difference of items recommended to different users [41]. Another is called *Intra-user diversity*. In [14], an *Intra-user diversity* is introduced to show the ability of a recommendation technique to suggest different items to the target user. Motivated by the metric above, Zhou *et al* [42] propose another intra-user diversity. For the target user u, the *Intra-user diversity* is defined as

$$D_u(k) = \frac{1}{k(k-1)} \sum_{a \neq b} s(I_a, I_b) \tag{17}$$

where $s(I_a, I_b)$ denotes the similarity between items I_a and I_b. Big value of *Intra-user diversity* means less diverse recommendation suggested to the users.

Novelty is used to measure the ability of an algorithm to recommend novel items to the target user. In our algorithm, we test the novelty of the algorithm for a user, and then the popularity is defined as

$$N(k) = \frac{1}{k} \sum_{\alpha \in O_R^i} d_\alpha \tag{18}$$

where O_R^i is the top-k list of a user, and d_α is the degree of item α. Then the lower popularity of the algorithm is, the more novel the recommendation is.

In our experiments, equation (15) is used to test the accuracy of MOEA/D-RS, and equation (17) and (18) are employed to test the diversity and novelty of MOEA/D-RS, respectively.

C. Experimental results

The data-sets used in MOEA/D-RS are two benchmark data sets MovieLens [43] and Jester [44]. Both of them can be download form the website of Grouplens (http://www.grouplens.org/). The MovieLens 100k data consists of 100,000 ratings from 943 users on 1682 movies, and every rating is a integer ranging from 1 to 5. In our experiments, the Jester data used is the Jester3 data, containing ratings from 24938 users who have rated between 15 and 35 jokes on 100 jokes, in which the ratings are values ranging from -10.00 to 10.00. While in our experiments, to simplify the data set, we just keep the users who have rated more than 34 jokes and the jokes that have been rated over 20 times. Then the data set contains 32563 ratings from 931 users on 90 items, and we re-scope all the ratings in the range of 1-5.

To evaluate the performance of algorithms, the data is divided into two parts: the training set E^T and the probe set E^P. The training set is considered as the known ratings and no information from the probe set are allowed to be used to give prediction. In our experiments, the training set consists of 80% of the data and the probe set contains the remaining 20% of the data.

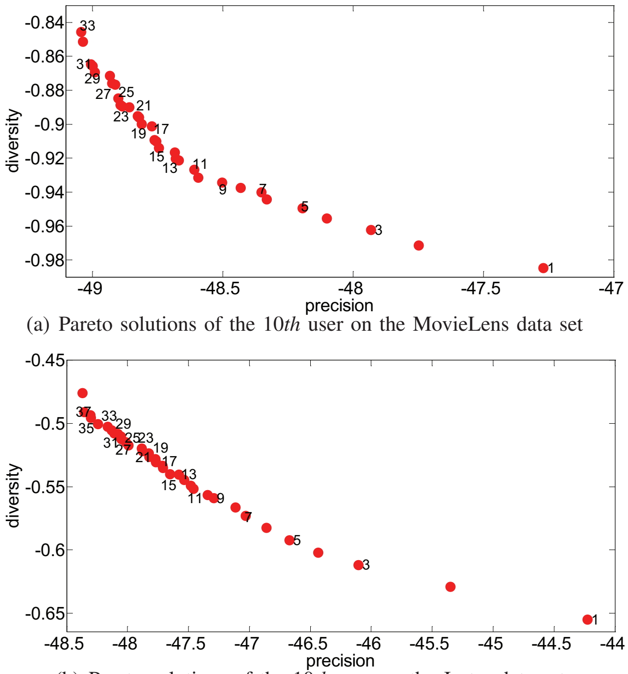
TABLE II

THE NUMBER OF PARETO SOLUTIONS IN MOEA/D-RS FOR FIRST TEN USERS ON TWO DATA-SETS.

ID	1	2	3	4	5	6	7	8	9	10
MovieLens	62	43	33	1	44	37	34	62	55	33
Jester	31	37	19	26	47	38	24	29	18	38

In our experiments, we just give the results on the first ten users. In TABLE II, we run MOEA/D-RS one time and the number of Pareto solutions i.e. different recommendation lists to every user are given. As shown in TABLE II, for the MovieLens data-set, except for the 4*th* user, MOEA/D-RS suggests many different recommendation lists to every user.

The 4th user is given just one recommendation list, because the top ranked items have similar scores rated by CF. And for the data-set Jester, MOEA/D-RS gives lots of different recommendation lists for each user.



(a) Pareto solutions of the 10th user on the MovieLens data set

(b) Pareto solutions of the 10th user on the Jester data set

Fig. 4. Pareto front of MOEA/D-RS on the 10th user of two test data-sets. The Pareto solution with small ID has low score but great diversity.

Fig. 3 shows the Pareto front of the 10th user on two data-sets in TABLE II. For the MovieLens data-set, our algorithm can give 33 different recommendation lists to this user, and for another data-set, 38 different recommendation results are generated. In Fig. 4, the Pareto solution with small ID has low summation of scores but high diversity value. The recommendation result with ID 1 has the highest precision but the lowest diversity, and the recommendations with ID 33 in (a) and 38 in (b) have the highest diversity but the lowest precision. The solutions become tight with the increase of the solution ID, so that our algorithm tends to find the lists with great accuracy but low diversity, because the data is sparse and items are given only a few ratings [27].

TABLE III reports the values of precision of our algorithm and the items-based CF. The 'mean' is the average precision of all the recommendation lists of MOEA/D-RS for a user. The 'max' and 'min' denotes the maximum and minimum of all the recommendation lists of MOEA/D-RS, respectively. Almost all the average values of our algorithm are lower than the values of CF. But we can see that MOEA/D-RS could give some recommendation lists that have better precision than CF, for for the 7th and 10th users, the maximum values of MOEA/D-RS are greater than that of CF. Besides, for other users our algorithm can generate recommendation lists with precision equal to values of CF, because the recommendation result of CF is just one of the Pareto solutions obtained by MOEA/D-RS.

TABLE IV gives the values of diversity of our algorithm and CF. All the labels in the table have the same meanings with TABLE III. For the data set of MovieLens, there are 6 users that their average values of recommendation results by MOEA/D-RS are less than the corresponding values by CF. And the first, fifth and tenth users can be suggest

TABLE III
THE VALUES OF PRECISION BY CF AND MOEA/D-RS ON TWO DATA-SETS.

	MovieLens				Jester			
ID	CF	MOEA/D-RS			CF	MOEA/D-RS		
		mean	max	min		mean	max	min
1	0.3	0.200	0.3	0.1	0.3	0.294	0.3	0.1
2	0.1	0.014	0.1	0	0.1	0.081	0.1	0
3	0.1	0.021	0.1	0	0.3	0.295	0.3	0.2
4	0	0	0	0	0.2	0.173	**0.3**	0
5	0	0	0	0	0.3	0.183	0.3	0
6	0.6	0.484	0.6	0.4	0.3	0.145	0.3	0
7	0.3	0.218	**0.4**	0.1	0.1	0.100	0.1	0.1
8	0.1	0.057	0.1	0	0.2	0.172	0.2	0
9	0	0	0	0	0.3	0.200	0.3	0
10	0.6	0.546	**0.7**	0.4	0.3	0.258	**0.4**	0.1

TABLE IV
THE VALUES OF DIVERSITY BY CF AND MOEA/D-RS ON TWO DATA-SETS.

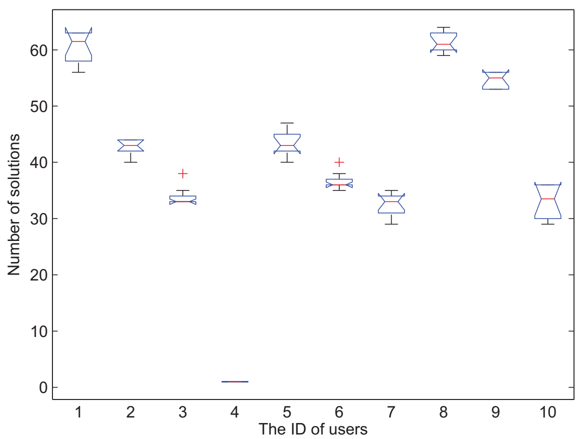
	MovieLens				Jester			
ID	CF	MOEA/D-RS			CF	MOEA/D-RS		
		mean	max	min		mean	max	min
1	0.046	0.060	0.095	**0.014**	0.184	**0.166**	0.195	0.143
2	0.122	**0.085**	0.139	0.038	0.184	**0.170**	0.204	0.122
3	0.024	**0.005**	0.024	0	0.159	0.159	0.181	**0.143**
4	0.085	0.112	0.112	0.112	0.145	0.149	0.197	**0.109**
5	0.023	0.023	0.062	**0.005**	0.185	0.185	0.235	**0.143**
6	0.059	**0.046**	0.084	0.024	0.153	**0.150**	0.211	0.118
7	0.070	**0.067**	0.089	0.046	0.148	0.158	0.206	**0.131**
8	0.081	**0.061**	0.086	0.033	0.152	0.167	0.220	**0.142**
9	0.118	**0.094**	0.135	0.072	0.160	0.161	0.203	**0.121**
10	0.076	0.089	0.121	**0.071**	0.164	**0.148**	0.179	0.125

TABLE V
THE VALUES OF NOVELTY BY CF AND MOEA/D-RS ON TWO DATA-SETS.

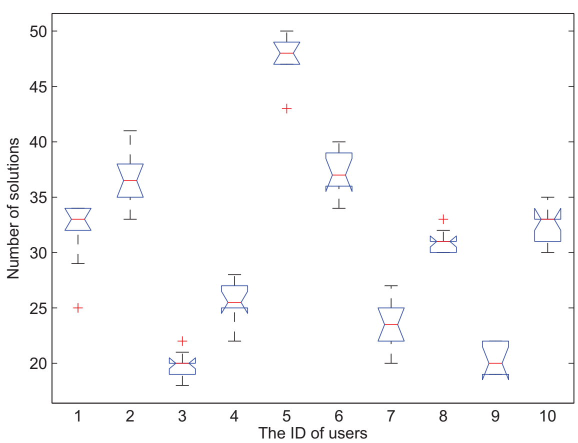
	MovieLens				Jester			
ID	CF	MOEA/D-RS			CF	MOEA/D-RS		
		mean	max	min		mean	max	min
1	97.6	**82.511**	110.3	19.4	306.5	**298.2**	320.2	139.6
2	83.9	93.863	113.3	**77.0**	202.2	**170.3**	207.0	55.2
3	27.5	**10.488**	28.4	3.7	315.5	**306.5**	336.8	196.4
4	181.7	**150.100**	150.1	150.1	203.3	**185.2**	268.7	67.0
5	61.2	**50.657**	65.5	33.1	270.9	**204.9**	289.6	59.0
6	119.6	**84.830**	124.0	53.7	294.5	**182.7**	296.2	54.1
7	86.1	**71.374**	120.0	51.0	328.8	**228.6**	330.3	126.0
8	139.0	**108.957**	177.5	64.3	213.7	**184.0**	215.4	59.2
9	75.6	**71.147**	104.0	46.8	297.6	**217.0**	297.6	56.0
10	198.0	**167.210**	202.7	124.7	294.4	**258.0**	354.0	138.1

some lists by MOEA/D-RS more diverse than the lists by CF. The fourth user is just given one Pareto solution, and this recommendation list is worse than the result of CF in diversity. For Jester, there are 4 users with average diversity of MOEA/D-RS less than the values of CF, and for the remaining six users, MOEA/D-RS can generate some lists more diverse than the lists by CF.

The novelty of the two algorithms on the two data-sets are shown in TABLE V. Equally, all the labels in this table have the same meanings with that in TABLE III. Except for the second user of MovieLens, the average novelty of our algorithm are lower than that of CF, so that our algorithm has a better performance to recommend unpopular items than CF.



(a) the results of the MovieLens data set



(b) the results of the Jester data set

Fig. 5. The boxplot of the number of solutions of MOEA/D-RS over 10 times on two data-sets.

We run MOEA/D-RS ten times, and record the number of recommendation lists to the first ten users every time. In Fig. 5, the stability of our algorithm is shown. For the 4*th* user in MovieLens, there is always only one recommendation list in every run. For other users, every time MOEA/D-RS gives different number of results, so our algorithm is unstable. The major contribution of MOEA/D-RS is to suggest many different recommendation results to a user. And in real recommendation, every time only one list is recommended to user, so we do not think instability of MOEA/D-RS is a fatal disadvantage.

V. Conclusion

In this paper, we propose a recommendation technique based on multiobjective evolutionary algorithm with decomposition, MOEA/D-RS. MOEA/D-RS does not only deal with the prediction score, but also maximize the diversity of the recommendation. And our algorithm can give a series of trade-off solutions to a user, and every Pareto solution in MOEA/D-RS is a recommendation list. Then MOEA/D-RS gives some alternative recommendation results for a decision maker to choose which will be recommended to users. The experiments show that our algorithm is effective to recommend diverse and unpopular items.

References

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

[2] G. Linden, B. Smith, and J. York, “Amazon. com recommendations: Item-to-item collaborative filtering,” *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[3] J. Bennett and S. Lanning, “The netflix prize,” in *Proceedings of KDD cup and workshop*, vol. 2007, p. 35, 2007.

[4] E. Brynjolfsson, Y. J. Hu, and M. D. Smith, “Consumer surplus in the digital economy: Estimating the value of increased product variety at online booksellers,” *Management Science*, vol. 49, no. 11, pp. 1580–1596, 2003.

[5] Z. Huang, H. Chen, and D. Zeng, “Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering,” *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 116–142, 2004.

[6] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen, “Collaborative filtering recommender systems,” in *The adaptive web*, pp. 291–324, Springer, 2007.

[7] A. M. Rashid, G. Karypis, and J. Riedl, “Learning preferences of new users in recommender systems: an information theoretic approach,” *ACM SIGKDD Explorations Newsletter*, vol. 10, no. 2, pp. 90–100, 2008.

[8] H.-N. Kim, A.-T. Ji, I. Ha, and G.-S. Jo, “Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation,” *Electronic Commerce Research and Applications*, vol. 9, no. 1, pp. 73–83, 2010.

[9] L. Martinez, L. G. Perez, and M. J. Barranco, “Incomplete preference relations to smooth out the cold-start in collaborative recommender systems,” in *Fuzzy Information Processing Society, 2009. NAFIPS 2009. Annual Meeting of the North American*, pp. 1–6, IEEE, 2009.

[10] A. Bellogin, P. Castells, and I. Cantador, “Precision-oriented evaluation of recommender systems: an algorithmic comparison,” in *Proceedings of the fifth ACM conference on Recommender systems*, pp. 333–336, ACM, 2011.

[11] L. Zhang, K. Zhang, and C. Li, “A topical pagerank based algorithm for recommender systems,” in *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 713–714, ACM, 2008.

[12] 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*, pp. 1097–1101, ACM, 2006.

[13] T. Jambor and J. Wang, “Optimizing multiple objectives in collaborative filtering,” in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 55–62, ACM, 2010.

[14] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, “Improving recommendation lists through topic diversification,” in *Proceedings*

of the 14th international conference on World Wide Web, pp. 22–32, ACM, 2005.
[15] N. Hurley and M. Zhang, "Novelty and diversity in top-n recommendation–analysis and evaluation," *ACM Transactions on Internet Technology (TOIT)*, vol. 10, no. 4, p. 14, 2011.
[16] P. Castells, J. Wang, R. Lara, and D. Zhang, "Workshop on novelty and diversity in recommender systems-divers 2011," in *Proceedings of the fifth ACM conference on Recommender systems*, pp. 393–394, ACM, 2011.
[17] T. Zhou, Z. Kuscsik, J.-G. Liu, M. Medo, J. R. Wakeling, and Y.-C. Zhang, "Solving the apparent diversity-accuracy dilemma of recommender systems," *Proceedings of the National Academy of Sciences*, vol. 107, no. 10, pp. 4511–4515, 2010.
[18] M. Rodriguez, C. Posse, and E. Zhang, "Multiple objective optimization in recommender systems," in *Proceedings of the sixth ACM conference on Recommender systems*, pp. 11–18, ACM, 2012.
[19] M. T. Ribeiro, A. Lacerda, E. S. de Moura, A. Veloso, and N. Ziviani, "Multi-objective pareto-efficient approaches for recommender systems," *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 1, pp. 1–20, 2013.
[20] M. T. Ribeiro, A. Lacerda, A. Veloso, and N. Ziviani, "Pareto-efficient hybridization for multi-objective recommender systems," in *Proceedings of the sixth ACM conference on Recommender systems*, pp. 19–26, ACM, 2012.
[21] Q. Zhang and H. Li, "Moea/d: A multiobjective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712–731, 2007.
[22] 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.
[23] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, and L. Thiele, "SPEA2: Improving the strength pareto evolutionary algorithm," 2001.
[24] J. D. Knowles and D. W. Corne, "Approximating the nondominated front using the pareto archived evolution strategy," *Evolutionary computation*, vol. 8, no. 2, pp. 149–172, 2000.
[25] M. Gong, L. Jiao, H. Du, and L. Bo, "Multiobjective immune algorithm with nondominated neighbor-based selection," *Evolutionary Computation*, vol. 16, no. 2, pp. 225–255, 2008.
[26] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
[27] L. Lü, M. Medo, C. H. Yeung, Y.-C. Zhang, Z.-K. Zhang, and T. Zhou, "Recommender systems," *Physics Reports*, vol. 519, no. 1, pp. 1–49, 2012.
[28] R. Burke, "Hybrid web recommender systems," in *The adaptive web*, pp. 377–408, Springer, 2007.
[29] B. Krulwich, "Lifestyle finder: Intelligent user profiling using large-scale demographic data," *AI magazine*, vol. 18, no. 2, p. 37, 1997.
[30] M. J. Pazzani, "A framework for collaborative, content-based and demographic filtering," *Artificial Intelligence Review*, vol. 13, no. 5-6, pp. 393–408, 1999.
[31] J. Salter and N. Antonopoulos, "Cinemascreen recommender agent: combining collaborative and content-based filtering," *IEEE Intelligent Systems*, vol. 21, no. 1, pp. 35–41, 2006.
[32] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 5–53, 2004.
[33] A. Felfernig, B. Gula, G. Leitner, M. Maier, R. Melcher, and E. Teppan, "Persuasion in knowledge-based recommendation," in *Persuasive Technology*, pp. 71–82, Springer, 2008.
[34] M. Mandl, A. Felfernig, E. Teppan, and M. Schubert, "Consumer decision making in knowledge-based recommendation," *Journal of Intelligent Information Systems*, vol. 37, no. 1, pp. 1–22, 2011.
[35] R. Burke, "Hybrid recommender systems: Survey and experiments," *User modeling and user-adapted interaction*, vol. 12, no. 4, pp. 331–370, 2002.
[36] A. Albadvi and M. Shahbazi, "A hybrid recommendation technique based on product category attributes," *Expert Systems with Applications*, vol. 36, no. 9, pp. 11480–11488, 2009.
[37] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-n recommendation tasks," in *Proceedings of the fourth ACM conference on Recommender systems*, pp. 39–46, ACM, 2010.
[38] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, p. 4, 2009.
[39] 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*, pp. 285–295, ACM, 2001.
[40] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[41] T. Zhou, L.-L. Jiang, R.-Q. Su, and Y.-C. Zhang, "Effect of initial configuration on network-based recommendation," *EPL (Europhysics Letters)*, vol. 81, no. 5, p. 58004, 2008.
[42] T. Zhou, R.-Q. Su, R.-R. Liu, L.-L. Jiang, B.-H. Wang, and Y.-C. Zhang, "Accurate and diverse recommendations via eliminating redundant correlations," *New Journal of Physics*, vol. 11, no. 12, p. 123008, 2009.
[43] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 230–237, ACM, 1999.
[44] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins, "Eigentaste: A constant time collaborative filtering algorithm," *Information Retrieval*, vol. 4, no. 2, pp. 133–151, 2001.