Multi-Objective Evolutionary Programming for Developing Recommender Systems based on Collaborative Filtering

Edward Hinojosa-Cardenas[1]
Universidad Nacional de San Agustin
de Arequipa, Peru

Edgar Sarmiento-Calisaya[2]
Universidad Nacional de San Agustin
de Arequipa, Peru

Cesar A. Martinez-Salinas[3]
Universidad Nacional de San Agustin
de Arequipa, Peru

Lehi Quincho-Mamani[4]
Universidad Nacional de San Agustin
de Arequipa, Peru

Jair F. Huaman-Canqui[5]
Universidad Nacional de San Agustin
de Arequipa, Peru

***Abstract*—In the era of internet, several online platforms offer many items to users. Users could spend a lot of time to find (or not) some items they are interested, sometimes, they will probably not find the desired items. An effective strategy to overcome this problem is a recommender system, one of the most popular applications of machine learning. Recommender systems select most appropriate items to an specific user based on previous information between items and users, and they are developed using diffeent approaches. One of the most successful approach for developing recommender systems is collaborative filtering, which can filter out items that a user might like based on reactions of users with similar profiles. Often, traditional recommender systems only consider *precision* as evaluation metric of performance, however, others metrics (like recall, diversity, novelty, etc) are also important. Unfortunately, some metrics are conflicting, e.g., precision impacts negatively on other metrics. This paper presents a multi-objective evolutionary programming method for developing a recommender system, which is based on a new collaborative filtering technique, while maximizes the *recall* for a given *precision*, The new collaborative filtering technique uses three components for recommending an item to a user: 1) clustering of users; 2) a previous memory-based prediction; and 3) five decimal parameters (threshold average clustering, threshold penalty, threshold incentive, weight attached to average clustering and weight attached to *Pearson correlation*). The multi-objective evolutionary programming optimizes the clustering of users and the five decimal parameters, while, it searches maximizes both similarity precision and recall objectives. A comparison between the proposed method and a previous non-evolutionary method shows that the proposed method improves precision and recall metric on a benchmark database.**

***Keywords*—Collaborative filtering; clustering; evolutionary programming; multi-objective; recommender systems**

I. Introduction

Huge amounts of user data are generated and collected every day on the web, given the explosive growth of information, users are often greeted with more than countless choices [1]. Recommender system is an effective tool for helping the user in cutting the time needs to find personalised movies, products, documents, friends, places, services, among others [2]. Also, a recommender system is one of the most important and new research area in machine learning [3].

The most commonly recommendation approaches [4] used to produce a list of items for a user are: content-based, collaborative filtering and hybrid approaches. Content-based filtering is based on the item to define the prediction, i.e., it uses features of the item to make a similar item recommendations. Collaborative filtering is one of the most prominent and popular approaches, It recommends similar items to similar users (similar users is based on past behavior, previous purchases, preferences, ratings of other products, average purchase amount, etc.). And, hybrid filtering combines the two filtering approaches.

Evolutionary computing (or evolutionary algorithm) is a research area within computer science, as the name suggests, it is a special flavour of computing, which draws inspiration from the process of natural evolution. The fundamental metaphor of evolutionary computing relates this powerful natural evolution to a particular style of problem solving - that of trial-and-error [5]. The main branches [6] of evolutionary computing are: Genetic Algorithm, Evolutionary Strategies, Differential Evolution, Genetic Programming and Evolutionary Programming. Evolutionary Programming (EP) is a computational optimization method to find global optimal solution for a given problem and, it is used in this paper. All of these branches have been used to resolve different trends in recommender systems research [7], among these trends, the use of evolutionary computing for optimizing weights of recommendation techniques/component and learning recommendation systems.

The performance of classical recommendation algorithms is usually evaluated by accuracy related metrics [8], like *precision* and *recall*. From the definitions, precision and recall are conflicting objectives because while recall tries to increase the number of tagged entries as much is possible or the fraction of relevant information that are retrieved, precision tries to increase the number of correctly tagged entries or the fraction of information retrieved that are relevant [9]. A common strategy for dealing with this problem is the evolutionary multi-objective optimization, which refers to the use of evolutionary algorithms to solve problems with two or more (often conflicting) objective functions [10].

The proposed method, named MOEP-CF, is based on a Multi-Objective Evolutionary Programming for developing recommender systems, which uses a new Collaborative Filtering technique and, improves precision and recall objectives. The new collaborative filtering technique is based on three components to recommend an item to a user: 1) clustering of the users; 2) a previous memory-based prediction; and 3) five decimal parameters (threshold average clustering, threshold penalty, threshold incentive, weight attached to average clustering and weight attached to *Pearson correlation*). The optimization process is guided by a proposed new mutation operator with ten types of mutation and improves six components or sub components of the new collaborative filtering technique: 1) users assigned to each cluster; 2) threshold for penalty; 3) threshold for incentive; 4) threshold for average cluster ranking; 5) weight attached to average cluster ranking; and 6) weight attached to memory-based prediction. The MOEP-CF proposed method is based on a previous non multi-objective evolutionary method proposed in [11].

The main contributions of this work are summarized as follows:

- The new collaborative filtering technique based on three components.
- The use a multi-objective Evolutionary programming for developing recommender systems based on new collaborative filtering technique and improving precision and recall objectives.
- The new mutation operator with ten types of mutation to improve components or sub components of the new collaborative filtering technique.

The rest of the paper is structured as follows. Related work is presented in Section II. Section III defines collaborative filtering and gives an example of the clustering-based collaborative filtering algorithm used in this paper for recommendation process. Section IV presents a review of evolutionary programming and the multi-objective evolutionary programming algorithm used in the proposed method is presented. A detailed description of the proposed method is presented in Section V. Section VI shows experimental results and compares that results with the previous method, and presents advantages of the proposed method. Section VII concludes the work and presents future work.

II. Related Work

The proposed method in this paper is related to four broader research areas, namely CF approaches in recommender systems, clustering-based recommender systems, evolutionary computing in recommender systems and recommender systems that analyzed the performance metrics such as precision and recall.

A. CF-Based Recommendation Approaches

Collaborative filtering, as one of the most successful recommendation techniques, has been widely studied and applied by various research institutions and industries. However, CF-based approach often suffers from several shortcomings [16] [17], such as data sparsity, cold start, and scalability issues, which seriously affect the efficiency of a recommender system (RS). To overcome the mentioned problems, many data mining and machine learning techniques such as clustering [19] [22] [20] [21] and matrix factorization (Surveyed in [18]) are proposed to improve the performance of RS. Matrix factorization, one of the unsupervised learning methods, can play a role in reducing dimensionality and eventually alleviating the data sparsity [16].

Yang et al. [16] and Chen et al. [17] review and summarize the traditional CF-based approaches and techniques used in RS, classify and compare several typical CF algorithms as memory-based approaches and model-based approaches, and study some recent hybrid CF-based recommendation approaches and techniques, including the latest hybrid memory-based and model-based CF recommendation algorithms.

B. Clustering-based Recommender Systems

There has been diverse research to enhance recommendation accuracy by means of clustering methods, such as [19] [22] [20] [21]. In [19], is presented an approach (D2P) to addresses the tradeoff between privacy and quality of recommendation, it can be applied to any collaborative recommender system. The main intuition behind D2P is to rely on a distance metric between items so that groups of similar items can be identified. As a result of grouping, the K most similar users based on the similarity measure were selected for recommendation. In [22], a k-means clustering-based recommendation algorithm is proposed, which addresses the scalability issues associated with traditional recommender systems. An issue with traditional k-means clustering algorithms is that they choose the initial k centroid randomly, which leads to inaccurate recommendations and increased cost for offline training of clusters. In [20], a fuzzy C-Means approach has been proposed for user-based collaborative filtering and its performance against different clustering approaches (including K-Means, self-organizing maps, and fuzzy C-Means) has been assessed. A collaborative filtering algorithm based on singular value decomposition (SVD) and fuzzy clustering was shown in [21]. It also reduces the dimensionality (scalability problem) and the search range of the neighbors.

C. Evolutionary Computing in Recommender Systems

Evolutionary Computing (EC) can optimize and improve RS in the various applications. Horváth & de Carvalho [7] and Sadeghi & Asghari [23] provide comprehensive reviews of more relevant publications focusing on three relevant aspects: approaches in which EC are used to optimize weights of recommendation techniques or different component, approaches utilizing EC for clustering of items or users, and hybrid and other approaches. In [24] [25] [27] [26] [28], novel heterogeneous evolutionary clustering algorithms are presented.

In [24], a new genetic algorithm encoding is proposed as an alternative of k-means clustering. The initialization issue in the classical k-means is targeted by proposing a new formulation of the problem, to reduce the search space complexity affect as well as improving clustering quality.

In [25] [27] [26] [28], novel heterogeneous evolutionary clustering algorithms are presented. The goal of these algorithms is to gather users with similar interest into the same cluster and to help users find items that fit their personal tastes best. Firstly, items and users are regarded as heterogeneous individuals in the network. According to the constructed network clustering model, states of users evolve over time. States of users would be stable after some period of iteration. In light of stable states of users, they are clustered into several groups. Liji et al. [27] compute the user attribute distances. In chen et al. [26], a dynamic evolutionary clustering algorithm based on time weight and latent attributes is proposed. Then, collaborative filtering is applied in each cluster to predict the ratings.

In [28], a novel collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering is presented. Firstly, score matrix is pre-processed with normalization and dimension reduction. Based on these processed data, clustering principle is generated and dynamic evolutionary clustering is implemented. Secondly, the search for the nearest neighbors with highest similar interest is considered. A measurement about the relationship between users is proposed, called user correlation, which applies the satisfaction of users and the potential information.

In [29], a hybrid approach to increase the accuracy of recommendation of user-based collaborative filtering video recommender system is proposed. The proposed approach combines k-means clustering algorithm and two different evolutionary algorithms which are Accelerated Particle Swarm Optimization Algorithm (APSO) and Forest Optimization Algorithm (FOA).

D. Performance Analysis in terms of Accuracy (Precision and Recall)

Many evaluation metrics are available for recommendation systems and each has its own pros and cons, but a few works provide guidance on how to choose and evaluate the impacts among them. For example, Schröder et al. [30] describe accuracy related evaluation metrics and discuss their applicability for different types of recommender systems.

Classical recommender systems mainly focus on the accuracy related metrics. 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. Lin et al. [31] present a multi-objective personalized recommendation algorithm using extreme point guided evolutionary computation (called MOEA-EPG). In MOEA-EPG, the accuracy, diversity, and novelty of recommendations are chosen as the three conflicting objectives, and the aim of that algorithm is to optimize the modeled MOP for personalized recommendation.

In the context of trade-offs among accuracy related metrics, specifically precision and recall, Karabadji et al. [32] and Tran et al. [11] present methods to improve recommendation performance in terms of accuracy related metrics. Karabadji et al. [32] propose an evolutionary multi-objective optimization-based recommendation system to pull up a group of profiles that maximizes both similarity with the active user and diversity between its members. The recommendation system will provide high performances in terms of both accuracy (precision, recall, f-measure) and diversity. Tran et al. [11] propose a new clustering-based CF (CBCF) method using an incentivized/penalized user (IPU) model only with ratings given by users. The purpose of CBCF with the IPU model is to improve recommendation performance such as precision, recall, and F1 score by carefully exploiting different preferences among users, i.e, maximize the recall (or equivalently F1 score) for a given precision. In addition, performance on the precision and recall of other various recommender systems was analyzed in [33] [34] [35].

III. Collaborative Filtering

As one of the most successful approaches to building recommender systems, Collaborative Filtering (CF) uses the known preferences of a group of users to make recommendations or predictions of the unknown preferences for other users [12]. Three main categories of CF techniques are: memory-based, model-based, and hybrid (that combine the two before techniques).

Memory-based techniques use the entire or a sample of the user-item database to generate a prediction. Model-based techniques design and develop models that analyzes the training data to recognize complex patterns and then make intelligence predictions for test data. The proposed MOEP-CF method uses both a memory-based and a model-based technique.

The memory-based technique used by the proposed method is the Weighted Sum of Others' Ratings [13]. Predict rating of a particularity item i to a user a is calculated by the following equation:

$$\hat{r}_{a,i} = \overline{r}_a + \frac{\sum_{u \in U} \left(r_{u,i} - \overline{r}_u\right) \times w_{a,u}}{\sum_{u \in U} |w_{a,u}|} \tag{1}$$

where:

- $\hat{r}_{a,i}$: Predict rating for the user a on the item i
- $\overline{r}_a$: Average rating for the user a on all other rated items
- $\overline{r}_u$: Average rating for the user u on all other rated items
- $\sum_{u \in U}$: The summations over all the users $u \in U$ who have rated the item i
- $w_{a,u}$: Weight or *Pearson correlation* between the user a and the user u.

The *Pearson correlation* between the user a and user u is calculated by the following equation:

$$w_{a,u} = \frac{\sum_{i \in I} \left(r_{a,i} - \overline{r}_a\right)\left(r_{u,i} - \overline{r}_u\right)}{\sqrt{\sum_{i \in I} \left(r_{a,i} - \overline{r}_a\right)^2}\sqrt{\sum_{i \in I} \left(r_{u,i} - \overline{r}_u\right)^2}} \tag{2}$$

where:

- $w_{a,u}$: Weight or *Pearson correlation* between the user a and the user u
- $\overline{r}_a$: Average rating of the co-rated items of the user a
- $\overline{r}_u$: Average rating of the co-rated items of the user u

- $\sum_{i \in I}$: The summations over all the items that both the users u and the user a have rated
- $w_{a,u}$: Weight or *Pearson correlation* between the user a and the user u.

For illustration, Table I shows a rating matrix for 8 users and 5 items.

TABLE I. A SIMPLE EXAMPLE OF RATING MATRIX

	i1	**i2**	**i3**	**i4**	**i5**
u1	4	4	1		5
u2	2	5		3	1
u3	3	1	5		1
u4	5	2	2	3	1
u5	1	4	3	5	5
u6	1		2		2
u7	3	4	2	4	3
u8	3	5		1	5

Based on information available in Table I, it is possible to calculate the *Pearson correlation* between users and, predict rating between users and items. For instance, $w_{1,5} = 0.28$ and $w_{3,7} = -0.82$; and, $\hat{r}_{1,4} = 3.18$ and $\hat{r}_{8,3} = 2.72$. Tables II and III show the *Pearson correlation* between users and predicted ratings for missed ratings, respectively.

TABLE II. *Pearson Correlation* OR WEIGHTS BETWEEN USERS

	u1	**u2**	**u3**	**u4**	**u5**	**u6**	**u7**	**u8**
u1	0.00	-0.69	-0.83	0.00	0.28	-0.28	0.71	0.50
u2	-0.69	0.00	-0.28	-0.03	0.08	-1.00	0.85	0.05
u3	-0.83	-0.28	0.00	0.45	-0.68	-0.19	-0.82	-1.00
u4	0.00	-0.03	0.45	0.00	-0.77	-0.97	0.08	-0.56
u5	0.28	0.08	-0.68	-0.77	0.00	0.87	0.43	0.05
u6	-0.28	-1.00	-0.19	-0.97	0.87	0.00	-0.50	1.00
u7	0.71	0.85	-0.82	0.08	0.43	-0.50	0.00	-0.30
u8	0.50	0.05	-1.00	-0.56	0.05	1.00	-0.30	0.00

TABLE III. PREDICTED RATINGS

	i1	**i2**	**i3**	**i4**	**i5**
u1				3.18	
u2			2.70		
u3				2.70	
u4					
u5					
u6		1.62		0.94	
u7					
u8			2.72		

IV. MULTI-OBJECTIVE EVOLUTIONARY PROGRAMMING

Evolutionary Programming (EP) was proposed by Lawrence Fogel [14], it is based on natural evolution and it has been applied with success to many numerical and combinatorial optimization problems.

EP does not require the use of a specific form of representation (for example, real-valued or integer strings), allowing the user to select the most suitable representation for the problem at hand. That is an important feature used in this paper.

Whatever the choice of representation, EP uses an iterative improvement process whereby a parent population structures are perturbed using a suitably defined mutation operator, with a selection process taking place to see which structures survive into the next iteration of the Algorithm [6]. An overview of the canonical EP algorithm is provided in Algorithm 1.

Algorithm 1: Evolutionary Programming Algorithm

```
RepresentationIndividuals();
$t = 0$;
Initialize($P_p(t)$);
Evaluate($P_p(t)$);
while isNotTerminated() do
    $P_o(t)$ = Mutation($P_p(t)$);
    Evaluate($P_o(t)$);
    $P_p(t+1)$ = Select($P_p(t) \cup P_o(t)$);
    $t = t + 1$;
end
```

Multi-Objective Evolutionary Programming (MOEP) is an extended version of single objective EP. The MOEP algorithm with non-domination sorting can be described in Algorithm 2 and, it was proposed in [15]. The MOEP-CF method is based on MOEP.

Algorithm 2: Multi−Objective Evolutionary Programming

```
RepresentationIndividuals();
$t = 0$;
Initialize($P_p(t)$);
Evaluate($P_p(t)$);
while isNotTerminated() do
    $P_o(t)$ = Mutation($P_p(t)$);
    Evaluate($P_o(t)$);
    $P_m(t)$ = $P_p(t) \cup P_o(t)$;
    IdentifyNonDominated($P_m(t)$);
    AssingFrontNumber($P_m(t)$);
    SortByFrontNumber($P_m(t)$);
    $P_p(t+1)$ = Select($P_m(t)$);
    $t = t + 1$;
end
```

V. PROPOSED METHOD

The MOEP-CF proposed method is based on a method proposed in [11], and hybrids Collaborative Filtering (see Section III) and Multi-Objective Evolutionary Programming (see section IV). The MOEP-CF recommends or does not recommend a certain item i to a certain user a using a model-based technique based on three features: 1) Previous predicted ranking, 2) Clustering of the users, and 3) Five following parameters:

- γ: threshold average clustering of item i in cluster c
- α: threshold penalty
- β: threshold incentive
- ρ: weight attached to average clustering
- σ: weight attached to *Pearson correlation*

To decide whether to recommend or not to recommend a certain item i to a certain user a, the proposed method follows the Algorithm 3.

Algorithm 3: Algorithm to Recommend or not Recommend the Item i to the User a (in the Cluster c)

```
if $\overline{C}_c^i \times \rho \geq \gamma$ then
    if $\hat{r}_{a,i} \times \sigma \geq \beta$ then
        Recommend item $i$ to user $a$;
    else
        Don’t recommend item $i$ to user $a$;
    end
else
    if $\hat{r}_{a,i} \times \sigma \geq \alpha$ then
        Recommend item $i$ to user $a$;
    else
        Don’t recommend item $i$ to user $a$;
    end
end
```

For illustration, Table IV shows the following features: three clusters and five parameters.

TABLE IV. Examples of Features for Proposed Collaborative Filtering Technique

Features	Value
Cluster 1	$C_1 = \{u_1, u_4, u_7\}$
Cluster 2	$C_2 = \{u_2, u_6\}$
Cluster 3	$C_3 = \{u_3, u_5, u_8\}$
γ	2.21
α	0.51
β	0.99
ρ	0.94
σ	0.35

Following the Algorithm 1 and the information provided in Table IV, it is possible to recommend the item 4 to the user 1; and don’t recommend the item 3 to the user 8. A detail of that recommendations are explained in Algorithm 4 and 5, respectively.

Algorithm 4: Recommending the item 4 to the user 1 - Explanation

```
if $((3 + 4) \div 2 = 3.5) \times 0.94 \geq 2.21$ then
    if $3.18 \times 0.35 \geq 0.99$ then
        **Recommend item 4 to user 1**;
    else
        Don’t recommend item 4 to user 1;
    end
else
    if $3.18 \times 0.34723 \geq 0.51$ then
        Recommend item 4 to user 1;
    else
        Don’t recommend item 4 to user 1;
    end
end
```

Algorithm 5: Recommending (not) the item 3 to the user 8 - Explanation

```
if $((5 + 3) \div 2 = 4.0) \times 0.94 \geq 2.21$ then
    if $2.72 \times 0.35 \geq 0.99$ then
        Recommend item 3 to user 8;
    else
        **Don’t recommend item 3 to user 8**;
    end
else
    if $2.72 \times 0.35 \geq 0.51$ then
        Recommend item 3 to user 8;
    else
        Don’t recommend item 3 to user 8;
    end
end
```

In order to define features shown in the Table IV for maximizing the precision and recall metrics, the MOEP-CF proposed method follows the Algorithm 5. It is detailed in next subsections step by step.

A. Representation of Individuals

Each individual or solutions is encoded in a vector with two parts: the clusters part and the parameters part, such as depicted in Fig. 1.

The clusters part is an integer vector with $|U|$ dimensions (U is the set of users); the dimension 1 contains the index of the cluster where user 1 belongs (iC_1), the dimension 2 contains the index of the cluster where user 2 belongs (iC_2), and so on.

The parameters part is a decimal vector with five dimensions; each dimension represents the parameters γ, α, β, ρ and σ, respectively.

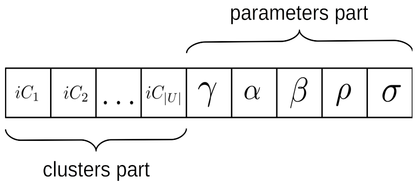

Fig. 1. Representation of Individuals.

Fig. **??** depicts a representation of features detailed in Table IV.

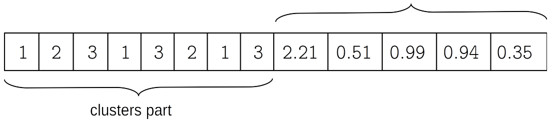

Fig. 2. Example of a Representation.

B. Initialize Parents Population

Initial population or parent population ($P_p(t)$) of size $sizeP$ are randomly generated. For the clusters part, each dimensions contains a random integer between 1 and iC_{max} (iC_{max} represents the maximum number of clusters). For the parameters part, γ, α and β contains a random decimal between $minRank$ and $maxRank$; ρ and σ contains a random decimal between 0 and 1.

Values $sizeP$, iC_{max}, $minRank$ and $maxRank$ are a parameters of the proposed method.

C. Evaluate Parent Population

The MOEP-CF proposed method uses training and test datasets. The test dataset contains some registers with ratings from users to items that are not in training dataset. For each individual in $P_p(t)$, it is calculated the precision and recall.

Based on training dataset, the Weighted Sum of Others' Ratings (see Section III) is used to define predicted ratings for the ratings in test dataset. Then, for each individual in $P_p(t)$, the Algorithm 1 is used to recommend or don't recommend an item to a user on test dataset. Finally, precision and recall are calculated based on the following equations:

$$precision = \frac{tp}{tp + fp} \tag{3}$$

$$precision = \frac{tp}{tp + fn} \tag{4}$$

$$tp = \sum_{v=0}^{T} \left[R^{test}_{v,u,i} \ and \ r^{test}_{v,u,i} \geq \delta_{ref}\right] \tag{5}$$

$$fp = \sum_{v=0}^{T} \left[NR^{test}_{v,u,i} \ and \ r^{test}_{v,u,i} < \delta_{ref}\right] \tag{6}$$

$$fn = \sum_{v=0}^{T} \left[NR^{test}_{v,u,i} \ and \ r^{test}_{v,u,i} \geq \delta_{ref}\right] \tag{7}$$

where:

- tp: true positive.
- fp: false positive.
- fn: false negative.
- T: number of registers in test dataset.
- $R^{test}_{v,u,i}$: Recommend the item i to user the u (in register v) in test dataset following the Algorithm 1.
- $NR^{test}_{v,u,i}$: Don't recommend the item i to user the u (in register v) in test dataset following the Algorithm 1.
- $r^{test}_{v,u,i}$: Rating of the user u on the item i (in register v) in test dataset.
- δ_{ref}: Threshold value for determining whether a user is really satisfied with the recommended item. It is a parameter of the proposed method and generally set to 4.0 (or 8.0) in case of a five-point scale (or a ten-point scale).
- $[pred]$: Evaluate to 1 if the predicate $pred$ is true and 0 otherwise.

D. Terminating Condition

The terminating condition of MOEP is the maximum number of iterations $maxIter$.

E. Mutation

The mutation process is applied on each individual in $P_p(t)$ to generate offspring population ($P_o(t)$) and, it is one of the main contributions of the MOEP-CF proposed method. Each parent solution suffer a type of mutation (with a probability) to generate a offspring individual. Table V shows the ten probabilities and types of mutation using in the proposed method.

TABLE V. TYPES OF MUTATION

Probability	Type of Mutation
10%	Changing randomly the cluster of the 10% of user, i.e., changing randomly 10% of dimensions in clusters part
10%	Changing randomly the cluster of the 20% of user, i.e., changing randomly 20% of dimensions in clusters part
10%	Changing randomly the cluster of the 30% of user, i.e., changing randomly 30% of dimensions in clusters part
10%	Changing randomly the cluster of the 40% of user, i.e., changing randomly 40% of dimensions in clusters part
10%	Changing randomly the cluster of the 50% of user, i.e., changing randomly 50% of dimensions in clusters part
10%	Changing the γ parameter to a random value between $minRank$ and $maxRank$ values.
10%	Changing the α parameter to a random value between $minRank$ and $maxRank$ values.
10%	Changing the β parameter to a random value between $minRank$ and $maxRank$ values.
10%	Changing the ρ parameter to a random value between 0 and 1.
10%	Changing the σ parameter to a random value between 0 and 1.

F. Evaluate Offspring Population

The process for calculating precision and recall of each individual in $P_o(t)$ follows the same evaluation process explained in subsection V-C for $P_p(t)$. After that, the $P_o(t)$ is merged to $P_p(t)$ to generate a merged population ($P_m(t)$).

G. Identify Non-Dominated Solutions

A non-dominated solution is a solution that is not dominated by any other solution in $P_m(t)$. In the MOEP-CF proposed method, there is no other solution that performs better in precision and recall objective (or equal to one objective and better than another) than the non-dominated solution.

H. Assign and Sort by Front Number

The no-dominated solutions in $P_m(t)$, identified in the previous step, are inserted in the Front 0. A new search for non-dominated solutions is carried out in the population $P_m(t)$ without considering the solutions in the Front 0, the new non-dominated solutions found are inserted in the Front 1, and so on.

I. Select the New Parent Population

The next parent population ($P_p(t+1)$) is filled according to front ranking until the size of $P_p(t+1)$ is equal to $sizeP$. If one front is taking partially, perform, for each solution, the sum of Euclidean distance between the objectives of that solution and other solutions in order to diversify the solutions for next iterations. The greater sum of Euclidean distance are preferred.

Finally, the evolution process is repeated until a termination condition has been reached. The source code of the proposed MOEP-CF proposed method is available on github (username: Edward-Hinojosa-Cardenas, project: MOEP-CF).

VI. Experimental Results and Performance Comparison

In this section, the performance of the MOEP-CF proposed method is evaluated and compared with a similar previous method (non-evolutionary multi-objective method, named CBCF [11]). To show that MOEP-CF method performs better, it is applied on the MovieLens 100K dataset (collected by the GroupLens Research Project at the University of Minnesota). Table VI describes the dataset used in the evaluation. The proposed method is evaluated by 5 folds cross-validation.

TABLE VI. Description of MovieLens 100K dataset

Feature	Value
Ratings or registers	100 000 anonymous ratings (each user has at least 20 ratings)
Scale	5-star scale
Users	943
Movies	1682

The parameters used in the MOEP-CF proposed method are outlined in Table VII.

TABLE VII. Parameters of the MOEP-CF Proposed Method

Parameter	Value
$sizeP$	50
C_{max}	10
$minRank$	1.0
$maxRank$	5.0
$maxRank$	5.0
δ_{ref}	4

Fig. **??**, **??**, **??**, **??** and **??** show the results (non-dominated solutions in the last iteration for precision and recall objectives) of the MOEP-CF proposed method for each fold on the MovieLens 100K dataset.

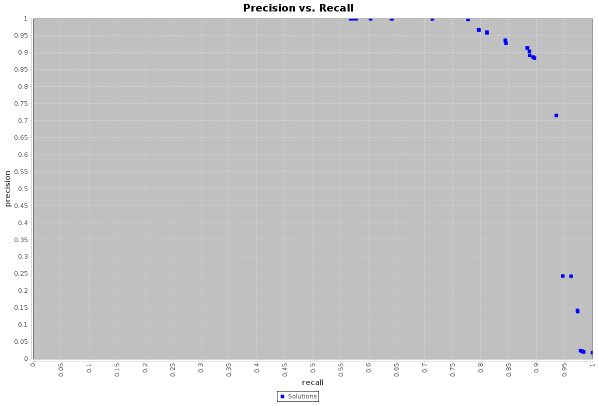

Fig. 3. Results for Two Objectives: Precision and Recall, on Fold 1 of the MovieLens 100K Dataset.

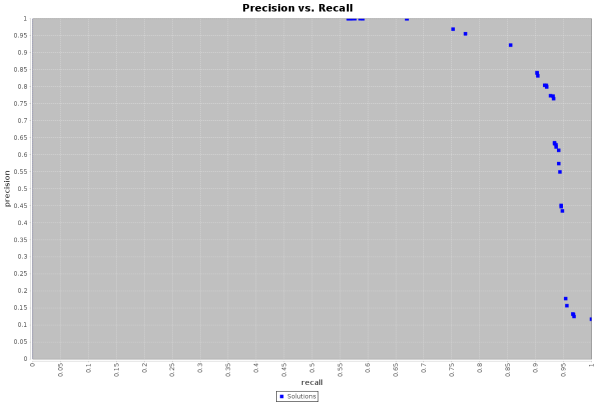

Fig. 4. Results for Two Objectives: Precision and Recall, on Fold 2 of the MovieLens 100K Dataset.

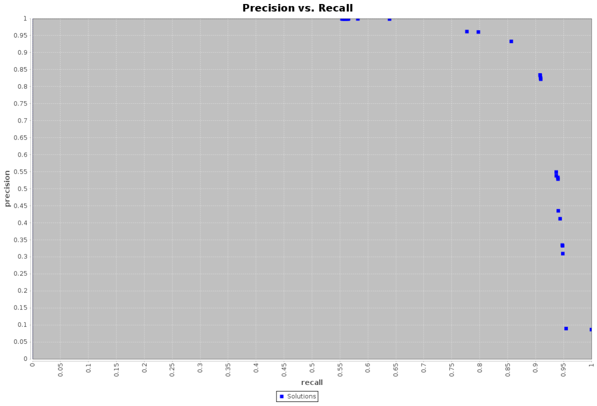

Fig. 5. Results for Two Objectives: Precision and Recall, on Fold 3 of the MovieLens 100K Dataset.

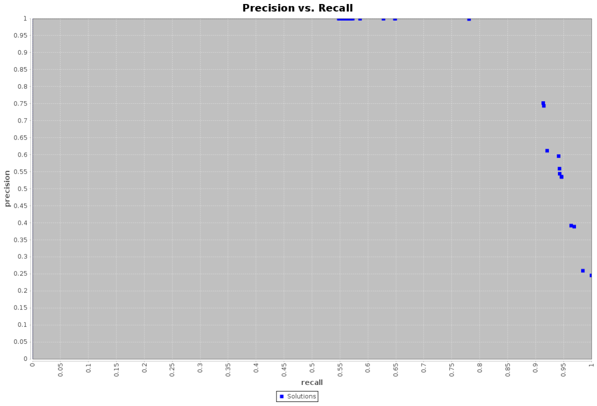

Fig. 6. Results for Two Objectives: Precision and Recall, on Fold 4 of the MovieLens 100K Dataset.

TABLE VIII. RESULTS OF THE MOEP-CF PROPOSED METHOD.

Fold	Gamma	alpha	beta	rho	sigma	Precision	Recall	F1
Fold 1	0.5150	1.0714	1.1484	0.9375	0.2659	0.8838	0.9134	0.8984
Fold 2	0.0576	4.0897	1.9213	0.6292	0.4712	0.8553	0.9220	0.8874
Fold 3	0.2445	4.1708	1.8449	0.7097	0.4500	0.8566	0.9328	0.8930
Fold 4	0.0576	4.0897	1.9213	0.6292	0.4712	0.8553	0.9220	0.8874
Fold 5	0.4838	1.1868	0.9666	0.7030	0.2367	0.8654	0.9458	0.9038
Average	**0.2717**	**2.9217**	**1.5605**	**0.7217**	**0.3790**	**0.8633**	**0.9272**	**0.8940**

TABLE IX. COMPARING THE MOEP-CF METHOD WITH THE CBCF METHOD.

Method	Clustering	Gamma	alpha	beta	rho	sigma	Precision	Recall	F1
CBCF	10 clusters (Fuzzy C-Means)	3.3	3.7	2.9	-	-	0.6448	0.8730	0.7418
MOEP-CF	10 clusters	0.2717	2.9217	1.5605	0.7217	0.3790	**0.8633**	**0.9272**	**0.8940**

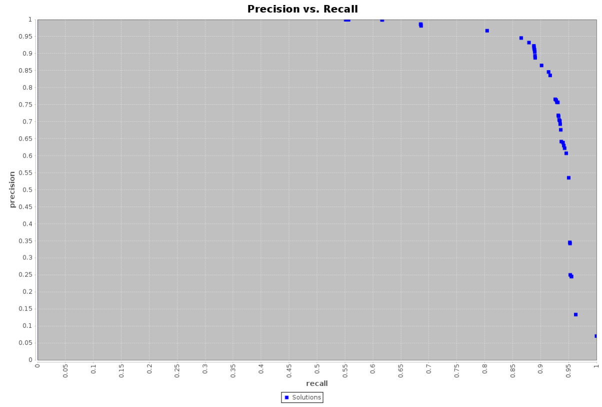

Fig. 7. Results for Two Objectives: Precision and Recall, on Fold 5 of the MovieLens 100K Dataset.

In order to assess the performance of the proposed method against the CBCF method [11]), we used the same dataset. Before of comparison, for each fold, the MOEP-CF method selects a random solution from non-dominated solutions in the last iteration. The average values for all selected solutions are shown in the Table VIII.

Table IX shows the clustering γ, α and β values and, the precision, recall and $F1$ values obtained using the CBCF method. Next, it shows the number of cluster; the average of α, β, γ, ρ and σ optimized parameters; and, the average values of precision, recall and $F1$ obtained in for that MOEP-CF proposed method.

The results show that the MOEP-CF method achieved better results than previous CBCF method on the three metrics mentioned above. The MOEP-CF method with a flexible clustering and two additional parameters (in comparison with the previous method), improves in 33.89%, 6.21% and 20.52% the precision, recall and $F1$ metrics, respectively.

However, some *limitations* should be noted. First, the proposed method uses static probability values for each mutation which can influence the final result. Second, the proposed method doesn't consider important new metrics like novelty, diversity, stability and reliability in recommendation systems. Third, the parameters of the proposed method, showed in Table VII are defined empirically or experimentally.

VII. CONCLUSION

Optimizing different objectives simultaneously is a well recognized problem in a recommender system setting. In this work, a multi-objective evolutionary programming method for developing recommender systems is proposed. It is based on a new collaborative filtering technique that achieves high precision and recall.

The main *contributions* of the proposed method are three: a new collaborative filtering technique; a multi-objective evolutionary programming for developing recommender systems improving precision and recall objectives, and, a new mutation operator with ten types of mutations.

Future Work includes: adding non-accuracy metrics like novelty, diversity, stability and reliability to multi-objective evolutionary optimization process; using other predicted ranking methods before multi-objective evolutionary optimization process; and, evaluating the proposed method taking as input other popular datasets.

ACKNOWLEDGMENT

This work was supported by the Universidad Nacional de San Agustin de Arequipa under Project IBAIB-06-2019-UNSA.

REFERENCES

[1] Zhang, Shuai and Yao, Lina and Sun, Aixin and Tay, Yi "Dearning Based Recommender System: A Survey and New Perspectives," *ACM Computing Surveys,* vol. 1, no. 1, Feb. 2019.

[2] Bushra Alhijawi and Yousef Kilani "The Recommender System: A Survey," *International Journal of Advanced Intelligence Paradigms,* vol. 5, no. 3, March 2020.

[3] Mohamed, Marwa *et al.,* "Recommender Systems Challenges and Solutions Survey," *2019 International Conference on Innovative Trends in Computer Engineering (ITCE'2019),* pp. 149–155, 2019.

[4] R. Forsati *et al.,* "A fuzzy co-clustering approach for hybrid recommender systems," *International Journal of Hybrid Intelligent Systems,* vol. 10, no. 2, pp. 71–81, April 2013.

[5] A.E. Eiben and J.E. Smith, "Introduction to Evolutionary Computing," in *Evolutionary Computing,* 2nd ed. Ed. Berlin, Germany: Springer-Verlag Berlin Heidelberg, 2015, pp. 13–16.

[6] A. Brabazon, M. O'Neill and S. McGarraghy, "Natural Computing Algorithms," in *Natural Computing,* 1st ed. Ed. Berlin, Germany: Springer-Verlag Berlin Heidelberg, 2015, pp. 17–20.

[7] Horváth, Tomáš and Carvalho, André C. "Evolutionary computing in recommender systems: a review of recent research," *Natural Computing: An International Journal,* vol. 16, no. 3, pp. 441–462, Sep. 2017.

[8] L. Cui *et al.,* "A novel multi-objective evolutionary algorithm for recommendation systems," *Journal of Parallel and Distributed Computing,*, vol. 103, pp. 53–63, 2017.

[9] A. Ekbal, S. Saha, and C. S. Garbe, "Multiobjective Optimization Approach for Named Entity Recognition," in *PRICAI 2010: Trends in Artificial Intelligence,* 1st ed. Ed. Berlin, Germany: Springer, Berlin, Heidelberg, 2010, pp. 55–57.

[10] C. A. Coello Coello. "Evolutionary multi-objective optimization: A historical view of the field," *Computational Intelligence Magazine, IEEE,*. vol. 1, no. 1, pp. 28–36, 2006.

[11] Cong Tran *et al.,* "Clustering-Based Collaborative Filtering Using an Incentivized/Penalized User Model," *IEEE Access,*. vol. 7, pp. 62115–62125, 2019.

[12] Su, Xiaoyuan and Khoshgoftaar, Taghi M. "A Survey of Collaborative Filtering Techniques," *Advances in Artificial Intelligence,*. Jan. 2009.

[13] P. Resnick *et al.,* "Grouplens: an open architecture for collaborative filtering of netnews," *Proceedings of the ACM Conference on Computer Supported Cooperative Work,*. pp. 175–186, 1994.

[14] L. Fogel "Autonomous Automata" *Industrial Research Magazine*. vol. 2, no. 2, pp. 14–19, Feb. 1962.

[15] P. Venkatesh and K. Y. Lee "Multi-Objective Evolutionary Programming for Economic Emission Dispatch problem" *IEEE Power and Energy Society 2008 General Meeting: Conversion and Delivery of Electrical Energy in the 21st Century*, Aug. 2008.

[16] Z. Yang et al. A survey of collaborative filtering-based recommender systems for mobile internet applications. IEEE Access, v. 4, p. 3273-3287, 2016.

[17] R. Chen et al. A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. IEEE Access, v. 6, p. 64301-64320, 2018.

[18] D. Bokde, S. Girase and D. Mukhopadhyay, "Matrix factorization model in collaborative filtering algorithms: A survey", Proc. Comput. Sci., vol. 49, pp. 136-146, Apr. 2015.

[19] R. Guerraoui et al., "D2P: distance-based differential privacy in recommenders," J. Proc. VLDB Endowment, vol. 8, no. 8, pp. 862–873, Apr. 2015.

[20] H. Koohi and K. Kiani, "User based collaborative filtering using fuzzy c-means," Measurement, vol. 91, pp. 134–139, Sep. 2016.

[21] J. Lin, X. H. Yan and B. Huang, "Collaborative filtering recommendation algorithm based on SVD and fuzzy clustering", Comput. Syst. Appl., vol. 25, no. 11, pp. 156-163, 2016.

[22] S. Zahra, M. A. Ghazanfar, A. Khalid, M. A. Azam, U. Naeem and A. Prugel-Bennett, "Novel centroid selection approaches for KMeans-clustering based recommender systems", Inf. Sci., vol. 320, pp. 156-189, Nov. 2015.

[23] M. Sadeghi and S. A. Asghari, Recommender systems based on evolutionary computing: a survey. Journal of Software Engineering and Applications, v. 10, n. 5, pp. 407-421, 2017.

[24] C. Berbague, N. E. I. Karabadji and H. Seridi, An evolutionary scheme for improving recommender system using clustering. In IFIP International Conference on Computational Intelligence and Its Applications, pp. 290-301, Springer, 2018.

[25] J. Chen, H. Wang, and Z. Yan, "Evolutionary heterogeneous clustering for rating prediction based on user collaborative filtering," Swarm and Evolutionary Comput., vol. 38, pp. 35–41, Feb. 2018.

[26] J. Chen, L. Wei, and L. Zhang, "Dynamic evolutionary clustering approach based on time weight and latent attributes for collaborative filtering recommendation," Chaos, Solitons & Fractals, vol. 114, pp. 8–18, Sep. 2018.

[27] U. Liji, Y. Chai, and J. Chen, "Improved personalized recommendation based on user attributes clustering and score matrix filling," Comput. Standards & Interfaces, vol. 57, pp. 59–67, Mar. 2018.

[28] J. Chen, C. Zhao, and L. Chen, Collaborative filtering recommendation algorithm based on user correlation and evolutionary clustering. Complex & Intelligent Systems, v. 6, n. 1, pp. 147-156, 2020.

[29] N. Tohidi and C. Dadkhah, Improving the performance of video Collaborative Filtering Recommender Systems using Optimization Algorithm. International Journal of Nonlinear Analysis and Applications, v. 11, n. 1, p. 283-295, 2020.

[30] G. Schröder, M. Thiele and W. Lehner, Setting goals and choosing metrics for recommender system evaluations. In UCERSTI2 workshop at the 5th ACM conference on recommender systems, Chicago, USA, v. 23, p. 53, 2011.

[31] Q. Lin et al., Multiobjective personalized recommendation algorithm using extreme point guided evolutionary computation. Complexity, v. 2018, 2018.

[32] N. E. I. Karabadji et al., Improving memory-based user collaborative filtering with evolutionary multi-objective optimization. Expert Systems with Applications, v. 98, p. 153-165, 2018.

[33] Y. Wu et al. "Collaborative denoising auto-encoders for top-n recommender systems." Proceedings of the Ninth ACM International Conference on Web Search and Data Mining, 2016.

[34] Y. Liu, J. Cheng, C. Yan, X. Wu, and F. Chen, "Research on the matthews correlation coefficients metrics of personalized recommendation algorithm evaluation," Int. J. Hybrid Inf. Technol., vol. 8, no. 1, pp. 163–172, Jan. 2015.

[35] D. Wu, G. Zhang, and J. Lu, "A fuzzy preference tree-based recommender system for personalized business-to-business E-services," IEEE Trans. Fuzzy Syst., vol. 23, no. 1, pp. 29–43, Feb. 2015.