Search Based Recommender System Using Many-objective Evolutionary Algorithm

Li, Bingdong; Qian, Chao; Li, Jinlong; Tang, Ke; Yao, Xin

DOI:
10.1109/CEC.2016.7743786

Search Based Recommender System Using Many-Objective Evolutionary Algorithm

Bingdong Li*, Chao Qian*, Jinlong Li*, Ke Tang*, and Xin Yao†*
*USTC-Birmingham Joint Research Institute in Intelligent Computation and Its Applications (UBRI),
School of Computer Science and Technology,
University of Science and Technology of China,
Hefei, China, 230027
†The Centre of Excellence for Research in Computational Intelligence and Applications (CERCIA),
School of Computer Science,
the University of Birmingham,
Birmingham B15 2TT, U.K.
Emails: lbingd17@mail.ustc.edu.cn; chaoqian@ustc.edu.cn; jlli@ustc.edu.cn; ketang@ustc.edu.cn; x.yao@cs.bham.ac.uk

***Abstract*—With the explosively increase of information and products, recommender systems have played a more and more important role in the recent years. Various recommendation algorithms, such as content-based methods and collaborative filtering methods, have been proposed. There are a number of performance metrics for evaluating recommender systems, and considering only the precision or diversity might be inappropriate. However, to the best of our knowledge, no existing work has considered recommendation with many objectives. In this paper, we model a many-objective search-based recommender system and adopt a recently proposed many-objective evolutionary algorithm to optimize it. Experimental results on the Movielens data set demonstrate that our algorithm performs better in terms of Generational Distance (GD), Inverted Generational Distance (IGD) and Hypervolume (HV) on most test cases.**

I. Introduction

The last few years have witnessed explosive data which is far beyond the users' capability to extract the useful information [1]. For example, electronic retailers, such as Amazon, Alibaba, have a huge amount of products/items and try to meet a large variety of users with different interests; content providers, such Netflix, lastfm, and Douban, intend to recommend huge amount of movies, songs, and books that satisfy each user's specific interest [2]. The dilemma between the large amount of products and the relative low consuming ability of the users has been a major challenge for these companies. In order to deal with the dilemma, recommender systems are proposed and become a hot topic in research community[3], [4], [5]. Generally, recommender systems try to provide personalized recommendation for all users with various interests. Broadly speaking, recommender systems can be categorized into two major classes: content-based methods and collaborative filtering methods.

The content-based methods [6] try to create user profiles or product profiles based on their natural attributes. For example, a user profile might contain the age or gender of the user, while a book profile might contain the price, writer, and genre of the book. Since they have established the archive for all users or products, making recommendations according to the profile information seems to be direct by matching the users and the products. Newly entered users or the inactive users who have not contributed enough ratings. Improving the recommendation results for new users and new items is a challenging task, which is referred to as cold start [7].With the collected profiles, the content-based methods can alleviate the cold-start problem to some extent. Yet, the advantage comes with the cost of collecting the information for creating the profiles, which might not be trivial work since some domain-specific knowledge might be needed.

The collaborative filtering methods, on the other hand, provide the users with recommendation lists based on the users' behavior history [8], [9]. Here, a user's behavior can be his clicks, ratings, buying logs, or comments. The advantage of this scheme is that the recommender system does not bear the burden of collecting information for user profiles and product profiles. Although when confronted with a new user which does not provide enough behavior information, the system might need more time to warm up before it is able to provide a promising recommendation list for the new user. This phenomenon is often referred to as the cold-start problem. Generally speaking, there are two sub-classes of collaborative filtering methods: the nearest-neighborhood-based methods [10] and the model-based methods [6]. The nearest-neighborhood-based methods build the system based on the similarities between users and/or items according to the obtained user-item ratings [10], [11], [12], [13]. The model-based methods, on the other hand, perform the recommendation task by mapping both the users and the items into a low-dimensional latent factor space [6], [14]. Singular Value Decomposition (SVD) is one popular technique for these methods.

Recently, a series of hybrid methods are proposed to combine the different classes/sub-classes of recommender systems. Content-based Collaborative Filtering (CCF) [15] combines the content-based approach and the collaborative

filtering approach by taking advantage of the rich contexts of new topic recommendation and exploiting the long-tail users. Factored Item Similarity Model (FISM) combines the nearest-neighborhood-based method and the model-based method by learning the similarities between items according to the latent factors [9]. In addition to combining different methods, information such as item usage context [16], expert opinion [17], and social network information [18], can also be taken into account during the recommendation.

It has been widely recognized that considering only the precision of the recommender system might not be enough for evaluating a recommender system [19], [20]. Other performance metrics for recommender systems, such as diversity, novelty, coverage, and serendipity, also need to be taken into account [21], [16], [22]. The seemingly confliction between different kinds of performance metrics make the recommendation task a multi-objective optimization problem naturally [20], [23]. There has been a number of works on multi-objective optimization based recommender systems. Ribeiro et al. [24] selected Pareto-efficient items according to multiple recommender algorithms and tried to optimize the weighted combination of multiple recommender systems using multi-objective evolutionary algorithm. Zuo et al. [20] proposed a personalized recommender system by optimizing the accuracy and coverage of the recommendation lists for all users. Agarwal et al. [25] considered both the clicks and the post-click downstream utilities and proposed a multi-objective programming approach based on a constrained optimization framework to address the problem. Rodriguez et al. [26] also introduced to consider multiple objectives in recommender systems. However, similar to [25], the approach proposed in [26] is a constrained optimization method.

Although there has been much related work on multi-objective recommender systems, to the best of our knowledge, no existing work has studied personalized recommendation with many objectives (more than three objectives). In this paper, we model top-N personalized recommendation as a many-objective optimization problem and address the problem with a recently proposed algorithm named Stochastic Ranking Algorithm (SRA) [27]. The experimental results on the Movielens data set demonstrate the effectiveness of our algorithm when compared with three many-objective evolutionary algorithms.

The main contributions of this paper are listed as follows:

- A many-objective search based recommender system is proposed based on performance metrics of recommender systems.
- A recently proposed many-objective evolutionary algorithm named Stochastic Ranking Algorithm (SRA) is employed to tackle the search problem.
- The Stochastic Ranking Algorithm (SRA) is compared with three popular many-objective evolutionary algorithms on the Movielens data set. Experimental results show that SRA performs better in terms of GD and IGD.

The rest of the paper is organized as follows: The many-objective search-based recommender system is presented in Section II. Section III describes our approach, which is based on the Stochastic Ranking Algorithm (SRA). Section IV is devoted to introduce the experimental studies. The last section concludes this paper and points out some future research directions.

II. Many-objective Search-based Recommender System

A. Many-objective Optimization

Without loss of generality, a minimization multi-objective optimization problem can be stated as follows [28]:

$$\begin{aligned} &minimize\ \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))^T \\ &subject\ to\ \mathbf{x} \in \Omega \end{aligned} \tag{1}$$

where $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ is the decision vector, $\mathbf{F}$: $\Omega \rightarrow \Lambda$ is the objective vector, Ω is the (nonempty) *decision space*, Λ is the *objective space*. m ($m \geq 2$) is the number of objectives, n is the number of decision variables. Usually, the objective space Λ is a subset of R^m. Multi-objective problems with more than 3 objectives are often called many-objective problems.

The Pareto dominance relation is critical in multi-objective optimization, which is defined as follows:

$Pareto\ dominance$ [29]: Given the objective vectors $\mathbf{F}(\mathbf{x})$, $\mathbf{F}(\mathbf{y}) \in R^m$ of two feasible solutions $\mathbf{x}$, $\mathbf{y} \in \Omega_f$, $\mathbf{x}$ is said to *dominate* $\mathbf{y}$ (denoted as $\mathbf{x} \prec \mathbf{y}$) if and only if $\forall i \in \{1, 2, ..., m\}, f_i(\mathbf{x}) \leq f_i(\mathbf{y})$ and $\exists j \in \{1, 2, ..., m\}, f_j(\mathbf{x}) < f_j(\mathbf{y})$.

$Pareto\ optimal\ solution$: A solution $\mathbf{x}^* \in \Omega_f$ is said to be a Pareto optimal solution if and only if $\nexists \mathbf{x} \in \Omega_f$, $\mathbf{x} \prec \mathbf{x}^*$.

$Pareto\ set$: The Pareto set PS consists of all the Pareto optimal solutions, denoted as $PS = \{\mathbf{x} \in \Omega_f | \nexists \mathbf{y} \in \Omega_f, \mathbf{y} \prec \mathbf{x}\}$.

$Pareto\ Front$: The image of the Pareto set in the objective space is called the Pareto Front (PF).

The goal of multi-objective optimization is to obtain an approximation solution set A of the problem which satisfies two sub-goals: 1) A is pushed towards the Pareto front as close as possible. 2) The spread of objective vectors of the solutions in A is as diverse as possible.

B. Performance Metrics for Recommender Systems

There has been a series of work on the performance evaluation of recommender systems [30], [23], [31]. Maksai et al. categorized the performance metrics into five different groups [23], which are described as follows:

1) Accuracy/Error metrics indicate the precision of the recommendations or predictions in terms of users' interest.
2) Diversity metrics measure the dissimilarities between items in the recommendation list of one user ("intra-list" diversity), or the lists of different users ("extra-list" diversity).
3) Novelty metrics measure to what extent can a recommender system provide new items to a user.
4) Coverage measures the range that a recommendation list can cover.
5) Serendipity shows the quality of recommending both interesting and new items to the users.

C. Many-objective Search-based Recommender System

In this paper, we study the top-N recommendation problem described as follows. Given a set of users denoted as $U = \{u_1, u_2, \ldots, u_{n_u}\}$, a set of items denoted as $O = \{o_1, o_2, \ldots, o_{n_o}\}$, a $n_u * n_o$ rating matrix R where each row corresponds to a user's ratings to all items and each column corresponds to an item's ratings from all users (most elements of the matrix are un-rated, denoted as NaN), and recommendation list length N, a recommender algorithm is supposed to provide n_u recommendation lists. Each list corresponds to a user and consists of N items recommended to the user. The pseudocode of the main framework of many-objective search-based recommender system is shown in Algorithm 1.

Algorithm 1: Main Framework of Many-objective Search-based Recommender System

input : user set $U = \{u_1, u_2, \ldots, u_{n_u}\}$,
object set $O = \{o_1, o_2, \ldots, o_{n_o}\}$,
$n_u * n_o$ rating matrix R,
recommendation list length N
population size $popSize$
output: $n_u * N$ recommendation matrix A

1 Invoke some rating estimation methods to predict the missing ratings and complete the rating matrix: $\hat{R} \leftarrow R$
2 Initialize population P with $popSize$ solutions
3 Evaluate the solutions using matrix $\hat{R}$
4 Call a many-objective optimization algorithm to search for better solutions

As shown in the pseudocode, we tackle the recommendation task in a two-step manner as [20]. First, some rating prediction method is invoked to predict the missing ratings and complete the rating matrix: $\hat{R} \leftarrow R$. In this paper, the ProbS algorithm [13] is introduced to tackle the prediction task. Then, with the full rating matrix, it is possible to compute the values of the performance metrics. As a start-up, we choose one metric from each of the metric groups introduced above. Thus, the goal of recommendation is to maximize the F1 value, maximize the diversity metric value, maximize the novelty metric value, maximize the coverage metric value, and maximize the serendipity metric value. Those metrics are defined as follows:

1) Accuracy/Error: As shown in Table I, an item falls into one of the four categories. Given the recommendation list $L_N(u_i)$ for the i-th user u_i, the precision P, recall R, and the F1 value are defined as [5], [32]:

$$P(L_N(u_i)) = \frac{\#tp}{\#tp + \#fp} \tag{2}$$

$$R(L_N(u_i)) = \frac{\#tp}{\#tp + \#fn} \tag{3}$$

$$F1(L_N(u_i)) = \frac{2P(L_N(u_i))R(L_N(u_i))}{P(L_N(u_i)) + R(L_N(u_i))} \tag{4}$$

The average F1 value for the recommendation lists of all users are used as the first objective f_1.

TABLE I
FOUR CATEGORIES OF POSSIBLE RESULTS WHEN AN ITEM IS RECOMMENDED OR NOT RECOMMENDED TO A USER

	Recommended	Not recommended
Interested	true-positive (tp)	false-negative (fn)
Not interested	false-positive (fp)	true-negative (tn)

2) Diversity: Zhou et al. proposed a diversity metric called personalization [12], which is defined as:

$$h(L_N(u_i), L_N(u_j)) = 1 - \frac{q(L_N(u_i), L_N(u_j))}{N} \tag{5}$$

where $q(L_N(u_i), L_N(u_j))$ is defined as the number of different items in the recommendation lists for the i-th and j-th user. The average $h(L_N(u_i), L_N(u_j))$ for all pairs of recommendation lists is used as the second objective f_2.

3) Novelty: The degree of an item o_i (denoted as d_i) is the times it has been rated/consumed, i.e. the number of non-NaN elements in the i-th column of R. Thus the chance that a random user has rated the item is d_i/n_u. The self-information [12] of the item is given by I_i,

$$I_i = log_2\left(\frac{n_u}{d_i}\right) \tag{6}$$

where d_i is the degree of the item. The mean self-information of all items in the top-N list of a user is used to measure the self-information of the list. The average self-information of all users' recommendation lists is used as the third objective f_3.

4) Coverage: The coverage-at-top-N is defined as the number of different items in all users' top N lists [22]:

$$coverage\text{-}in\text{-}top\text{-}N = \left| \bigcup_{u_i \in U} L_N(u_i) \right| \tag{7}$$

In this paper, we divide $coverage\text{-}in\text{-}top\text{-}N$ with the number of items and the number of users and use this value as the forth objective f_4.

5) Serendipity: In order to measure the serendipity of a recommendation list, we use SRDP introduced in [21]. A recommended item is considered unexpected if it is recommended to the user by a primitive recommendation model PM. Thus the unexpected set is defined as

$$UNEXP(u_j) = L_N(u_j) \setminus PM(u_j) \tag{8}$$

where $L_N(u_j)$ is the recommendation list for the j-th user given by a recommender system (the targe algorithm to be measured), $PM(u_j)$ is the recommendation list given by the primitive recommendation model PM. Only the useful items of the unexpected recommendations are effective. The usefulness of the i-th item o_i to the j-th user u_j is defined as :

$$u(o_i, u_j) = \begin{cases} 1 & \text{if } R(i,j) > th \\ 0 & \text{otherwise} \end{cases}$$

where th is the threshold of the user's interest. Recommending the i-th item o_i to the j-th user u_j is said to be effective if the

predicted rating is above the threshold. Otherwise, the item is useless to the user. In our implementation, th is set to 0.6. The serendipity metric [21] for the recommendation list of the j-th user is defined as follows:

$$SRDP(L_N(u_j)) = \frac{\sum_{o \in UNEXP(u_j)} u(o, u_j)}{N} \tag{9}$$

The mean SRDP value of the recommendation lists for all the users is used as the fifth objective f_5.

III. Search Based Recommender System Using SRA

A. SRA Many-objective optimization

Stochastic Ranking Algorithm (SRA) [27] is a many-objective evolutionary algorithm based on multiple indicators and aggregation functions. It uses stochastic ranking technique [33] to balance the search bias of different indicators and maintains an archive based on weight vectors. The pseudocode of the main loop of SRA is shown in Algorithm 2. First, the population and the archive are initialized using ramdom solutions. The weight vectors are initialized to representing search directions. During each iteration, an offspring population is generated using the population and archive and then merged with the parent population. After that, the Stochastic Ranking based Enviromental Selection (SRES) is invoked to select $popSize$ solutions from $2 * popSize$ solutions, where the stochasting ranking technique is used to rank solutions according to multiple indicators. At the end of each iteration, the Archive Update Procedure (AUP) is called to update the archive using the penalty-based boundary intersection (PBI) [34] fitness function.

Algorithm 2: Main Loop of SRA

input : a many-objective problem
output: approximation set A_{out}

Initialize population, archive, and input weight vectors.
while $t <$ *MaxGen* **do**
  Create a new population Q_t from A_t and P_t
  Evaluate the new generation Q_t
  Merge the current population and the offspring population: $U_t \leftarrow P_t \cup Q_t$
  Compute the indicator values $I_1(\mathbf{u}_i)$ and $I_2(\mathbf{u}_i)$ for all $\mathbf{u}_i \in U_t$
  Stochastic ranking based environmental selection: $P_{t+1} \leftarrow$ SRES(U_t)
  Update archive using selected population : $A_{t+1} \leftarrow$ AUP(P_{t+1})
  $t \leftarrow t + 1$
end
Return the set A_{out} which consists of the non-dominated solutions of A_{MaxGen}

B. Solution Representation

Here the many-objective optimization algorithm treats the problem as a black-box function. The solution is denoted as a matrix $A = \{a_{i,j}\}$ $(1 \leq a_{i,j} \leq n_o)$, where the $a_{i,j}$ is the id of the j-th object recommended to the i-th user u_i, $1 \leq i \leq n_u$, $1 \leq j \leq N$, . The i-th row in A corresponds to the top-N recommended objects for the i-th user u_i, $1 \leq i \leq n_u$. In our implementation, the matrix is transformed into a integer vector with $n_u * N$ values.

C. Solution Variation

Since the solution is a integer vector with $n_u * N$ values, we use classic one point crossover operator and bit flip mutation operator from the jMetal package [35] to create new solutions. During one point crossover, two parent solutions are selected and one crossover point is randomly chosen. Then the two parts of the two parent solutions are interchanged to create a new solution. For bit flip mutation operator, each value of the decision variable is randomly changed to another feasible integer value with a mutation rate p_m.

IV. Experimental Studies

A. Data Set

In order to test the effectiveness of the stochastic ranking algorithm, the Movielens benchmark data set (http://www.grouplens.org/) [36] is studied. According to [20], we divide the users into four clusters based on the cosine distance of users using the k-means clustering algorithm. Thus the data set results in four sub-problems, denoted as SBRS1p5d—SBRS4p5d, including 200, 258, 227, 258 users respectively.

B. Peer Algorithms

In order to test the algorithm, the following three many-objective evolutionary algorithms (implemented in the jMetal framework [35] on a 4-core 2.50 GHz Intel Core i5-3210M CPU with 3.7 Gb RAM) are considered:

1) NSGA-III [37] is a hybrid algorithm based on dominance and decomposition. The dominance-based sorting pushes the population towards the Pareto front, while the weight vector based niching maintains diversity of the population. It has been shown that the algorithm performs well on many-objective benchmark problems as well as real-world test problems [37].
2) MOEA/D [34] is a popular decomposition based algorithm. In MOEA/D, a many-objective problem is decomposed into a series of single-objective sub-problems. Since it adopts the neighborhood definition of sub-problems, the convergence of the algorithm is quite promising. The diversity is maintained using the weight vector based search directions.
3) IBEA [38] is an many-objective evolutionary algorithm based on the $I_{\epsilon+}$ indicator. It is the first indicator-based algorithm and performs well in terms of convergence quality. IBEA has been tested on various benchmark problems [39].

C. Parameter Settings

The general and algorithm-specific parameter settings are summarized as follows:

1) Population size. The population size for the algorithm (MOEA/D, NSGA-III, SRA, IBEA) is set to 70 for computational efficiency.
2) Reproduction operators. The one point crossover operator and bit flip mutation operator are used for reproducing offspring solutions. The mutation probability is set to 0.1, and the crossover probability is set to 1.0.
3) The neighborhood size is set to 20 and the maximum replacement number is set to 2 for MOEA/D and SRA during the Archive Update Procedure (UAP).
4) Termination criterion. All algorithms are allowed for a maximum of 35,000 fitness evaluations for all the problem instances.
5) Number of runs. For all the problems instances, all the algorithms are repeated 20 times independently.
6) Statistical test. In order to test the difference of algorithms, the Wilcoxon rank sum test [40] (0.05 significance level) is applied for analysis.
7) The parameter p_c in SRA is set to $[0.4, 0.6]$ based on some preliminary experimental results in [27].
8) The recommendation list length is set to 10.

D. Performance Metrics

In order to compare SRA with other many-objective evolutionary algorithms, the following performance metrics are chosen for comparison.

- Inverted Generational Distance (IGD) [41], [42] is a distance-based metric, which is defined as follows:

$$IGD(A, PF^{'}) = \frac{1}{|PF'|}\Big(\sum\nolimits_{i=1}^{|PF^{'}|} distance(\mathbf{p}_i, A)^p\Big)^{\frac{1}{p}} \quad (10)$$

  where $A = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_{|A|}\}$ is the output approximation set of a many-objective evolutionary algorithm, $PF^{'} = \{\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_{|PF'|}\}$ is a subset of the true Pareto front. In our implementation, since the true Pareto front is unknown, the non-dominated solutions from all runs of all algorithms are merged into a solution set and used as $PF^{'}$. The distance between a solution $\mathbf{p}$ and a solution set A is defined as follows:

$$distance(\mathbf{p}, A) = \min_{\mathbf{a} \in A} distance(\mathbf{p}, \mathbf{F(a)}) \quad (11)$$

  In our experiment, p in Equation (10) is set to 1 and the Euclidean distance is used to compute distance between solutions. The IGD metric can show the quality of the solution set in terms of both convergence and diversity. A smaller IGD value indicates a better solution set.
- Generational Distance (GD) [43] is also a distance-based metric for many-objective optimization. It is defined as follows:

$$GD = \frac{1}{|A|}\Big(\sum\nolimits_{i=1}^{|A|} distance(\mathbf{F(a}_i), PF^{'})^p\Big)^{\frac{1}{p}} \quad (12)$$

  where A and $PF^{'}$ are described above. Roughly speaking, GD indicates how good the convergence quality of approximation set is.
- Hypervolume (HV) is defined as the volume of solutions that solution set dominates [44], [45]. Given a reference point $\mathbf{z}^{\dagger}$ and an approximation set A, $HV(\mathbf{z}^{\dagger}, A)$ is defined as follows:

$$HV(\mathbf{z}^{\dagger}, A) = L(\bigcup_{\mathbf{a} \in A} \{\mathbf{b} \in \Lambda | \mathbf{a} \prec \mathbf{b} \prec \mathbf{z}^{\dagger}) \quad (13)$$

  where L is the Lebesgue measure, $\mathbf{z}^{\dagger}$ is the reference point in the objective space. It measures both convergence and diversity of an approximation set. The higher the hypervolume value is, the better the approximation set is. In our experiments, the population is first normalized using the $1.0 * \mathbf{z}^{nadir}$ [1]. After that, the hypervolume is computed using $(1.0, \ldots, 1.0)^T$ as the reference point.

E. Experimental Results

First, the Wilcoxon rank sum test (0.05 significance level) is shown in Table II. Each element in the table shows the pairwise win-tie-loss counts of rows against columns. As the table shows, SRA performs better than the peer algorithm generally.

TABLE II
THE STATISTICAL TEST RESULTS ON SBRS1P5D-SBRS4P10D IN TERMS OF GD, IGD, AND HV. EACH ELEMENT IN THE TABLE SHOWS THE PAIRWISE WIN-TIE-LOSS COUNTS OF ROWS AGAINST COLUMNS.

GD	MOEA/D	IBEA	SRA
NSGAIII	0-0-4	0-0-4	0-0-4
MOEA/D		0-0-4	0-1-3
IBEA			2-2-0
IGD	MOEA/D	IBEA	SRA
NSGAIII	0-0-4	4-0-0	0-0-4
MOEA/D		4-0-0	0-0-4
IBEA			0-0-4
HV	MOEA/D	IBEA	SRA
NSGAIII	4-0-0	4-0-0	0-0-4
MOEA/D		0-0-4	0-0-4
IBEA			0-0-4

In order to provide more details, the performance metric values are shown in Tables III (GD), IV (IGD), and V (HV). The best and second best results are highlighted in gray and light gray. As the table demonstrates, SRA always ranks 1st or 2nd place in terms of both convergence (GD, IGD, HV) and diversity (IGD, HV) on most test cases of the Movielens data set.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we model a many-objective search-based recommender system and adopt a recently proposed many-objective evolutionary algorithm to optimize it. Experimental results on the Movielens data set demonstrate that our algorithm performs better in terms of GD, IGD and HV on most

[1] $\mathbf{z}^{nadir}$ is the nadir point which consists of the worst values for all objective dimensions in the Pareto front.

TABLE III
GD. MEAN (IN LARGE FONT SIZE) AND STANDARD DEVIATION (IN SMALL FONT SIZE)

	SRA	NSGA-III	MOEA/D	IBEA
SBRS1p5d	$3.08e-03_{8.4e-04}$	$1.45e-02_{1.7e-03}$	$4.86e-03_{4.8e-04}$	$2.23e-03_{5.9e-04}$
SBRS2p5d	$2.84e-03_{7.8e-04}$	$1.30e-02_{1.2e-03}$	$3.75e-03_{1.6e-03}$	$2.50e-03_{4.8e-04}$
SBRS3p5d	$2.63e-03_{7.3e-04}$	$1.53e-02_{1.2e-03}$	$4.26e-03_{5.3e-04}$	$2.07e-03_{4.7e-04}$
SBRS4p5d	$2.58e-03_{8.9e-04}$	$1.38e-02_{1.5e-03}$	$4.03e-03_{1.0e-03}$	$2.17e-03_{5.1e-04}$

TABLE IV
IGD. MEAN (IN LARGE FONT SIZE) AND STANDARD DEVIATION (IN SMALL FONT SIZE)

	SRA	NSGA-III	MOEA/D	IBEA
SBRS1p5d	$4.41e-02_{4.8e-03}$	$6.65e-01_{7.1e-02}$	$2.24e-01_{1.1e-02}$	$1.35e+00_{1.0e-02}$
SBRS2p5d	$2.97e-02_{2.7e-03}$	$7.02e-01_{5.5e-02}$	$1.97e-01_{2.4e-02}$	$1.49e+00_{1.4e-02}$
SBRS3p5d	$3.89e-02_{4.8e-03}$	$6.53e-01_{5.7e-02}$	$1.92e-01_{1.2e-02}$	$1.37e+00_{1.2e-02}$
SBRS4p5d	$3.96e-02_{5.9e-03}$	$7.32e-01_{5.2e-02}$	$2.14e-01_{2.5e-02}$	$1.40e+00_{1.6e-02}$

TABLE V
HV. MEAN (IN LARGE FONT SIZE) AND STANDARD DEVIATION (IN SMALL FONT SIZE)

	SRA	NSGA-III	MOEA/D	IBEA
SBRS1p5d	$3.00e-01_{1.5e-02}$	$1.88e-01_{1.7e-02}$	$1.61e-02_{5.9e-03}$	$6.68e-02_{6.9e-03}$
SBRS2p5d	$3.05e-01_{1.1e-02}$	$2.23e-01_{1.2e-02}$	$9.60e-03_{1.1e-02}$	$1.06e-01_{2.5e-02}$
SBRS3p5d	$2.97e-01_{1.4e-02}$	$1.87e-01_{1.1e-02}$	$1.52e-02_{4.9e-03}$	$6.86e-02_{7.1e-03}$
SBRS4p5d	$3.35e-01_{1.6e-02}$	$2.37e-01_{1.4e-02}$	$7.80e-03_{7.0e-03}$	$8.89e-02_{1.4e-02}$

test cases. However, as a preliminary study, there are many potential future directions:

1) Testing the algorithms on larger benchmark data sets industial applications.
2) Comparing the algorithm with the single-objective algorithms from the recommender system area in terms of recommender system specific performance metrics.
3) Analyzing the parameter sensitivity of the algorithm.
4) Considering more performance metrics of recommender systems as optimization objectives in our framework may give us more thorough understanding of how good a recommendation list is.
5) Trying to optimize the problem of filling scoring matrix.
6) Investigating how to use mutli-criteria decision making methods to select a final recommendation list from the solution set.

ACKNOWLEDGMENT

This work was supported in part by the National Natural Science Foundation of China (Grant No. 61329302 and Grant No. 61573328), the Program for New Century Excellent Talents in University (Grant No. NCET-12-0512) and EPSRC (Grant No. EP/K001523/1) and the Natural Science Key Research Project for Higher Education Institutions of Anhui Province (KJ2016A438) and the Fundamental Research Funds for the Central Universities (WK2150110002). Xin Yao was also supported by a Royal Society Wolfson Research Merit Award. The authors would like to thank Prof. Maoguo Gong from Xidian University, Xi'an, China for kindly sharing the sourcecode with us.

REFERENCES

[1] Y. Koren, "Factorization meets the neighborhood: a multifaceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2008, pp. 426–434.

[2] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 17, no. 6, pp. 734–749, 2005.

[3] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *Proceedings of the 1994 ACM conference on Computer supported cooperative work*. ACM, 1994, pp. 175–186.

[4] W. Hill, L. Stead, M. Rosenstein, and G. Furnas, "Recommending and evaluating choices in a virtual community of use," in *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM Press/Addison-Wesley Publishing Co., 1995, pp. 194–201.

[5] 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.

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

[7] M. Zhang, J. Tang, X. Zhang, and X. Xue, "Addressing cold start in recommender systems: A semi-supervised co-training algorithm," in *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. ACM, 2014, pp. 73–82.

[8] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 4, no. 1, p. 1, 2010.

[9] S. Kabbur, X. Ning, and G. Karypis, "Fism: factored item similarity models for top-n recommender systems," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 659–667.

[10] 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*. ACM, 2001, pp. 285–295.

[11] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.

[12] 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.

[13] 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.

[14] A. Mnih and R. Salakhutdinov, "Probabilistic matrix factorization," in *Advances in neural information processing systems*, 2007, pp. 1257–1264.

[15] Z. Lu, Z. Dou, J. Lian, X. Xie, and Q. Yang, "Content-based collaborative filtering for news topic recommendation," in *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[16] K. Niemann and M. Wolpers, "A new collaborative filtering approach for increasing the aggregate diversity of recommender systems," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 955–963.

[17] X. Amatriain, N. Lathia, J. M. Pujol, H. Kwak, and N. Oliver, "The wisdom of the few: a collaborative filtering approach based on expert opinions from the web," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009, pp. 532–539.

[18] I. Guy and W. Geyer, "Social recommender system tutorial," in *Proceedings of the 8th ACM Conference on Recommender systems*. ACM, 2014, pp. 403–404.

[19] K. Onuma, H. Tong, and C. Faloutsos, "Tangent: a novel,'surprise me', recommendation algorithm," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2009, pp. 657–666.

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

[21] M. Ge, C. Delgado-Battenfeld, and D. Jannach, "Beyond accuracy: evaluating recommender systems by coverage and serendipity," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 257–260.

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

[23] A. Maksai, F. Garcin, and B. Faltings, "Predicting online performance of news recommender systems through richer evaluation metrics," in *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 2015, pp. 179–186.

[24] M. T. Ribeiro, N. Ziviani, E. S. D. Moura, I. Hata, A. Lacerda, and A. Veloso, "Multiobjective pareto-efficient approaches for recommender systems," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 5, no. 4, p. 53, 2014.

[25] D. Agarwal, B.-C. Chen, P. Elango, and X. Wang, "Click shaping to optimize multiple objectives," in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011, pp. 132–140.

[26] M. Rodriguez, C. Posse, and E. Zhang, "Multiple objective optimization in recommender systems," in *Proceedings of the sixth ACM conference on Recommender systems*. ACM, 2012, pp. 11–18.

[27] B. Li, K. Tang, J. Li, and X. Yao, "Stochastic ranking algorithm for many-objective optimization based on multiple indicators," *accepted by Evolutionary Computation, IEEE Transactions on*, 2016.

[28] K. Miettinen, *Nonlinear Multiobjective Optimization*. Springer, 1999, vol. 12.

[29] P. Yu, "Cone convexity, cone extreme points, and nondominated solutions in decision problems with multiobjectives," *Journal of Optimization Theory and Applications*, vol. 14, no. 3, pp. 319–377, 1974.

[30] 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*. ACM, 2010, pp. 39–46.

[31] T. Murakami, K. Mori, and R. Orihara, "Metrics for evaluating the serendipity of recommendation lists," in *New frontiers in artificial intelligence*. Springer, 2008, pp. 40–46.

[32] F. Ricci, L. Rokach, and B. Shapira, *Introduction to recommender systems handbook*. Springer, 2011.

[33] T. P. Runarsson and X. Yao, "Stochastic ranking for constrained evolutionary optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 4, no. 3, pp. 284–294, 2000.

[34] Q. Zhang and H. Li, "MOEA/D: A multiobjective evolutionary algorithm based on decomposition," *Evolutionary Computation, IEEE Transactions on*, vol. 11, no. 6, pp. 712–731, 2007.

[35] J. J. Durillo and A. J. Nebro, "jmetal: A java framework for multi-objective optimization," *Advances in Engineering Software*, vol. 42, no. 10, pp. 760–771, 2011.

[36] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 5, no. 4, p. 19, 2015.

[37] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: Solving problems with box constraints," *Evolutionary Computation, IEEE Transactions on*, vol. 18, no. 4, pp. 577–601, Aug 2014.

[38] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature-PPSN VIII*. Springer, 2004, pp. 832–842.

[39] H. Wang, L. Jiao, and X. Yao, "Two_arch2: An improved two-archive algorithm for many-objective optimization," *Evolutionary Computation, IEEE Transactions on*, vol. 19, no. 4, pp. 524–541, Aug 2015.

[40] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, pp. 80–83, 1945.

[41] C. A. C. Coello and M. R. Sierra, "A study of the parallelization of a coevolutionary multi-objective evolutionary algorithm," in *MICAI*. Springer, 2004, pp. 688–697.

[42] H. Ishibuchi, H. Masuda, and Y. Nojima, "A study on performance evaluation ability of a modified inverted generational distance indicator," in *Proceedings of the 2015 on Genetic and Evolutionary Computation Conference*. ACM, 2015, pp. 695–702.

[43] D. Van Veldhuizen and G. Lamont, "Evolutionary computation and convergence to a pareto front," in *Genetic Programming Conference*, 1998, pp. 221–228.

[44] E. Zitzler and L. Thiele, "Multiobjective optimization using evolutionary algorithmsa comparative case study," in *Parallel Problem Solving from Nature (PPSN V)*. Springer, 1998, pp. 292–301.

[45] M. Emmerich, N. Beume, and B. Naujoks, "An emo algorithm using the hypervolume measure as selection criterion," in *Evolutionary Multi-Criterion Optimization*. Springer, 2005, pp. 62–76.