Multi-objective Personalized Recommendation Algorithm using Extreme Point Guided Evolutionary Computation

Qiuzhen Lin[1], Xiaozhou Wang[1], Bishan Hu[1], Lijia Ma[1]*, Fei Chen[1], Jianqiang Li[1], Carlos A. Coello Coello[2]

[1] *College of Computer Science and Software Engineering, Shenzhen University, Shenzhen, P.R. China*

[2] *CINVESTAV-IPN, Department of Computer Science, Mexico, D.F., 07360, Mexico*

Abstract:

Recommender systems recognize items to users based on their potential interests and they are important to alleviate the search and selection pressures induced by the increasing item information. Classical recommender systems mainly focus on the accuracy of recommendation. However, with the increase of the diversified demands of users, multiple metrics which may conflict with each other have to be considered in modern recommender systems, especially for the personalized recommender system. In this paper, we design a personalized recommendation system considering the three conflicting objectives, i.e., the accuracy, diversity and novelty. Then, to let the system provide more comprehensive recommended items, we present a multi-objective personalized recommendation algorithm using extreme point guided evolutionary computation (called MOEA-EPG). The proposed MOEA-EPG is guided by three extreme points and its crossover operator is designed for better satisfying the demands of users. The experimental results validate the effectiveness of MOEA-EPG when compared to some state-of-the-art recommendation algorithms in terms of the accuracy, diversity and novelty on recommendation.

Keywords*:* Evolutionary algorithm, personalized recommendation, multi-objective recommendation.

1. Introduction

With the rapid development of internet and Web2.0, consumers are facing a large amount of advertising and shopping information. To alleviate the search and selection burden for consumers, recommender systems (RSs) [1, 2] aim to suggest the suitable items by deducing the interesting information revealed from commodity. Nowadays, RSs have been successfully used in many applications, e.g., e-commerce websites recommend commodities [3, 4], social websites push some potential friends [5, 6], link inferences [7], and e-learning educational institution assists the students to choose courses and learning activities [8].

In recent years, many RSs have been proposed [9-12], including the content-based RSs, the collaborative filtering based RSs, the knowledge-based RSs and the hybrid RSs. The content-based RSs suggest the similar items by extracting the content's features from the profiles of items and users, the collaborative filtering based RSs exploit the community data (e.g., feedback rating, tags or clicks from other users) to make recommendation, while the knowledge-based RSs uses the knowledge bases and knowledge models (e.g., ontologies) for recommendation. The hybrid RSs are the combination of the above approaches. A comprehensive survey of the state-of-the-art RSs can be found in [13, 14].

* Corresponding author: Lijia Ma (omegamalj@gmail.com)

In general, most traditional RSs [15-19] mainly focus on the accuracy of recommendations. However, with the increase of diversified demands of users, multiple metrics are often considered in personalized RSs, such as diversity and novelty that may conflict with each other. As pointed out by the studies in [20, 21], it is easy to guarantee the accuracy for recommendation, but it is difficult to maintain the diversity and novelty for recommendation. Generally, the increase of diversity or novelty in RSs will decrease their accuracy. Therefore, personalized RSs are often required to consider the accuracy, diversity, and novelty simultaneously when making recommendation.

Some research studies have been proposed to balance the accuracy and diversity of recommendations [22-24]. In [22], the balancing problem between the accuracy and diversity was modeled as a quadratic programming problem using a control parameter, which determines the importance of diversity in the recommendation lists. Then, several optimization strategies were accordingly designed to optimize the target model [22]. Similarly, in [23], a hybrid recommendation algorithm was proposed by combining Heat-spreading (HeatS) algorithm and probabilistic spreading (ProbS) algorithm to improve the diversity and accuracy of recommendations. Moreover, in [24], some item ranking techniques were presented for giving diverse recommendations with comparable levels of accuracy. However, these algorithms [22-24] are limited to give recommendations with both high accuracy and good diversity, but ignore novelty.

Multi-objective optimization problems (MOPs) try to simultaneously optimize a set of conflicting objectives, which can be found in various research fields, such as engineering design [25], route planning [26], etc. When solving MOPs, a set of non-dominated solutions are returned, each of which is a tradeoff among different objectives [47]. Based on the studies in [48, 49], there is a trade-off among multiple conflicting objectives (i.e., accuracy, diversity, and novelty) when modeling the MOPs for RSs. Recently, multi-objective evolutionary algorithms (MOEAs) have been proposed to solve the modeled MOPs for RSs. For instance, in [27], the personalized recommendation was first modeled as an MOP with two objectives (accuracy and diversity), and then an MOEA with the ProbS technique was proposed. Following this work, two novel MOEAs [28, 29] were designed based on the framework of MOEA/D [29] and NSGA-II [44] for recommendations with long tail items. The works in [27-29] have showed their performances on balancing the accuracy and diversity of recommendations. However, these MOEAs are also limited to consider the two conflicting objectives (accuracy and diversity) and don’t consider the algorithm-specific knowledge in their designs.

In this paper, a multi-objective personalized recommendation algorithm with extreme point guided evolutionary computation is proposed, called MOEA-EPG. In MOEA-EPG, the accuracy, diversity and novelty of recommendations are regarded as the three conflicting objectives, and the aim of our algorithm is to optimize the modeled MOP for personalized recommendation. The prior knowledge in RSs is extracted from the MOP by finding the extreme value of each objective, which is then incorporated into our MOEA-EPG. As revealed by our experiments, this knowledge can effectively guide the population’s evolution, so as to improve the search efficiency. Moreover, the crossover operator is further designed for better satisfying the demands of users. To conclude, the main contributions of this paper are summarized as follows.

1) The recommendation problem is modeled as an MOP with three conflicting objectives: accuracy, diversity and novelty. The first objective measures the accuracy of recommendations determining whether the predicted ratings are similar to the true ones of users, the second one optimizes the diversity of recommendations by calculating the recommendation coverage, while the last one reflects the capability of RSs to recommend unknown items to users.
2) An extreme point guided method is proposed based on the prior knowledge of RSs, which is used to enhance the performance of personalized recommendation. For each objective, we choose the solution with the maximal value from the available items as the extreme point. Then, the chosen extreme points are embedded into the initialization of population to guide the evolutionary search toward the Pareto-optimal front.
3) At the crossover process, the similarity between users' interests is evaluated and then the users with a higher similarity will be selected for crossover with a larger probability, which helps to generate offspring solutions with good quality.
4) The experiments on two classical datasets (Movielens and Netflix) show the superiority of our MOEA-EPG over the current MOEAs [27, 30] in terms of accuracy, diversity and novelty.

The rest of this paper is organized as follows. Section 2 introduces the related background, such as the definition of RSs, MOPs, and classical recommendation algorithms. Section 3 presents the proposed MOEA-EPG algorithm, including the modeled MOP for RSs, individual representation, and the extreme point guided MOEA. Section 4 provides the experimental comparisons of MOEA-EPG with other competitive recommendation algorithms. At last, conclusions and future work are given in Section 5.

2. Related Background

2.1 Definition of RSs

RSs aim to predict potential likes and interests of users, according to their historical data about the evaluations of users to items. Items in RSs may represent any consumed contents, like books, movies, news, and music. Let U be a number of M users, and let I be a number of N items. Based on the historical data, a rating matrix R is generated to measure the evaluations of users to items [33]. The aim of RSs is to push lists of personalized recommended items to the users. To do recommendation, the ratings of all the items should be assigned for all users. However, in practical cases, some users may not give evaluations to all items, and these evaluations need to be predicted. Let $R^{M\times N}$ indicate the true ratings of M users to N items, and let $Pr^{M\times N}$ represent the predicted ratings. Then, a number of L items with the highest predicted ratings for a given user will be added into its recommendation list. Fig. 1 gives a simple RS with four users (u_1, u_2, u_3, u_4) and four items (i_1, i_2, i_3, i_4), which records the ratings of users to movies. Please note that each rating in this recommendation would be "like" and "dislike", while the solid and dashed lines indicate their true and predicted ratings, respectively.

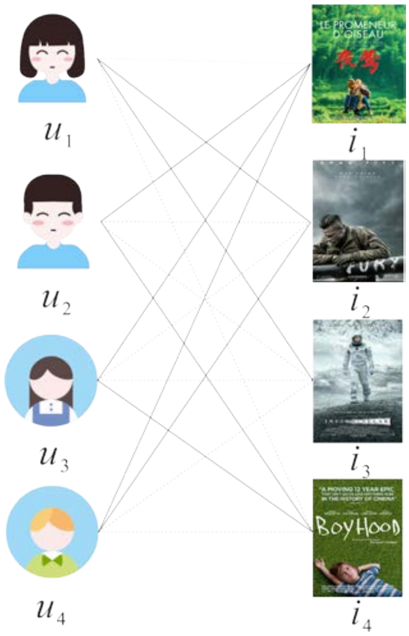


Fig. 1. Illustration of a simple RS.

2.2 Multi-objective Optimization Problems

Multi-objective optimization problems (MOPs) often consist of several conflicting objectives. In MOPs, the optimization of one objective would affect that of the others. Thus, the result of optimizing all the objectives is not a single optimal solution, but a set of solutions, each of which is the best trade-off among all the objectives [34-36]. Generally, an MOP can be modeled by

$$\text{Maximize} \quad F(x) = \{f_1(x), f_2(x), \ldots, f_m(x)\} \tag{1}$$

where $x = (x_1, x_2, \ldots, x_n) \in \Omega$ is a decision vector with n dimensions, Ω is the decision space, and m is the number of objectives. The MOP in Eq. (1) aims to maximize all the objectives simultaneously. In the following, some definitions about the Pareto optimum theory [37] are given.

Definition 1 (Pareto-dominance): A decision vector x dominates a decision vector y (noted as $x \succ$ y) if and only if

$$(\forall i \in \{1, 2, \ldots, m\}: f_i(x) \geq f_i(y)) \wedge (\exists j \in \{1, 2, \ldots, m\}: f_j(x) > f_j(y)). \tag{2}$$

Definition 2 (Pareto-optimal): A solution x is Pareto-optimal if and only if

$$\neg \exists y \in \Omega: \; y \succ x. \tag{3}$$

Definition 3 (Pareto-optimal set, PS): This set PS includes all the Pareto-optimal solutions.

$$PS = \{x \mid \neg \exists y \in \Omega: \; y \succ x\}. \tag{4}$$

Definition 4 (Pareto-optimal front, PF): This set PF contains all the values of the objective functions related to the Pareto-optimal solutions in PS.

$$PF = \{F(x) = (f_1(x), f_2(x), \ldots, f_m(x))^T \mid x \in PS\}. \tag{5}$$

2.3 Classical Recommendation Algorithms

User based Collaborative Filtering: It is one classical RS method [31, 38], which predicts the ratings of a user to items based on the preference of some users who show similar interests. For a given user u, its K most similar users are denoted by $S(u, K)$. Let I_u represent the items which are rated by the users in $S(u,$

K), but not by the user u. The predicted rating $Pr(u,i)$ of user u to an item i in I_u is computed by

$$Pr(u,i) = \frac{\sum_{v \in S(u,K)} s_{uv} \times r_{vi}}{\sum_{v \in S(u,K)} ||s_{uv}||}, \tag{6}$$

where s_{uv} is the similarity between users u and v, $||s_{uv}||$ is the absolute value of s_{uv}, and r_{vi} is the rating of user v to item i. Then, a number of L items in I_u with the largest Pr values are recommended to user u, where L is the length of recommendation list. There are many well-known methods for computing the users' similarity s_{uv}, such as Cosine Similarity (CS) [33], Pearson Correlation Similarity (PCS) [39], and Adjusted Cosine Similarity (ACS) [15]. In this paper, the CS method [33] is chosen due to its simplicity, which computes the similarity s_{uv} as follows:

$$s_{uv} = \frac{r_u \cdot r_v}{|r_u||r_v|}, \tag{7}$$

where r_u is the ratings of user u to all items, and $|r_u|$ is the length of r_u.

MOEA with ProbS (MOEA-ProbS) [27]: This algorithm proposed a multi-objective recommendation model to solve the conflicts between accuracy and diversity. In MOEA-Probs, the accuracy is evaluated by the ProbS method [40], while the diversity is measured by the coverage of the recommendations. MOEA-ProbS can provide multiple recommendations for users in an independent run, such that users can choose what they want from the multiple recommendations according to their preferences. However, the predicted ratings in [27] are not so accurate when compared to the traditional collaborative filtering (CF) algorithm, due to the lack of prior knowledge of RSs.

Probabilistic MOEA (PMOEA) [30]: This algorithm mainly introduced a new diversity indicator and a multi-parent probability crossover to have a better recommendation. PMOEA was validated to obtain a good balance between the two objectives (accuracy and diversity) in recommendations.

Following the MOEAs-based recommendation algorithms in [27, 30], we propose a multi-objective personalized recommendation algorithm using extreme point guided evolutionary computation to enhance the performance of RSs, which shows some advantages in terms of accuracy, diversity, and novelty.

3. Our algorithm: MOEA-EPG

In this section, we first model the MOP for RSs with three conflicting objectives: accuracy, diversity and novelty, and then present the MOEA-EPG algorithm to solve this problem.

3.1 Optimization Objectives

Three optimization objectives, i.e., accuracy, diversity, and novelty, are used as the objectives for personalized recommendations.

The accuracy of recommendations measures the similarity between the predicted ratings and the true user's ratings. In fact, it is impossible to compute the true user's ratings in the training stage. Therefore, the predicted ratings of items are used as the optimization objective. The predicted ratings of items for all the users in one cluster C are computed as follows:

$$Pr = \frac{\sum_{u \in C} \sum_{i \in S_u} pr_{ui}}{|C| \cdot L}, \tag{8}$$

where $|C|$ is the number of users in the cluster C, L is the length of the recommendation list, and S_u represents the list of items recommended to user u. In MOEA-ProbS [27], the predicted ratings pr_{ui} can be obtained by ProbS algorithm [40]. Here, following the work in [31], pr_{ui} is modified as follows:

$$pr_{ui} = \frac{\sum_{v \in S(u,K)} s_{uv} \cdot r_{vi}}{K}, \tag{9}$$

where $S(u,K)$ denotes the K most similar users to user u.

The diversity of recommendations reflects the difference between items in a recommendation list. Classical diversity metrics [43], such as inter-user diversity, intra-user diversity, and coverage, can be used in our algorithm. Here, the coverage is chosen to evaluate the diversity of recommendations, which is computed as follows:

$$cov = \frac{Nu}{N}, \tag{10}$$

where N is the total number of items, and Nu is the number of non-duplicated items in the recommendation lists for the users under the same cluster. Generally, RSs with a high coverage in Eq. (9) is valuable to users as a lot of choices can be provided for the decisions.

The novelty reflects the degree of recommendations, i.e., the number of unknown items that are recommended to users. Here, we chose the most common index, self-information [27, 30], to evaluate the novelty of recommended items. Let d_i represent the popularity degree of an item i. The self-information of item i is defined as follows:

$$N_i = \log_2 \frac{M}{d_i}, \tag{11}$$

where M is the total number of users. Generally, items with a low popularity degree are novel to users. For better recommendations, the average novelty $N(L)$ which represents the mean self-information of all the items in the recommendation list is used, which is computed as follows:

$$N(L) = \frac{1}{M} \sum_{u=1}^{M} \sum_{i \in S_u} \frac{N_i}{L}, \tag{12}$$

where S_u is the recommendation list of user u.

3.2 Our Method

3.2.1 Extreme Point Guided Method

To accelerate the convergence of recommendations, an extreme point guided method is used, which incorporates the prior knowledge of RSs. In this method, three extreme points, each of which has the maximal value for one objective, are used to extend the Pareto-optimal front.

The accuracy of recommendations in Eq. (8) will be affected by the coverage in Eq. (10) and the novelty in Eq. (12). In our approach, an optimal value for Eq. (8) is found at the initialization phase and it is treated as an extreme point to guide the evolutionary process, aiming to search the extreme region toward the accuracy. For each user u in one cluster C, let S_u be the set of available items for user u sorted by a descending order based on the predicted ratings. Then, the first L items in S_u are assigned to the recommendation list $s_u^{max\_accuracy}$ and then recommended to user u. This extreme point $e_{accuracy}$ will have the maximal accuracy, as follows:

$$PR(e_{accuracy}) = \frac{\sum_{u \in C} \sum_{i \in s_u^{max\_accuracy}} pr_{ui}}{|C| \cdot L}. \tag{13}$$

Similarly, the second extreme point is found by only optimizing the coverage. The optimal value for coverage can include diverse categories of items. Therefore, assume that the users in a cluster C are represented by $U = \{u_1, u_2, ..., u_M\}$, where M is the number of users in the cluster C. To enhance the coverage, the items that have been recommended to previous users are removed and no longer pushed to the other users. This method assures the diversity of items and reduces duplicates in the cluster C. Thus, this extreme point $e_{coverage}$ shows the maximal coverage, as follows:

$$cov(e_{coverage}) = \frac{Nu_{\max}}{N}, \tag{14}$$

where $Nu_{\max}$ is the number of distinct items for all the users under the same cluster.

The third extreme point is obtained by only optimizing the novelty of recommendations. Similarly, for each user u in a cluster C, the L items in S_u with the largest self-information in Eq. (12) are first found, and then they are assigned to the recommendation list of user u. Based on Eq. (11), the items with low degree will have high self-information. Thus, the L items with the smallest degrees for each user u are actually recommended, and they are recorded by $S_u^{max\_novelity}$. This extreme point $e_{novelity}$ will have the maximal novelty, as follows:

$$N_L(e_{novelity}) = \frac{1}{M} \sum_{u=1}^{M} \sum_{i \in S_u^{max\_novelity}} \frac{N_i}{L}. \tag{15}$$

where N_i is the self-information of item i as defined in Eq. (11).

These three extreme points can be used to accelerate the convergence of our algorithm toward the entire Pareto-optimal front, and also to enhance the population's diversity.

3.2.2 Solution Encoding

The encoding formula in [27] is used to convert the recommendation list as the solutions for MOPs. In RSs, recommendation results should be provided to all the users in the same cluster. Therefore, a solution represents recommendation lists of all users in the same cluster. Given the number of users M in a cluster and the length of recommendation list L, the solution is encoded by a matrix with size $M \times L$. Therefore, the matrix consists of all items recommended to users in a cluster. The solution encoding is shown in Fig. 2, in which each row represents the L items recommended for each user. In general, RSs are not allowed to recommend the same items to a user twice.

3.2.3 Evolutionary Operators

The uniform crossover [27] is used in our approach due to its simplicity. As shown in Fig. 3, in each crossover process, two similar users are chosen to execute the crossover. To avoid generating invalid solutions (i.e., one item is recommended to a user twice or more), an additional operation which propagates the same elements in the two parents to the offspring is executed. In the remaining positions, the offspring will inherit from the first parent if a randomly generated value in [0, 1] is larger than 0.5; otherwise it will inherit from the second parent [27]. The above crossover is executed on all pairs of parent solutions.

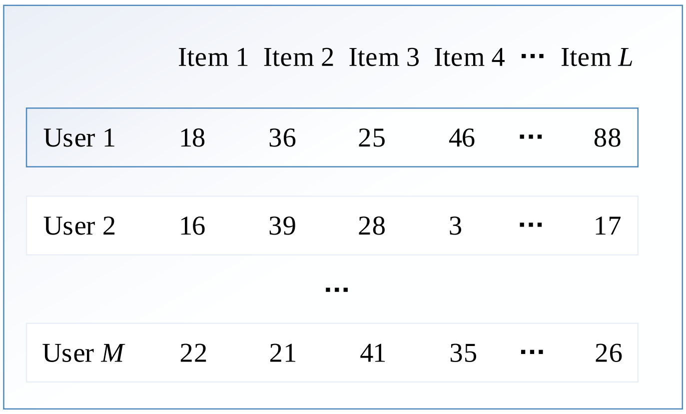


Fig. 2. Illustration of chromosome encoding.

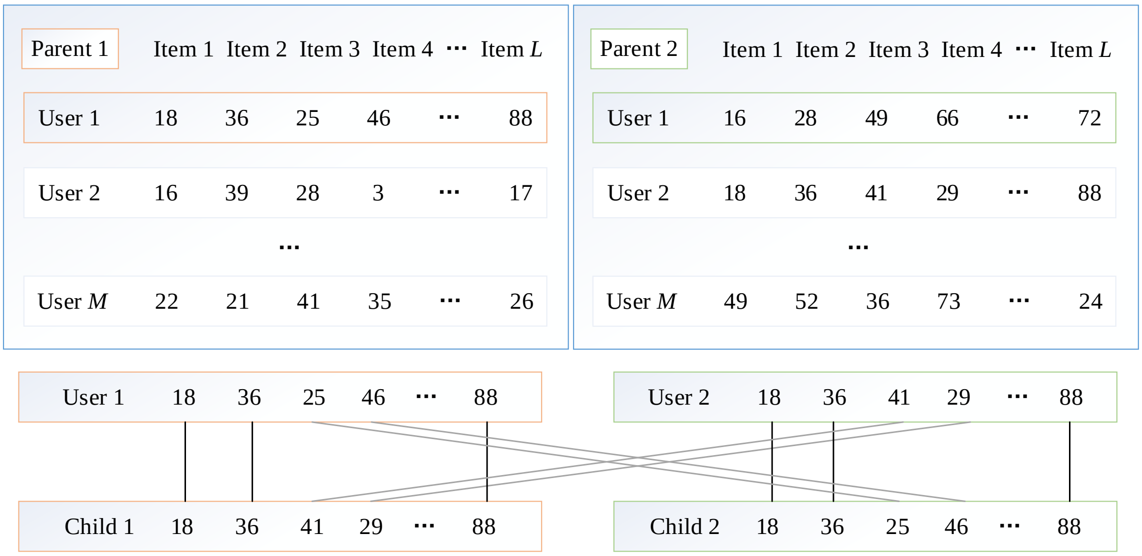


Fig. 3. Illustration of crossover operator for two similar users on two parent solutions (Parent 1 and Parent 2).

A mutation operator is performed on each individual generated by the crossover. Its steps are detailed as follows: if a gene from the parent matrix is selected for mutation, an available item is randomly selected from the items. To avoid useless solutions, we choose one of the remaining items that does not exist in the parent.

3.2.4 Complete Algorithm of MOEA-EPG

The flowchart of MOEA-EPG is shown in Fig. 4. MOEA-EPG includes two main parts: data processing and evolutionary algorithm. Moreover, the framework of MOEA-EPG is also provided in Algorithm 1, where N is population size, T means the maximum number of generations, K shows the ratio of similar users, P records the recommendation lists for all users, and L is the length of recommendation list.

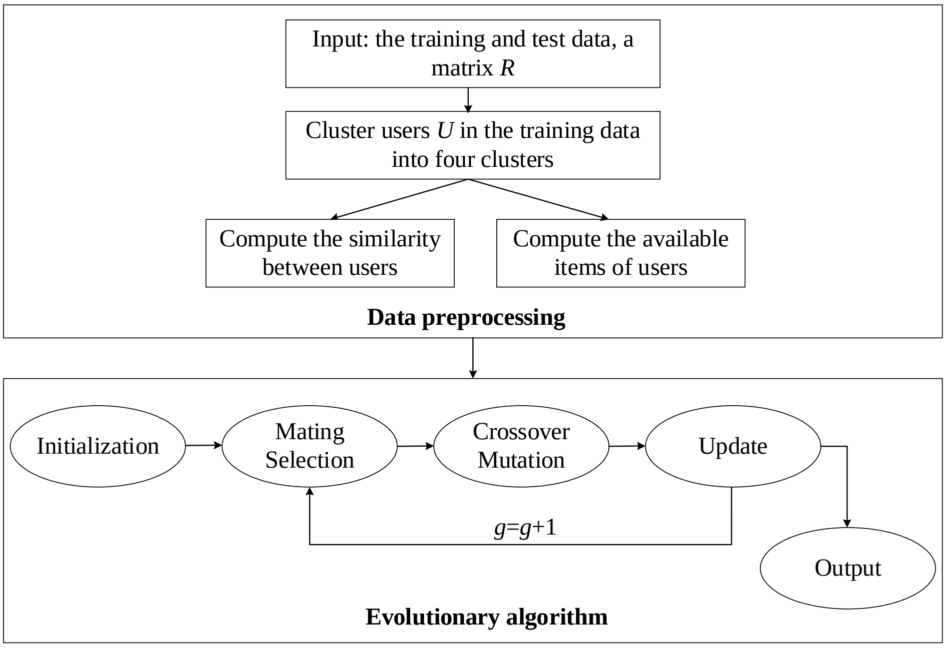


Fig. 4. Flowchart for MOEA-EPG.

Algorithm 1: Framework of MOEA-EPG

Input the training data, the test data, and a binary matrix R
Set g=0, L=10, N=100, T=3000, K=0.5
Clustering all the users U in the training data into four clusters with size M_j, $j=\{1,2,3,4\}$
$U=\{U_1,U_2,U_3,U_4\}$ and $U_j=\{u_1,u_2,...,u_{Mj}\}$
Set $P=\{p_1,\ p_2,\ p_3,\ p_4\}$ with $p_j=[\ ],\ j=1,2,3,4$
for j=1 to 4
    Compute the similarities of different users in U_j based on Eq. (7)
    Set $pr_j=\{0\}_{Mj\times Mj}$
    for k=1 to M_j
        Compute the available items of user u_k, and set it to be the set $I_{u_k}=\{i_1,i_2,...,i_{Ns}\}$
        for n=1 to Ns
            Compute the predict rating pr_{ui} for user u_k to item i_n by Eq. (8)
    $EPOP$ = Initialization(L, N, R, pr)
    while $g<T$
        Epa = Objective_Func($EPOP$)
        Epa = Fast_Non_Dominated_Sort(EPa)
        $EPOP$ = Selection($EPOP$, Epa)
        $EPOP$ = Uniform_Crossover($EPOP$)
        $NPOP$ = Mutation($EPOP$)
        POP = [$NPOP$; $EPOP$]
        $EPOP$ = Update(POP, Epa, N)
        $g=g+1$
    $p_i=EPOP$
Return $\boldsymbol{P}$

In the phase of data processing, the users U are classified into several clusters based on the K-means clustering [41] in the training data. In this paper, we divide users into four clusters, i.e., U_1, U_2, U_3, U_4, and four recommendation lists p_1, p_2, p_3, p_4 are initialized as four empty sets, as shown in lines 3-5. After that, the available items I_u of each user and the predicted ratings pr_{ui} of these items are calculated according to the similarities of different users in lines 6-12. The framework of MOEA-EPG is given as follows.

Algorithm 2：Fast_Non_Dominated_Sort(*Epa*)

1 **Input** the population Epa
2 **for each** $p \in Epa$
3 $\quad S_p = \varnothing$, $M_P = 0$
4 $\quad$ **for each** $q \in Epa$
5 $\quad\quad$ **if** $(p \prec q)$ then // If p dominates q
6 $\quad\quad\quad S_p = S_p \cup \{q\}$ // Add q to the set of solutions dominated by p
7 $\quad\quad$ **else if** $(q \prec p)$ then
8 $\quad\quad\quad M_P = M_P + 1$ // Increment the domination counter of p
9 $\quad$ **if** $M_P = 0$ **then** // p belongs to the first front
10 $\quad\quad EPa_{\text{rank}} = 1$
11 $\quad\quad F_1 = F_1 \cup \{p\}$
12 i=1 // Initialize the front counter
13 **while** $F_i \neq \varnothing$
14 $\quad S = \varnothing$ // Used to store the members of the next front
15 $\quad$ **for each** $p \in F_i$
16 $\quad\quad$ **for each** $q \in S_p$
17 $\quad\quad\quad M_q = M_q - 1$
18 $\quad\quad\quad$ **if** $M_q = 0$ **then** // q belongs to the next front
19 $\quad\quad\quad\quad q_{\text{rank}} = \text{i}+1$
20 $\quad\quad\quad\quad H = H \cup \{q\}$
21 $i = i+1$
22 $F_i = H$
23 **Return** $\boldsymbol{F_i}$

In line 13, the three extreme points ($e_{accuracy}$, $e_{coverage}$, and $e_{novelity}$) are calculated using Eqs. (13-15) and then the algorithm executes initialization. In line 15, the function Objective_Func(.) calculates the objective values of individuals in the population *EPOP* based on Eqs. (8), (10), (12) and records the final results at *Epa*. After that, the function Fast_Non_Dominated_Sort(.) in Algorithm 2 is executed on the evolutionary population to generate the non-dominated fronts based on *Epa* which are the solutions on the first level in line 16. Then, the function Selection(.) in line 17 selects parent solutions from the population *EPOP* based on the non-dominated sorting results by the binary tournament [45]. The functions Uniform_Crossover(.) and Mutation(.) in lines 18-19, execute the crossover and mutation operations in Section 3.3 to parent population *EPOP*, respectively, which produce offspring population *NPOP*. Finally, the function Update(.) chooses N individuals as *POP* from *EPOP* and *NPOP* for the next evolution based on the elitist selection strategy [44] (lines 20-21). This strategy is composed of a fast non-dominated sorting and a crowding distance comparison. Then, the loop of evolutionary algorithm is repeated until the current generation number g reaches the predefined maximal generation number T. If this termination condition is satisfied, our algorithm will return the recommendation results *P* and all users are recommended (line 24).

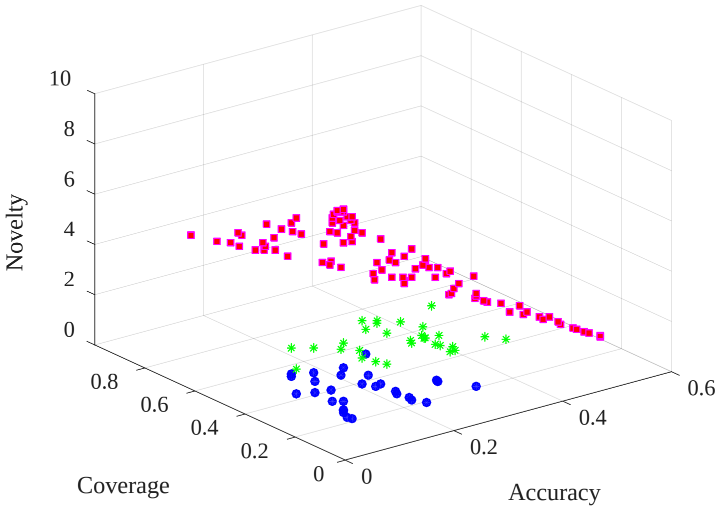


Fig. 5. A fast non-dominated sorting procedure in NSGA-II in the three objective optimizations [44].

Table 1
Properties of the Movielens data sets.

Data sets	Users	Items	Sparsity
Movieslens 1	200	1682	1.39×10^{-2}
Movieslens 2	258	1682	5.17×10^{-2}
Movieslens 3	227	1682	2.38×10^{-2}
Movieslens 4	258	1682	6.89×10^{-2}

Table 2
Properties of the Netflix data sets.

Data sets	Users	Items	Sparsity
Netflix 1	191	3952	2.11×10^{-2}
Netflix 2	250	3952	1.92×10^{-2}
Netflix 3	375	3952	2.73×10^{-2}
Netflix 4	184	3952	2.21×10^{-2}

The function Fast_Non_Dominated_Sort(.) in Algorithm 2 works as follows. Firstly, the solutions on the first level are found based on the non-dominated sort strategy (lines 4-11 in Algorithm 2). Secondly, the solutions on the first level are deleted, and the non-dominated sort strategy works on the rest of solutions to find the solutions on the top level (lines 13-20 in Algorithm 2). The steps above are executed until all solutions are sorted in a ranking level (lines 21-23 in Algorithm 2). In this case, all individuals are divided into multiple levels. The detailed descriptions to the Fast_Non_Dominated_Sort(.) in Algorithm 2 can be found in [44]. Here, a simple example is given in Fig. 5 when optimizing the three conflicting objectives (accuracy, coverage, and novelty). The tradeoff on these three objectives can be well observed and the solutions plotted by the red circles on the first level are preferred, as they are more effective and efficient to approach the PF.

4. Experimental Results

In this section, experiments on two datasets are executed to examine the performance of MOEA-EPG. First, some settings about the experiments are introduced, including the experimental data, the performance metrics and the parameters settings. Second, comparisons of MOEA-EPG with two state-of-the-art recommendation algorithms (i.e., User-based-CF [31] and MF [32]) and two competitive MOEAs for RSs (i.e., MOEA-ProbS [27] and PMOEA [30]) are made to demonstrate the effectiveness of MOEA-EPG in the accuracy, diversity and novelty of the recommendations. Finally, the effectiveness of the extreme point guided method and the impacts of some parameters are analyzed.

4.1 Experimental Data

In our experiments, two classical benchmark data sets, i.e., Movielens and Netflix, are used. The Movielens data set includes 943 users and 1682 movies, which can be downloaded from the website of GroupLens Research (http://www.grouplens.org/). The original ratings in this data are assigned from 1 to 5. Here, a binary rating system is used by using "like" or "dislike". An item is assigned to the binary rating "like" by a user when his original rating is larger than 3 [27]. Then, 80% of the data set is randomly selected as the training data set with known information for recommendations, while the remaining data are used as the test data set. To make a fair comparison, similar to [27], the users in this data are divided into four clusters by using *K*-means algorithm [46], each of which is used as the test data. The properties of these data sets are presented in Table 1 [23].

The Netflix data set was used for the Netflix prize, which can be downloaded from the contest website (http://www.netflixprize.com/). Here, a random subset of Netflix set is used, containing 155177 ratings of 1000 users on 3952 movies. The properties of this data set are shown in Table 2.

4.2 Performance Metrics

Here, the hypervolume (*HV*) metric [47] is adopted to evaluate the convergence and diversity of solutions, which is widely used for the evaluations of MOEAs [35, 36]. Let $z^r = (z_1^r, \cdots, z_m^r)^T$ be a reference point in the objective space that is dominated by all vectors in true PF. The *HV* metric calculates the size of the objective space dominated by the solutions in *S* and bounded by z^r, which is computed as follows:

$$HV(S) = VOL(\bigcup_{x \in S}[f_1(x), z_1^r] \times \cdots \times [f_m(x), z_m^r]), \tag{16}$$

where $VOL(\cdot)$ denotes the Lebesgue measure. In this paper, the reference point z^r is set to (0, 0) and (0, 0, 0) for bi-objective and three-objective problems, respectively. A large *HV* indicates the solution set has a good quality. In this experiment, all the algorithms with the highest *HV* from 30 runs are selected for comparisons.

Moreover, three performance metrics are used, i.e., precision, coverage, and novelty. Precision is the most important metric for evaluating RSs. For a given user *u*, its precision is defined as the ratio of relevant items, which are correctly recommended, as defined by

$$P_u = \frac{NR_u}{L}, \tag{17}$$

where NR_u is the number of common items in the recommendation list of user u and the test data set preferred by user u, and L is the length of recommendation list. The recommendation accuracy in this paper is measured by the mean precision of all the users.

The coverage metric of RSs illustrates the ratio of distinct items in the recommendation list over all the available items. Generally, RSs with a low coverage may be ineffective to users, as they are not enough to make decisions. In this paper, the objective in Eq. (10) is used as a performance metric to measure the diversity of RSs.

The novelty of RSs reflects how well RSs can recommend unknown items to users. Generally, RSs with a large novelty will extend user's interests. In this paper, the objective in Eq. (12) is used as a performance metric for this indicator.

4.3 Experimental Settings of Comparison Algorithms

The related parameters in comparison algorithms are listed in Table 3, and the length of recommendation list L is set to 10. To allow a fair comparison, the related parameters for each algorithm were set as suggested in the corresponding references [27, 30, 31, 32].

In Table 3, N is the population size, p_c and p_m are respectively the crossover probability and the mutation probability, p_n is the number of parents used for crossover in PMOEA (PMOEA-ProbS and PMOEA-CF), n is the number of neighbors, and Tr is the number of independent trials. Moreover, in MOEA-EPG, K means the ratio of the most similar users to the target users within a cluster. The comparisons of MOEA-EPG with these algorithms are comprehensive and effective to validate its performance.

Table 3: The related parameters in comparison algorithms.

Algorithms	Parameters Settings
User-based-CF	L=10
MF	L=10
MOEA-ProbS	p_c =0.8, p_m =1/L, L=10, N=100, T=3000, Tr=30
PMOEA	p_n =5, p_m =1/L, L=10, N=100, T =3000, Tr=30, n=20
MOEA-EPG	p_c =0.8, p_m =1/L, L=10, N=100, T=3000, Tr=30, $K = 0.5$

4.4 Effectiveness of MOEA-EPG

To study the effectiveness of our MOEA-EPG, a three-objective model is used to show the experimental results of MOEA-EPG on all the eight data sets. The non-dominated solutions with the highest HV value [47] are plotted as the final results for each data set. In Fig. 6, the final solutions of MOEA-EPG are plotted for all the eight data sets on the objective space. As observed from Fig. 6, there exists a tradeoff among the used three objectives, which forms a non-dominated front. Each solution in the non-dominated front is marked by the blue square. From these plots, we can observe that the quality and distribution of solutions are quite good.

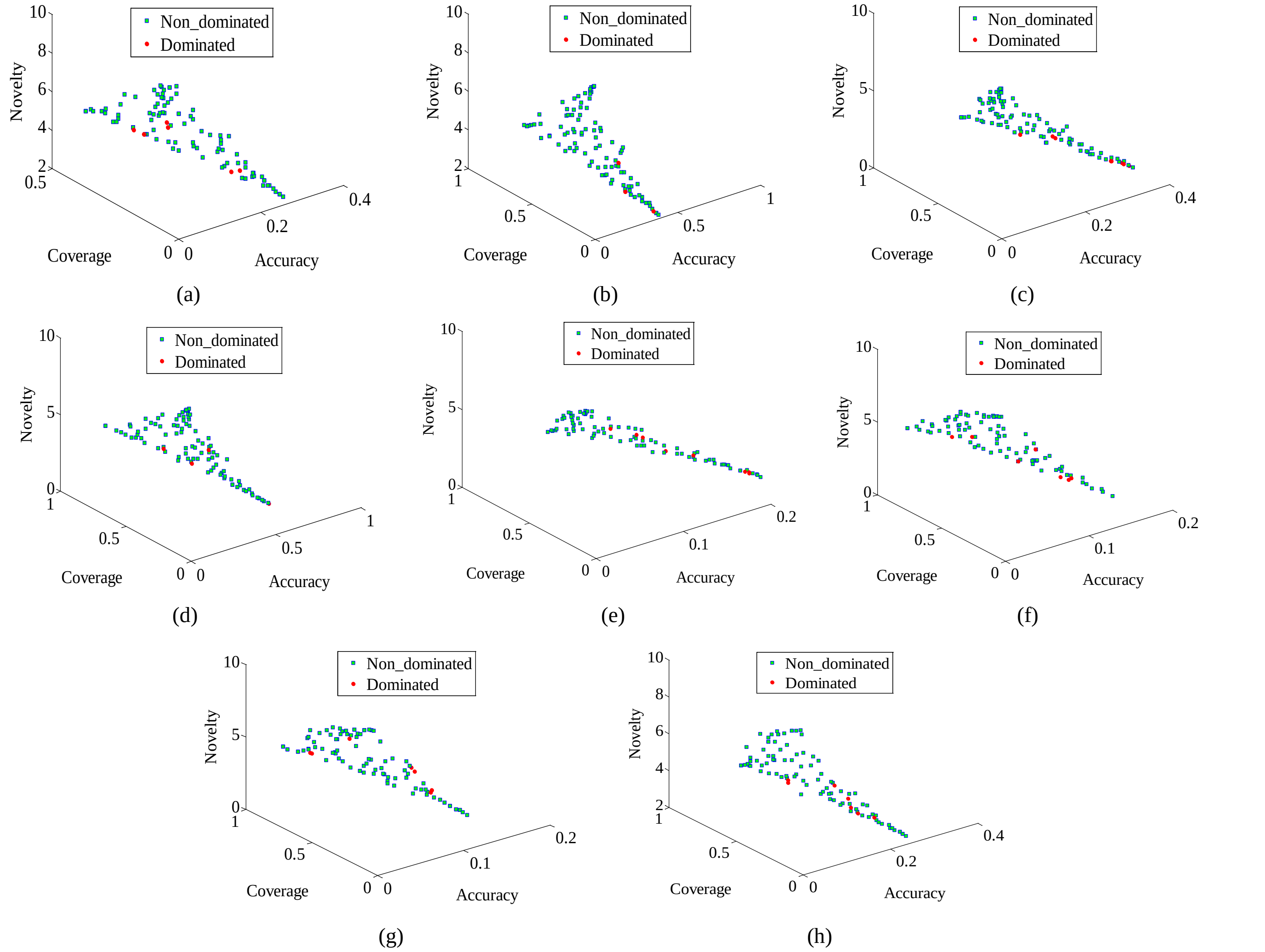


Fig. 6. Final solutions of MOEA-EPG under the accuracy-coverage-novelty space for (a) Movielens 1, (b) Movielens 2, (c) Movielens 3, (d) Movielens 4, (e) Netflix 1, (f) Netflix 2, (g) Netflix 3 and (h) Netflix 4.

To further show the distributions of *HV* results, Fig. 7 gives the boxplots of different *HV* values for eight data sets obtained by MOEA-EPG in the accuracy-coverage space under 30 runs. On each box diagram, the central mark is the median, while the upper and lower edges of the box mean the 25th and 75th percentiles, respectively. Obviously, the distribution of *HV* values of MOEA-EPG is robust, which further confirms the effectiveness of our algorithm.

4.5 Comparisons with Other Competitors

In this subsection, our algorithm is compared with five competitors on the classical Movielens and Netflix data sets. As User-based-CF and MF only obtain one optimal value in one run, they cannot be used as comparison algorithms using *HV*. In our experiments, the performance comparisons are conducted under different objectives, i.e., the accuracy-coverage, the accuracy-novelty, and the accuracy-coverage-novelty spaces.

Since the computation difficulties under different objectives are varied, the parameter T is set to 3000 for the accuracy-coverage and accuracy-novelty spaces, and set to 6000 for the accuracy-coverage-novelty space, while the remaining parameters are set the same as listed in Table 3. All the mean *HV* values and the standard deviations are collected for performance comparisons in Table 4, where the best result is highlighted in boldface. From Table 4, it can be observed that MOEA-EPG shows the best performance among all the MOEAs based recommendation algorithms, especially in the accuracy-novelty space and accuracy-coverage-novelty spaces.

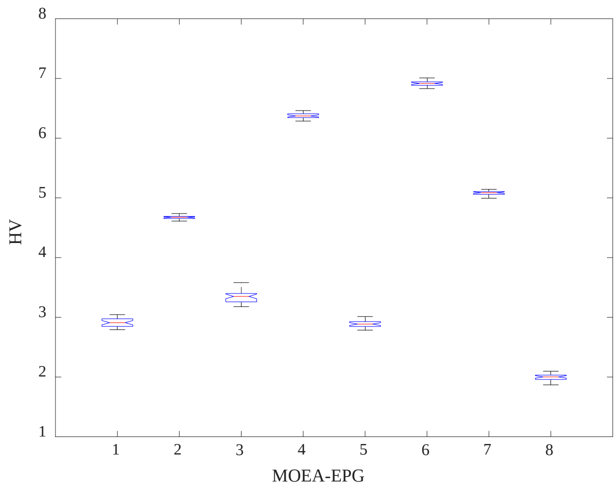


Fig. 7. The boxplots of *HV* values for MOEA-EPG under the accuracy-coverage space for different datasets, where 1: Movielens 1, 2: Movielens 2, 3: Movielens 3, 4: Movielens 4, 5: Netflix 1, 6: Netflix 2, 7: Netflix 3 and 8: Netflix 4.

Table 4

HV comparison results of MOEA-EPG and other algorithms on all the objectives for eight data sets.

Data sets	Test Instance	MOEA-ProbS	PMOEA-ProbS	PMOEA-CF	MOEA-EPG
Movielens 1	Accuracy-coverage	1.443 1.38e-02	1.533 1.13e-02	1.368 1.36e-02	**3.202 7.74e-02**
	Accuracy-novelty	11.40 1.45e-01	9.667 1.16e-01	9.288 1.07e-01	**36.89 6.05e-02**
	Accuracy-coverage-novelty	5.504 1.06e-01	5.319 1.13e-01	4.818 9.50e-02	**12.23 5.34e-01**
Movielens 2	Accuracy-coverage	1.531 1.03e-02	1.192 1.83e-02	1.085 1.68e-02	**4.677 3.46e-02**
	Accuracy-novelty	10.96 1.12e-01	9.288 1.23e-01	8.819 1.34e-01	**35.43 7.83e-02**
	Accuracy-coverage-novelty	6.027 1.23e-01	5.494 1.43e-01	4.494 5.33e-02	**21.28 2.74e-01**
Movielens 3	Accuracy-coverage	1.543 1.36e-02	1.290 1.49e-02	1.030 1.34e-02	**3.343 1.07e-02**
	Accuracy-novelty	11.21 1.07e-01	9.218 2.15e-01	9.143 2.57e-01	**36.10 7.62e-02**
	Accuracy-coverage-novelty	5.770 9.50e-02	5.086 3.23e-01	4.833 5.46e-02	**14.05 3.52e-01**
Movielens 4	Accuracy-coverage	1.763 1.15e-02	1.180 1.13e-02	1.123 3.87e-02	**6.385 5.62e-02**
	Accuracy-novelty	13.90 1.66e-01	9.271 1.11e-01	9.332 1.76e-01	**38.24 1.22e-02**
	Accuracy-coverage-novelty	6.008 1.02e-01	5.332 1.13e-01	4.210 6.27e-02	**18.68 3.25e-01**
Netflix 1	Accuracy-coverage	1.848 6.54e-02	1.548 4.91e-02	1.174 5.34e-02	**2.877 7.49e-02**
	Accuracy-novelty	12.69 5.56e-03	10.34 4.72e-02	9.375 1.99e-02	**30.44 2.40e-02**
	Accuracy-coverage-novelty	6.972 3.08e-02	5.554 1.15e-02	4.307 4.22e-02	**10.41 2.97e-02**
Netflix 2	Accuracy-coverage	1.652 9.08e-02	1.359 1.02e-02	1.183 2.20e-02	**6.914 5.29e-02**
	Accuracy-novelty	12.82 6.54e-02	10.48 1.49e-02	8.576 6.76e-03	**31.96 1.22e-02**
	Accuracy-coverage-novelty	7.917 4.27e-02	6.547 5.40e-02	4.251 6.27e-02	**12.10 6.57e-03**
Netflix 3	Accuracy-coverage	1.763 3.27e-02	1.475 4.99e-02	1.146 4.48e-02	**5.082 3.62e-02**
	Accuracy-novelty	13.71 3.94e-03	9.692 2.13e-02	8.446 2.48e-02	**29.65 8.97e-03**
	Accuracy-coverage-novelty	7.330 4.14e-02	6.368 2.16e-03	4.302 2.49e-02	**13.27 3.25e-01**
Netflix 4	Accuracy-coverage	1.440 4.36e-02	1.252 1.13e-02	1.018 3.87e-02	**1.975 1.05e-02**
	Accuracy-novelty	12.63 3.34e-04	9.271 1.87e-02	7.659 1.92e-02	**28.25 6.08e-02**
	Accuracy-coverage-novelty	6.358 5.99e-02	5.630 4.97e-02	4.423 6.92e-02	**11.92 3.25e-02**

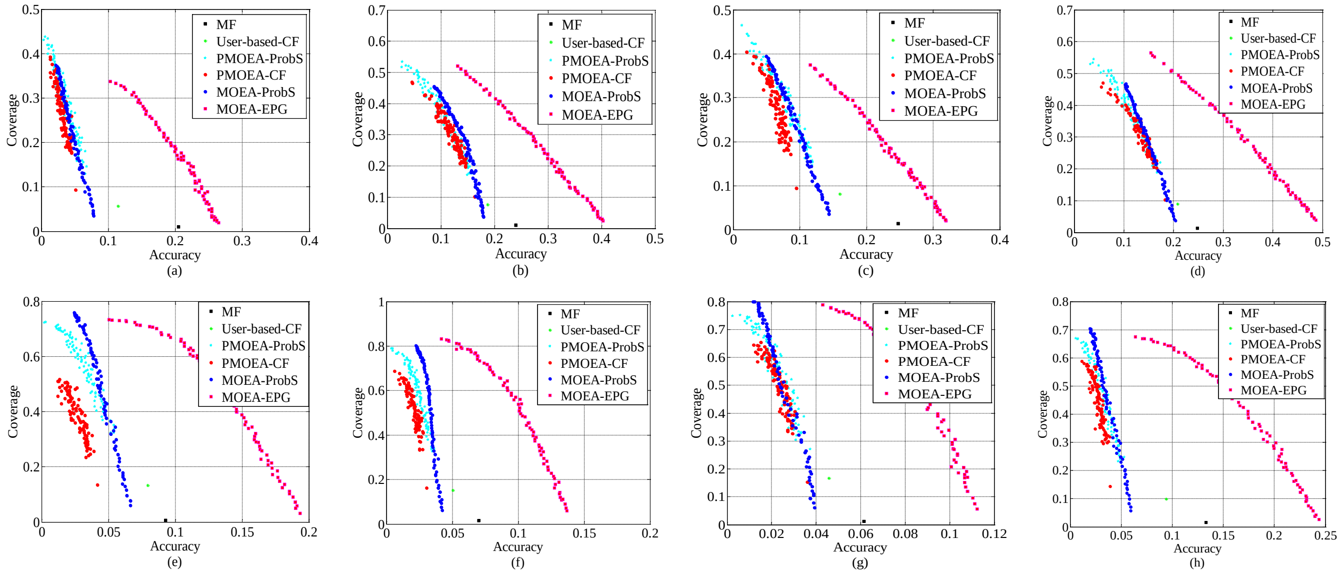


Fig. 8. Final recommendation results of comparison algorithms under the accuracy-coverage space for (a) Movielens 1, (b) Movielens 2, (c) Movielens 3, (d) Movielens 4, (e) Netflix 1, (f) Netflix 2, (g) Netflix 3 and (h) Netflix 4.

To visually show the advantages of MOEA-EPG, Figs. 8-10 give the final recommendation results with the best *HV* values regarding the accuracy-coverage, accuracy-novelty, and accuracy-coverage-novelty spaces, respectively. In Fig. 8, it is observed that, when compared to User-based-CF and MF, MOEA-EPG is advantageous, as it can find better results in the accuracy and coverage. Moreover, for the data sets of Movielens 2, Movielens 4 and Netflix 2, MOEA-EPG has the advantages over MOEA-ProbS, PMOEA-ProbS and PMOEA-CF. For the rest of data sets, MOEA-EPG performs competitively, as it can produce the best results among all the competitors. Only a few results of MOEA-ProbS, PMOEA-ProbS and PMOEA-CF are better than those of MOEA-EPG in the coverage. However, this performance enhancement for coverage decreases a large proportion of accuracy. For Movielens 1, Movielens 3 and Netflix1, the loss on their accuracy is significant. In Fig. 9, it is also confirmed that the advantages of MOEA-EPG are particularly pronounced, as MOEA-EPG can always outperform the other algorithms to find good recommendation solutions for all the data sets, regarding of both accuracy and novelty. In Fig.10, MOEA-EPG also performs much better than MOEA-ProbS, PMOEA-ProbS and PMOEA-CF. This indicates that MOEA-EPG is able to produce superior solutions in terms of the accuracy, coverage and novelty.

4.6 Effectiveness of the extreme point guided method

In this section, the effectiveness of the extreme point guided method is studied. As introduced in Section 3.2.1, three extreme points with respect to accuracy, coverage, and novelty, will be used in the evolutionary process to guide the search processes. To further study the underlying rationality of our proposed method, MOEA-EPG is compared to its three variants, i.e., MOEA-EPG-I, MOEA-EPG-II and MOEA-EPG-III. MOEA-EPG-I, MOEA-EPG-II, and MOEA-EPG-III are implemented, to remove the accuracy extreme point, the coverage extreme point, and the novelty extreme point from the original MOEA-EPG, respectively. In this experiment, we use Movielens data sets due to the limitation of paper.

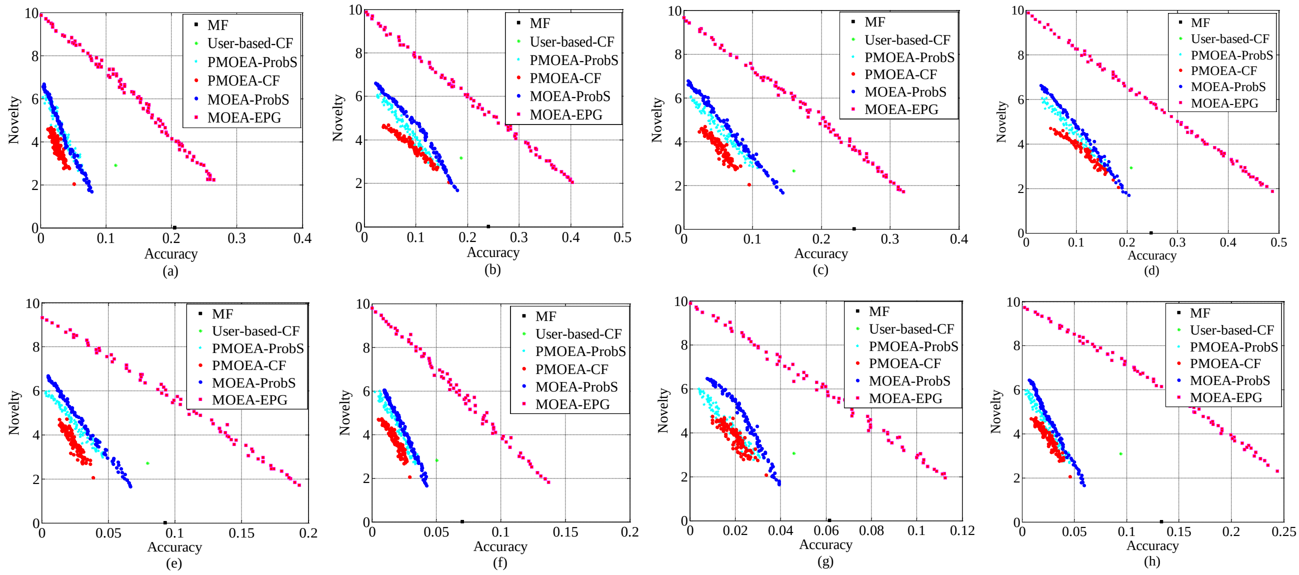


Fig. 9. Final recommendation results of comparison algorithms under the accuracy-novelty space for (a) Movielens 1, (b) Movielens 2, (c) Movielens 3, (d) Movielens 4, (e) Netflix 1, (f) Netflix 2, (g) Netflix 3 and (h) Netflix 4.

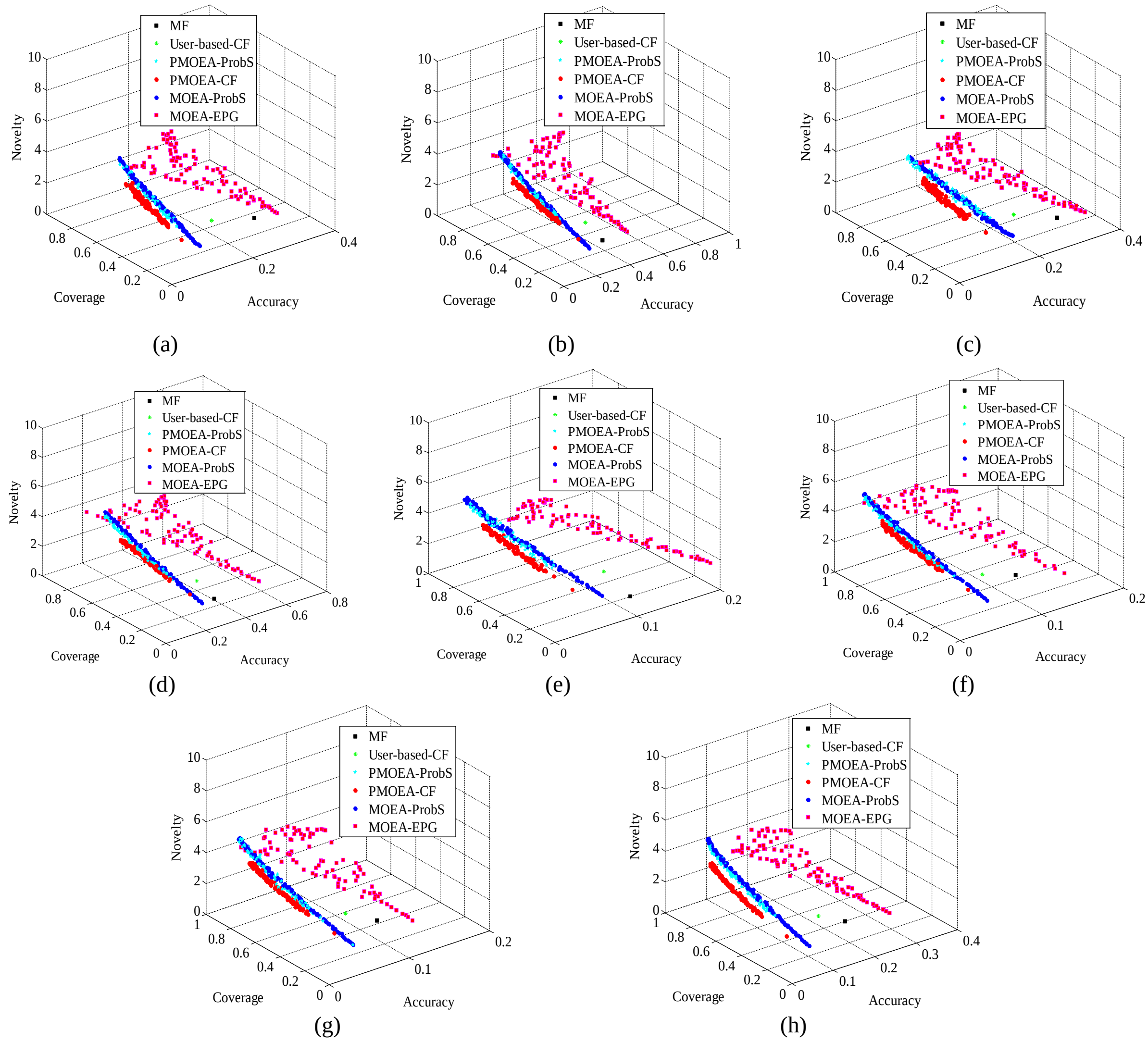


Fig. 10. Final recommendation results of comparison algorithms under the accuracy-coverage-novelty space for (a) Movielens 1, (b) Movielens 2, (c) Movielens 3, (d) Movielens 4, (e) Netflix 1, (f) Netflix 2, (g) Netflix 3, and (h) Netflix 4.

All the mean *HV* values and the standard deviation from 30 runs are recorded for performance comparisons in Table 5 and Table 6 for four data test sets (Movielens 1, Movielens 2, Movielens 3, and Movielens 4), where the best result is highlighted in boldface. Based on the experimental results, MOEA-EPG can find better solutions than MOEA-EPG-I and MOEA-EPG-II in the accuracy-coverage space, and it outperforms MOEA-EPG-III in the accuracy-novelty space. In particular, the accuracy extreme point and the novelty extreme point have a significant impact on the performance of the algorithm. They facilitate the efficiency in evolutionary search with better directions, and therefore the application of extreme point guided strategy is effective.

Table 5

HV Comparison results of MOEA-EPG, MOEA-EPG-I, and MOEA-EPG-II in the accuracy-coverage space.

Data sets	MOEA-EPG	MOEA-EPG-I	MOEA-EPG-II
Movielens 1	**3.203** $_{\mathbf{7.74e\text{-}02}}$	2.160 $_{6.98e\text{-}02}$	3.168 $_{1.44e\text{-}01}$
Movielens 2	**4.677** $_{\mathbf{3.46e\text{-}02}}$	3.409 $_{4.96e\text{-}02}$	4.536 $_{1.01e\text{-}01}$
Movielens 3	**3.343** $_{\mathbf{1.07e\text{-}02}}$	2.341 $_{7.24e\text{-}02}$	3.330 $_{9.94e\text{-}02}$
Movielens 4	**6.385** $_{\mathbf{5.62e\text{-}02}}$	4.503 $_{7.43e\text{-}02}$	6.158 $_{1.05e\text{-}01}$

Table 6

HV Comparison results of MOEA-EPG and MOEA-EPG-III in the accuracy-novelty space.

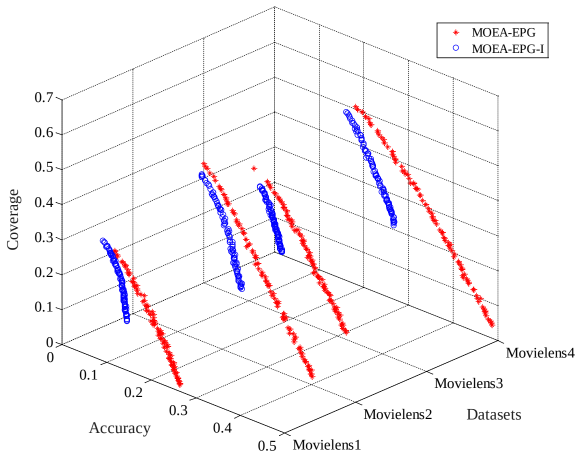
Data sets	MOEA-EPG	MOEA-EPG-III
Movielens 1	**36.89** $_{\mathbf{6.05e\text{-}02}}$	28.25 $_{3.71e\text{-}01}$
Movielens 2	**35.43** $_{\mathbf{7.83e\text{-}02}}$	23.79 $_{4.03e\text{-}01}$
Movielens 3	**36.10** $_{\mathbf{7.62e\text{-}02}}$	27.20 $_{4.77e\text{-}01}$
Movielens 4	**38.24** $_{\mathbf{1.22e\text{-}02}}$	24.11 $_{4.39e\text{-}01}$

Fig. 11 shows the final recommendation results with the best *HV* values regarding of MOEA-EPG with three variants (MOEA-EPG-I, MOEA-EPG-II and MOEA-EPG-III) on the accuracy-coverage and the accuracy-novelty spaces for all the data sets, respectively. Fig. 11(a) illustrates that MOEA-EPG has superior performance than MOEA-EPG-I in terms of the accuracy. It can be seen that the accuracy of the algorithm can be improved due to the introduction of the extreme accuracy point. Fig. 11 (b) shows that MOEA-EPG is better than MOEA-EPG-II on the coverage. Fig. 11 (c) confirms that MOEA-EPG gets a superior performance over MOEA-EPG-III regarding the novelty. Similar to the accuracy extreme point, the novelty extreme point also has a great impact on the performance of the algorithm. These results further validate the effectiveness of the extreme point guided method on accelerating the convergence of our algorithm.

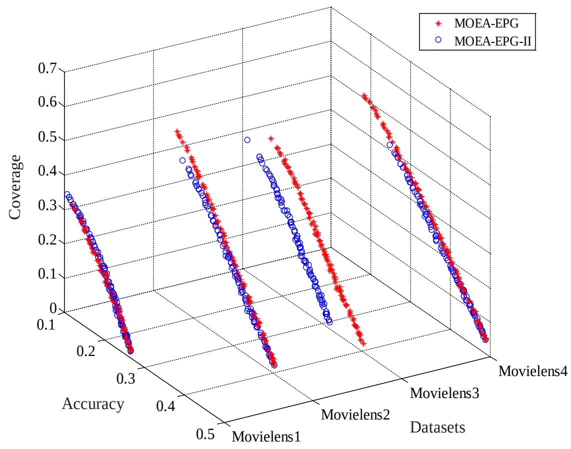
4.7 Impact of Parameters

In MOEA-EPG, the setting of parameter K (i.e., the similar ratio of users in Eq. (9)) may affect the performance of our algorithm. In this section, in order to study the impact of parameter K in MOEA-EPG, MOEA-EPG was run with different K values (i.e., 0.3, 0.5, 0.7, 0.9), while the other parameters in

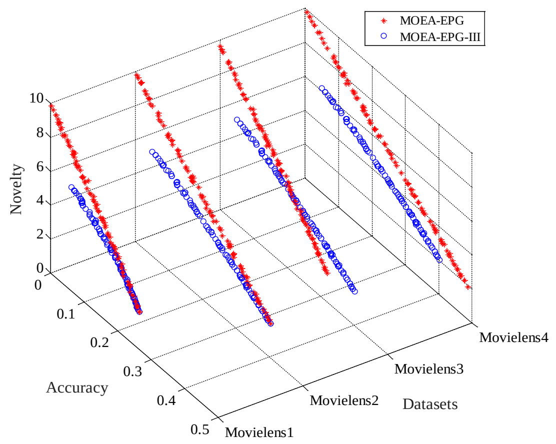
MOEA-EPG are kept the same as listed in Table 3. In Table 7, all the mean *HV* results and the standard deviations over 30 runs are recorded for performance comparisons. The best result for each data set is highlighted in boldface.



(a)



(b)



(c)

Fig. 11. Final solutions of MOEA-EPG and comparison algorithms under the accuracy-coverage space for (a) MOEA-EPG-I and (b) MOEA-EPG-II, and those under the accuracy-novelty space for (c) MOEA-EPG-III.

Table 7

HV Comparison results under different settings of *K* to MOEA-EPG on the accuracy-coverage space.

Data sets	*K*=0.3	*K*=0.5	*K*=0.7	*K*=0.9
Movielens 1	2.861 6.05e-02	**3.203 7.74e-02**	3.013 1.01e-01	2.914 7.30e-02
Movielens 2	3.944 3.06e-02	4.438 4.51e-01	3.334 4.03e-02	**4.592 2.86e-02**
Movielens 3	2.597 1.35e-02	**3.343 1.07e-01**	2.816 8.29e-02	2.387 6.77e-02
Movielens 4	5.659 3.46e-02	**5.985 5.62e-02**	5.461 4.52e-02	5.700 3.12e-02
Netflix 1	2.762 8.49e-02	**2.877 7.49e-02**	2.425 3.27e-02	1.961 1.28e-02
Netflix 2	**7.102 5.00e-02**	6.914 5.29e-02	6.434 3.58e-02	6.370 4.15e-02
Netflix 3	4.619 2.66e-02	**5.082 3.62e-02**	4.437 7.19e-02	3.696 3.55e-02
Netflix 4	1.791 4.82e-02	**1.975 1.05e-02**	1.594 1.45e-02	1.239 5.07e-02

Fig. 12 further gives the final results with the best *HV* values in data sets. From the comparison results, we observe that the setting of parameter K would affect the performance of MOEA-EPG. Moreover, the *HV* values of MOEA-EPG under K=0.5 are large in most data sets, whereas the performance of our algorithm becomes poor when K takes other values. Therefore, in this paper, we select 0.5 as the similar ratio of users.

4.8 Computational Complexity Analysis of MOEA-EPG

The computational complexity of MOEA-EPG can be easily deduced from its pseudo-code introduced in Algorithm 1. Its computational complexity is mainly determined by the evolutionary search in lines 13-24. In this evolutionary loop, the functions Objective_Func(.) in line 15 and Selection(.) in line 17 take a computational complexity with $O(N \cdot m^2 \cdot n)$, where N is the population size, m denotes the number users and n represents the number of items. According to [44], the functions Fast_Non_Dominated_Sort(.) in line 16 and Update(.) in line 21 own a computational complexity with $O(M \cdot N^2)$. Moreover, the functions Uniform_Crossover(.) and Mutation(.) in lines 18-19 have a computational complexity with $O(N \cdot m^2 \cdot n)$. Therefore, the overall computational complexity of MOEA-EPG is approximately $O(T \cdot N \cdot m^2 \cdot n)$ as $N \cdot m^2 \cdot n \gg N^2$ generally, where T is the predefined maximal generation number.

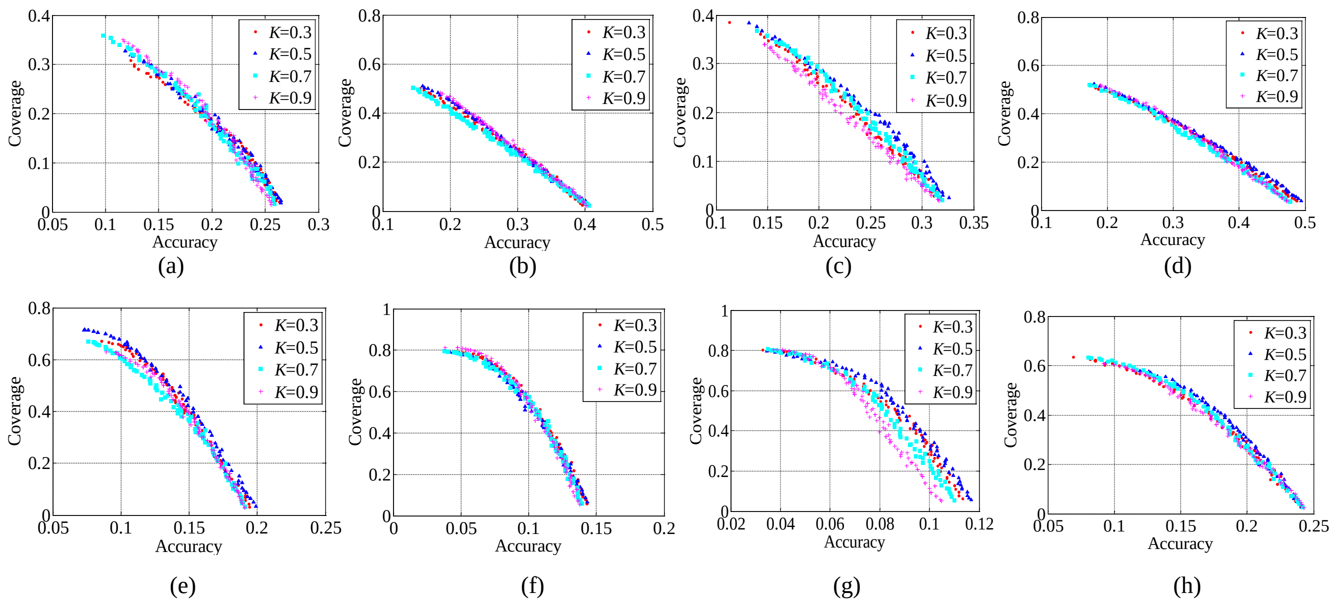


Fig. 12. Influence of similar ratio parameter K to MOEA-EPG under the accuracy-coverage space for (a) Movielens 1, (b) Movielens 2, (c) Movielens 3, (d) Movielens 4, (e) Netflix 1, (f) Netflix 2, (g) Netflix 3 and (h) Netflix 4.

Table 8

Computational complexities of all comparison recommendation algorithms

Comparison algorithms	Computational complexity
User-based-CF	$O(m^2 \cdot n)$
MF	$O(m^2 \cdot n)$
PMOEA-ProbS	$O(T \cdot N \cdot m^2 \cdot n)$
PMOEA-CF	$O(T \cdot N \cdot m^2 \cdot n)$
MOEA-ProbS	$O(T \cdot N \cdot m^2 \cdot n)$
MOEA-EPG	$O(T \cdot N \cdot m^2 \cdot n)$

Table 8 further gives the computational complexities of all comparison recommendation algorithms. User-based-CF [31] and MF [32] have a computation complexity with $O(m^2 \cdot n)$ while our computational complexity analysis is given above with $O(T \cdot N \cdot m^2 \cdot n)$. Since MOEA-ProbS and PMOEA adopt the same NSGA-II framework except for the crossover operators, they share the same computational complexity (i.e., $O(T \cdot N \cdot m^2 \cdot n)$) with MOEA-EPG.

Table 9

Average execution times in seconds of all the MOEAs based recommendation algorithms over 30 independent trials.

Data sets	MOEA-ProbS	PMOEA-ProbS	PMOEA-CF	MOEA-EPG
Movielens 1	2.584e+03	6.386e+03	5.080e+03	3.937e+03
Movielens 2	2.793e+03	7.859e+03	6.627e+03	4.532e+03
Movielens 3	2.630e+03	7.096e+03	6.163e+03	4.227e+03
Movielens 4	2.887e+03	7.663e+03	1.338e+04	4.963e+03
Netflix 1	2.718e+03	7.149e+03	6.018e+03	5.539e+03
Netflix 2	2.960e+03	7.542e+03	6.548e+03	5.707e+03
Netflix 3	2.964e+03	8.128e+03	7.633e+03	6.532e+03
Netflix 4	2.237e+03	5.843e+03	5.639e+03	5.431e+03

To show the actual running efficiency of all the MOEAs based recommendation algorithms, Table 9 records their average execution times over 30 independent trials. From the results in Table 9, it can be found that although they share the same computational complexity, the actual running times are different due to the running of different crossover operators and some extra strategies. For example, MOEA-ProbS shows the fast execution efficiency, as it adopts the original NSGA-II [44] on the recommendation system. Since MOEA-EPG has some extra processes, such as the calculation for the three extreme points and the selection of similar parents for crossover, its execution times are a little longer than that of MOEA-ProbS on all the data sets. It is noted that PMOEA-ProbS and PMOEA-CF consume the longest execution times, as they need to run the crossover operators by multiple times.

5. Conclusions

In this paper, we presented a multi-objective personalized recommendation algorithm MOEA-EPG based on extreme point guided evolutionary computation. In this algorithm, we modeled the personalized recommendation as an MOP with the three conflicting objectives, i.e., the accuracy, diversity and novelty. The accuracy of recommendations considers whether the predicted ratings are similar to the true ones of users, the diversity reflects the coverage on recommendation, while the novelty evaluates the capability of RSs to recommend unknown items to users. In MOEA-EPG, three extreme points on accuracy, diversity and novelty were incorporated to guide its evolutionary search toward the PF, and a novel crossover operator was designed to better satisfy the demands of users. The experimental results demonstrated the superior performance of the proposed MOEA-EPG over two state-of-the-art recommendations (i.e., User-based-CF [31] and MF [32]) and two competitive MOEAs for RSs (i.e., MOEA-ProbS [27] and PMOEA [30]), in terms of the accuracy, diversity and novelty. Moreover, the effectiveness of the extreme point guided method was also validated to accelerate the convergence speed toward the PF and the sensitivity analysis on parameters was conducted to have a robust performance for MOEA-EPG.

Our future work will focus on the performance metrics for the recommendation algorithms and further study other nature-inspired optimization algorithms (e.g., particle swarm optimization and ant optimization algorithms) for RSs.

Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant 61876110, 61803269, and 61672358, the Joint Funds of the National Natural Science Foundation of China under Key Program Grant U1713212, the Natural Science Foundation of Guangdong Province under grant 2017A030313338, Shenzhen Technology Plan under Grant JCYJ20170817102218122 and JCYJ20170302154032530, and CONACyT under Grant 221551.

Reference:

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

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

[3] 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.

[4] K. Wei, J. Huang, and S. Fu, “A survey of e-commerce recommender systems,” in *Proceeding of International Conference on Service Systems and Service Management*, Chengdu, China, 2007.

[5] J. B. Schafer, J. Konstan, and J. Riedl, “E-commerce recommendation applications,” *Data Mining and Knowledge Discovery*, vol. 5, no. 1-2, pp. 115-153, 2001.

[6] J.-C. Wang and C.-C. Chiu, “Recommending trusted online auction sellers using social network analysis,” *Expert Systems with Applications*, vol. 34, no. 3, pp. 1666-1679, 2008.

[7] L. Ma, J. Li, Q. Lin, M. Gong, C.A. Coello Coello, and Z. Ming, “Reliable Link Inference for Network Data With Community Structures,” *IEEE Transactions on Cybernetics*, doi: 10.1109/TCYB.2018.2860284, 2018.

[8] N. Divya, S. Sandhiya, and D.R. Anita Sofia Liz, “A collaborative filtering based recommender system using rating prediction,” *International Journal of Pure and Applied Mathematics*, vol. 119, no. 10, pp. 1-7, 2018.

[9] K. Soni, R. Goyal, B. Vadera, and S. More, “ A three way hybrid movie recommendation system,” *International Journal of Computer Applications*, vol. 160, no. 9, pp. 29-32, 2017.

[10] O. R. Zaiane, “Building a recommender agent for e-learning systems,” in *Proceeding of International Conference on Computers in Education*, Alberta, Canada, 2002.

[11] J. Lu, “A personalized e-learning material recommender system,” in *Proceeding of 2nd International Conference on Information Technology and Applications*, Harbin, China, 2004.

[12] C. Romero, S. Ventura, A. Zafra, and P. d. Bra, “Applying web usage mining for personalizing hyperlinks in web-based adaptive educational systems,” *Computers & Education*, vol. 53, no. 3, pp.

828-840, 2009.

[13] J. Beel, S. Langer, and M. Genzmehr, “Research paper recommender system evaluation: a quantitative literature survey,” in *Proceeding of the International Workshop on Reproducibility and Replication in Recommender Systems Evaluation*, Hong Kong, China, 2013.

[14] J. Fan, W. Pan, L. Jiang, “An improved collaborative filtering algorithm combining content-based algorithm and user activity,” in *Proceeding of IEEE International Conference on Big Data and Smart Computing*, Bangkok, Thailand, 2014.

[15] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, “Item-based collaborative filtering recommendation algorithms,” in *Proceeding of 10th International Conference on World Wide Web*, Hong Kong, China, 2001.

[16] H. Drachsler, H. G. K. Hummel, and R. Koper, “Personal recommender systems for learners in lifelong learning: requirements, techniques and model,” in *International Journal of Learning Technology*, vol. 3, no. 4, pp. 404-423, 2008.

[17] X. Su, T. M. Khoshgoftaar, “A survey of collaborative filtering techniques,” *Advances in artificial intelligence* vol. 2009, no. 421425, pp. 1-19, 2009.

[18] G. Adomavicius, Y. Kwon, “Improving aggregate recommendation diversity using ranking-based techniques,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, pp. 896-911, 2012.

[19] R. Burke, “Knowledge-based recommender systems,” *Encyclopedia of Library and Information Systems*, vol. 69, pp. 175-186, 2000.

[20] J. Bobadilla, F. Ortega, A. Hernando, and J. Alcalá, “Improving collaborative filtering recommender system results and performance using genetic algorithms,” *Knowledge-Based Systems*, vol. 24, no. 8, pp. 1310-1316, 2011.

[21] 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, p. 123008, 2009.

[22] M. Zhang and N. Hurley, “Avoiding monotony: Improving the diversity of recommendation lists,” in *Proceeding of ACM Conference on Recommender Systems*, New York, USA, 2008.

[23] 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 of the United States of America*, vol. 107, no. 10, pp. 4511–4515, 2010.

[24] G. Adomavicius and Y. Kwon, “Improving aggregate recommendation diversity using ranking-based techniques,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 896–911, 2012.

[25] A. Ropponen, R. Ritala, and E. N. Pistikopoulos, “Optimization issues of the broke management system in papermaking,” *Computing & Chemical Engineering*, vol. 35, no. 11, pp. 2510-2520, 2011.

[26] T. Jambor, J. Wang, “Optimizing multiple objectives in collaborative filtering,” in *Proceedings of the fourth ACM conference on Recommender systems*, Bilbao, Spain, 2010.

[27] Zuo Y, Gong M, Zeng J, Ma L, and Jiao L, “Personalized recommendation based on evolutionary

multi-objective optimization," *IEEE Computational Intelligence Magazine*, vol. 10, no. 1, pp. 52-62, 2015.

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

[29] Q. Zhang and H. Li, "MOEA/D: a multi-objective evolutionary algorithm based on decomposition," *IEEE Transactions on Evolutionary Computation*, vol. 11, no. 6, pp. 712-731, 2007.

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

[31] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Transactions on Information Systems*, vol. 22, no. 1, pp. 5–53, 2004.

[32] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[33] 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.

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

[35] Q. Lin, G. Jin, Y. Ma, and C. A. Coello, "A Diversity-Enhanced Resource Allocation Strategy for Decomposition-Based Multiobjective Evolutionary Algorithm," *IEEE Transactions on Cybernetic*, 2017, in print, dol.10.1109.

[36] Q. Zhu, Q. Lin, W. Chen, and K. C. Wong, "An External Archive-Guided Multiobjective Particle Swarm Optimization Algorithm," *IEEE Transactions on Cybernetic*, vol. 47, no. 9, pp. 2794–2808, 2017.

[37] P. A. N. Bosman and D. Thierens, "The balance between proximity and diversity in multiobjective evolutionary algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 174–188, 2003.

[38] N. Chang and T. Terano, "Improving the Performance of User-Based Collaborative Filtering by Mining Latent Attributes of Neighborhood," in *Proceeding of 2014 International Conference on Mathematics and Computers in Sciences and in Industry*, Varna, Bulgaria, 2014.

[39] H. Liu, Z. Hu, A. Mian, H. Tian, and X. Zhu, "A new user similarity model to improve the accuracy of collaborative filtering, " *Knowledge-Based Systems*, vol. 56, pp. 156-166, 2014.

[40] T. Zhou, J. Ren, M. Medo, and Y. C. Zhang, "Bipartite network projection and personal recommendation," *Physical Review E*, vol. 76, no. 4, p. 046115, 2007.

[41] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Journal of the Royal Statistical Society. Series C* (*Applied Statistics*), vol. 28, no. 1, pp. 100-108, 1979.

[42] T. V R Himabindu, V. Padmanabhan, V. R. Kagita, and A. K Pujari, "Recommender system algorithms: A comparative analysis based on monotonicity," in *Proceeding of 2015 Eighth*

International Conference on Advances in Pattern Recognition, Kolkata, India, 2015.

[43] 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.

[44] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGAII,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[45] A. Khan and A. R. Baig, “Multi-objective Feature Subset Selection using Non-dominated Sorting Genetic Algorithm,” *Journal of Applied Research and Technology*, vol. 13, no.1, pp. 145-159, 2015.

[46] Y. Lai, J. Liu, and G. Yang, “K-Means Clustering Analysis Based on Genetic Algorithm,” *Computer Engineering*, vol. 34, no. 20, pp. 200-202, 2008.

[47] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

[48] 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*, New York, 2012.

[49] M. T. Ribeiro, A. Lacerda, 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.