

# Efficient recommendations in collaborative filtering recommender system: A multi-objective evolutionary approach based on NSGA-II algorithm

Touraj Mohammadpour<sup>a</sup>, Amir Massoud Bidgoli<sup>a,\*</sup>, Rasul Enayatifar<sup>b</sup>, Hamid Haj Seyyed Javadi<sup>c</sup>

<sup>a</sup>Department of Computer Engineering, North Tehran Branch, Islamic Azad University, Tehran, Iran

<sup>b</sup>Department of Computer Engineering, Firoozkooh Branch, Islamic Azad University, Firoozkooh, Iran

<sup>c</sup>Department of Mathematics and Computer Science, Shahed University, Tehran, Iran

(Communicated by Gholamreza Askari)

---

## Abstract

The final objective of the Recommender Systems (RSs) is to offer recommendations to the user that are close to his/her taste. When the user enters the system, the most similar data cluster to the user's taste can be selected, and by creating a neighborhood of the users similar to him/her within the selected cluster, the proposal generation can be followed. Determining the appropriate number of neighbors of the user can lead to increased accuracy of the recommendations made. Due to the enormous size of the datasets, this requires more time. This study aimed to propose recommendations with the highest accuracy and in the shortest possible time through finding the best number of neighbors for the user applying the NSGA-II Multi-Objective Evolutionary Algorithm (MOEA). Here two objects of accuracy and time of recommendations are in a multi-objective state, thus a balance should be created between the two conflict objects. The simulation results on the MovieLens 100K, MovieLens 1M, Netflix and FilmTrust standard datasets indicated that the proposed MOEA was capable of providing recommendations with greater accuracy and at the proper time, hence it could improve the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Coverage, Precision, Recall and  $F_{measure}$  criteria.

Keywords: Accuracy, Multi-Objective Evolutionary Approach, NSGA-II Algorithm, Recommender System, Time  
2020 MSC: 58E17

---

## 1 Introduction

Recommender Systems (RSs) are information systems capable of analyzing past behaviors of users and offering recommendations for current problems [1, 2, 3, 4, 5]. In collaborative filtering RSs, it is tried to identify the nearest item to the user's taste and recommend it through identifying the user's way of thinking with the help of information obtained from the individual's behaviors and opinions as well as similar users. Therefore, the main purpose of many methods presented in these systems is to provide recommendations for the users with the accuracy desired [6, 7, 8].

---

\*Corresponding author

Email addresses: [mohammadpour.touraj@gmail.com](mailto:mohammadpour.touraj@gmail.com) (Touraj Mohammadpour), [drbidgoli@gmail.com](mailto:drbidgoli@gmail.com) (Amir Massoud Bidgoli), [r.enayatifar@gmail.com](mailto:r.enayatifar@gmail.com) (Rasul Enayatifar), [hamid.h.s.javadi@gmail.com](mailto:hamid.h.s.javadi@gmail.com) (Hamid Haj Seyyed Javadi)

When a user enters a system, the most similar data cluster to the active user can be chosen from the existing clustered data to generate recommendations, and the information filtering can be exclusively focused on the data existing in the selected cluster. A collaborative filtering RS attempts to select a fixed number of the most similar users among the users in the selected cluster to the active user as his/her neighbors and, according to their tastes, offer recommendations for the active user [9, 10, 11, 12, 13]. Given the number of users in the selected cluster, finding the proper number of neighbors ( $K$ ) able to provide the highest recommendation accuracy can be considered as a research area. Since this occurs in the system operation time (online phase), it is necessary to consider time in order to select the most suitable neighborhoods (constant  $K$ ) for maximizing the accuracy. That is, here two objects of accuracy and time of the provided recommendations by the system are in a multi-objective state, and thus a balance should be created between the two conflict objects.

Due to the enormous size of the dataset in the system, the problem solving space for finding a suitable neighborhood is very large. In fact, this is an NP-hard problem. Given that this is a multi-objective problem and also the problem solving space is discrete, in this paper, a Multi-Objective Evolutionary Algorithm (MOEA) based on Non-dominated Sorting Genetic Algorithm II (NSGA-II) has been presented to determine the most appropriate number of neighbors for the active user, so that it can produce more accurate recommendations at a suitable time for the user. The NSGA-II algorithm offers a faster solution of sorting in comparison with other methods, in addition to eliminating the computational complexity of the previous algorithms.

Several multi-objective methods have been introduced so far aiming to offer recommendations with higher accuracy in the RSs [14, 15]. In 2017, Cui et al. [16] presented an MOEA for RSs called PMOEA and tried to establish a balance between two objects of accuracy and prediction diversity in their proposed method. They suggested a new diversity index that can be used as a measure of the variety of items in a recommended list. In the PMOEA method, a new probabilistic genetic operator is proposed, which is actually a crossover operator for generating a new solution and is called a multi-parent probabilistic genetic operator. Extensive experiments revealed that the combination of PMOEA method and the recommendation algorithm was capable of creating a good balance between accuracy and diversity to generate recommendations in an RS.

Zuo et al. [17] presented a multi-objective recommendation model in 2015. In their model, the task of personalized recommendation was modeled as a multi-objective optimization problem that attempted to maximize the two conflicting performance objects of accuracy and diversity. In this model, the accuracy criterion was evaluated by a probabilistic spreading method, meanwhile the diversity criterion was assessed using the recommendation coverage. Their proposed MOEA-based recommendation method can provide several recommendations simultaneously for multiple users at a single run. The simulation results indicated that the proposed model was able to provide recommendations with greater diversity and accuracy in comparison to the similar methods.

In 2013, Ortega et al. [18] introduced a pre-filtering process to eliminate less representative users from the active user neighborhood selection process using the Pareto dominance method. They claimed that in a traditional collaborative filtering method, users are chosen as an active user's neighbor inappropriately, which can cause the recommendations provided by the system not to be sufficiently precise. However, their proposed approach aimed to enhance the accuracy of the system recommendations by choosing the suitable users as an active user's neighbors. The results of the experiments on the MovieLens and Netflix datasets showed that the proposed method resulted in the improvement of all quality measures tested.

Ribeiro et al. [15] presented a hybrid recommendation system in 2012, a combination of existing algorithms with different levels of accuracy, diversity and novelty. Their goal was to simultaneously improve the three conflicting objects of accuracy, diversity and novelty. In the proposed method, an evolutionary search algorithm for hybrids following the Pareto method was employed, each of which non-dominated by the other (which is called the Pareto front). The experimental findings of the proposed method could be indicative of two scenarios. First, the recommendation algorithms can be combined in order to improve an object without worsening other objects, and secondly, while the appropriateness of recommendations to the users can be dynamically adjusted according to the needs of different users, a compromise can also be established among the objects of accuracy, diversity and novelty.

Geng et al. [19] introduced a new RS based on the multi-objective optimization algorithm in 2015. Their goal of this system was to simultaneously optimize the conflicting objects of accuracy and diversity. The proposed algorithm could generate a set of non-dominated solutions in each run, with each solution specifying a unique recommendation list for the target user. Experimental results were indicative of diversity in their proposed algorithm in terms of the offered recommendations.

Xu [35] propose a novel recommendation method based on MapReduce framework in 2019. The goal of his method is to handle well the multiple conflicts between accuracy and diversity. In MapReduce framework, a block computational

technique is used to shorten the operational time. Also, an improved collaborative filtering model is refined with a novel similarity computational process which considers many factors.

In general, in the MOEA proposed in this study, first, based on the items rated by the active user, the closest cluster to his/her preferences is determined. Then, given the similarity of the users in the chosen cluster to the active user, a population of potential solutions is created randomly. In the next step, in each iteration of the loop, the population is improved performing the various steps of the NSGA-II algorithm, and then the number of neighbors ( $K$ ) of the active user is determined on the basis of the chromosome with the best fitness in the population. Subsequently, based on the weighted average of items rated by the neighbors, the desirability of these items is predicted for the active user, and a number of the most appropriate items are offered to him/her. This process continues by re-implementation of the NSGA-II algorithm and relative improvement of the existing population, and thus determining the appropriate amount for  $k$ , and this process can be followed until achieving the active user's desired accuracy for the generated recommendations. To evaluate the proposed approach, the Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Coverage, Precision, Recall and  $F_{measure}$  criteria were examined on the MovieLens 100K, MovieLens 1M, Netflix and FilmTrust standard datasets.

The remaining of the paper is organized as follows: the preliminaries are introduced in Section 2. Section 3 provides a detailed description of the proposed MOEA. In Section 4, the results of simulation of the proposed approach are presented and finally, the conclusion will be expressed in Section 5.

## 2 Preliminaries

### 2.1 Collaborative Filtering Recommender System

The collaborative filtering approach is a type of Recommender Systems (RSs) that by collecting taste information from many similar users to the target user, makes a recommendation for him/her. One of the problems of this method is the cold start issue emerging for new users the opinions or interests of whom are unknown to the system. Data sparsity and scalability are another problems of these systems [6, 9, 20, 32, 33, 34].

### 2.2 Non-dominated Sorting Genetic Algorithm II

Non-dominated Sorting Genetic Algorithm II (NSGA-II) was introduced by Deb et al. in 2000. This algorithm is an elitist multi-objective evolutionary optimization algorithm that is recognized by the non-dominated sorting concept [21]. The general trend of the NSGA-II algorithm is that at first, the initial population of random solutions with a size  $N$  called  $P_0$  is generated. Then, using genetic operators such as crossover and mutation, a new population of size  $N$  is generated and merged with the previous population, leading to a population of  $2N$  size called  $R_0$ . At this stage, the non-dominated sorting is performed on the population  $R_0$  and the front number and the crowding distance of each of the solutions are determined. To create the next generation, a set of solutions of size  $N$  is required. To perform this, first, solutions with the smaller number of the Pareto front (the solutions closer to the optimal Pareto edge) will be selected to move to the next generation. In case of a front incapable of completely transferring to the next generation and completing it, the solutions on that front are sorted based on their density, and initially the solutions with the least amount of density (the largest crowding distance) are transferred to the next generation until its capacity is completed. Thus, the new generation with size  $N$  called  $P_1$  forms and replaces  $P_0$ . The above process will continue in an iteration loop until reaching the convergence criteria and termination conditions [21, 22, 23, 24, 25].

The NSGA-II algorithm offers a faster solution of sorting in comparison with other methods, in addition to eliminating the computational complexity of the previous algorithms. Moreover, this algorithm uses the crowding distance to obtain a more uniform solution fronts compared with other algorithms and uses the density estimates of points around the solutions. It is worth noting that the crowding distance is a factor which is used for better selection of the solutions in terms of dispersion on a front [21, 26].

## 3 Proposed Approach

When a user enters a system, the most similar data cluster to the active user can be chosen from the existing clustered data to generate recommendations. A collaborative filtering Recommender System (RS) attempts to select a fixed number of the most similar users ( $K$ ) among the users in the selected cluster to the active user as his/her neighbors. Finding the proper number of neighbors able to provide the highest recommendation accuracy can be considered as a research area. Due to the enormous size of the datasets in the system that contains a large number

of users, items, and user ratings on items, the problem solving space for finding a suitable neighborhood to maximize accuracy is very large. In fact, this is an NP-hard problem. Since this problem occurs in the system operation time (online), it is necessary to consider the opposite parameters of accuracy and time of the provided recommendations by the system simultaneously. In fact, two objects of accuracy and time are in a multi-objective state, and thus a balance should be created between them. Given that this is a multi-objective optimization problem and the problem solving space is discrete, the NSGA-II multi-objective evolutionary algorithm is used in the proposed evolutionary approach, because, it offers a faster solution of sorting in comparison with other methods.

Therefore at first, in order to enhance the speed and performance of the proposed approach in the online phase of the RS, the similarity among the items as well as the similarity among the users in the system are calculated offline. In this paper, the procedure of these calculations is presented in the following.

The similarity among the items was calculated using the Pearson correlation coefficient and Equation 3.1 [18].

$$PC(a, b) = \frac{\sum_{i \in U_{ab}} (R_{ai} - \bar{R}_a) \times (R_{bi} - \bar{R}_b)}{\sqrt{\sum_{i \in U_{ab}} (R_{ai} - \bar{R}_a)^2} \times \sqrt{\sum_{i \in U_{ab}} (R_{bi} - \bar{R}_b)^2}} \tag{3.1}$$

In the above equation,  $U_{ab}$  is the set of users who scored both  $a$  and  $b$  items.  $R_{ai}$  and  $R_{bi}$  are the scores given by user  $i$  to items  $a$  and  $b$ , respectively. In addition,  $\bar{R}_a$  and  $\bar{R}_b$  designate the mean scores received respectively by items  $a$  and  $b$ .

Furthermore, using Pearson correlation coefficient, the similarity among the users was calculated based on Equation 3.2 [27].

$$PC(u, v) = \frac{\sum_{i \in I_{u,v}} (R_{u,i} - \bar{R}_u) \times (R_{v,i} - \bar{R}_v)}{\sqrt{\sum_{i \in I_{u,v}} (R_{u,i} - \bar{R}_u)^2} \times \sqrt{\sum_{i \in I_{u,v}} (R_{v,i} - \bar{R}_v)^2}} \tag{3.2}$$

In the above equation,  $PC(u, v)$  is the degree of similarity of user  $u$  to user  $v$ .  $I_{u,v}$  is the set of items scored by both users  $u$  and  $v$ .  $R_{u,i}$  and  $R_{v,i}$  are scores given to the  $i$ -th item by the users  $u$  and  $v$ , respectively. Moreover,  $\bar{R}_u$  and  $\bar{R}_v$  are the average scores given by the users  $u$  and  $v$  to the items which are obtained through Equations 3.3 and 3.4, respectively [27].

$$\bar{R}_u = \frac{1}{|I_{u,v}|} \times \sum_{i \in I_{u,v}} R_{u,i} \tag{3.3}$$

$$\bar{R}_v = \frac{1}{|I_{u,v}|} \times \sum_{i \in I_{u,v}} R_{v,i} \tag{3.4}$$

Since the proposed approach is run in the context of a clustered dataset, once the user logs in to the system, initially the closest cluster to his/her preference is determined. Then, users who have evaluated the set of items in that cluster are considered to be the active user's neighbors and, based on their similarity to him/her, are arranged in order from the highest similarity to the lowest in an array (*Rank* array). In the next step, given the arranged array of neighbors, the initial population of the NSGA-II algorithm is generated. After that, the population is improved in an iterative loop using this algorithm, and recommendations are presented to the active user at the end of each cycle of the iteration loop. The flowchart of the proposed MOEA in the collaborative filtering RS has been depicted in Figure 1.

Taking into account the flowchart, the proposed approach consists of three phases, each of which is described below.

### 3.1 Phase I: Selection of Suitable Cluster

The pseudocode of the first phase of the proposed approach is presented in Table 1.

**Table 1.** Pseudocode of the first phase of the proposed approach

First phase pseudo-code (The best cluster selection)	
1. BEGIN	
2. $AU = \text{Login}();$	// Active user login to the system
3. $BC = \text{S\_cluster}(AU);$	// Select the best cluster for active user
4. END	

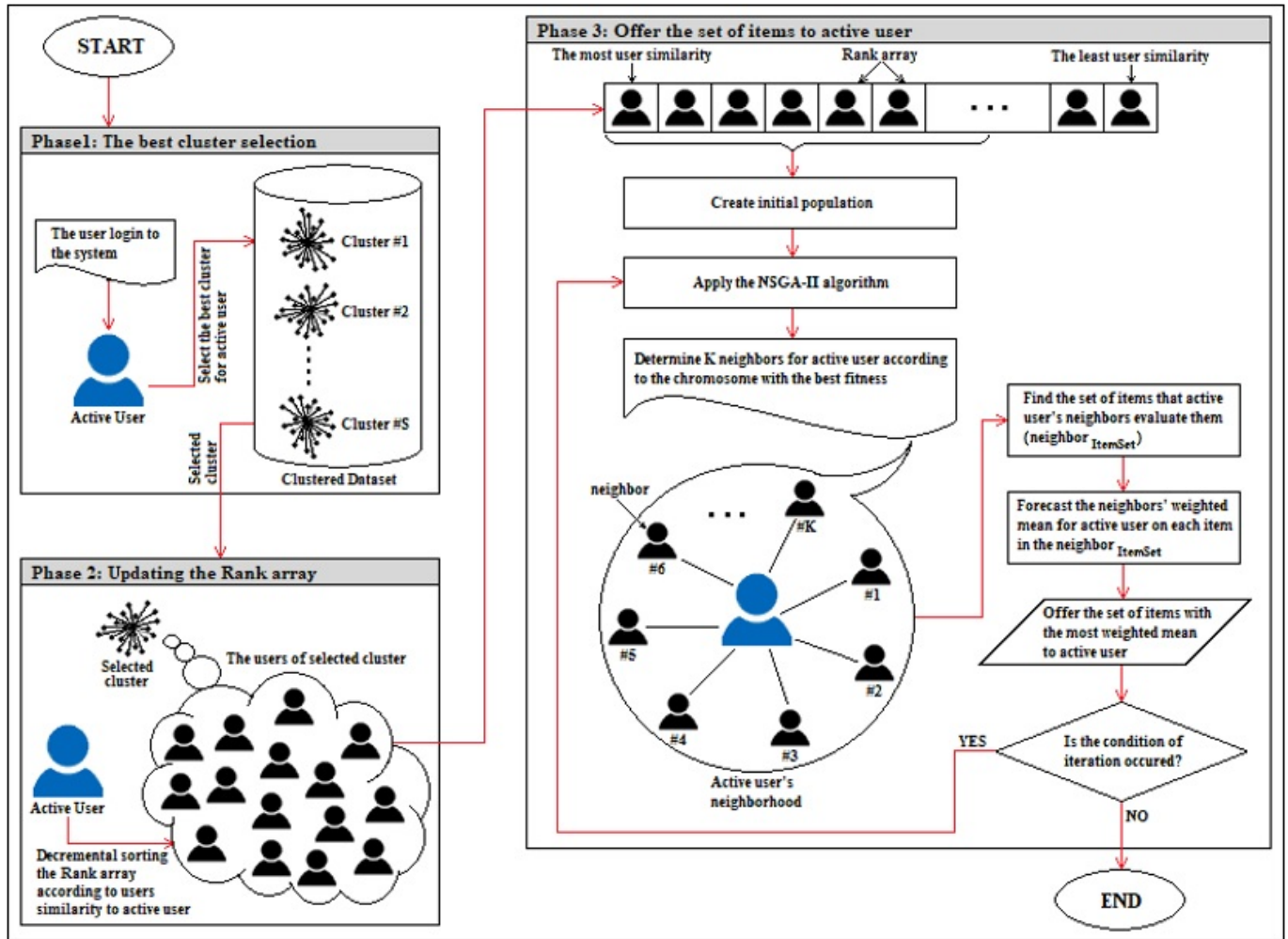


Figure 1. Flowchart of the proposed MOEA in the collaborative filtering RS

When the active user logs in to the system, the implementation of the proposed approach begins to produce the recommendation with an appropriate accuracy in the online phase. The first phase of implementation of the proposed approach is the selection of one of the clusters for the active user, so that the most appropriate items in the selected cluster can be introduced to the user. This will be achieved in case the items in the selected cluster have the highest similarity to the active user's taste. To identify such a cluster, the similarity among the items scored by the active user must be obtained with the cluster head of each of the clusters in the dataset. Since the cluster head has the highest similarity to the items within that cluster, the similarity among the items scored by the active user with the cluster head is the same as the similarity to the items within that cluster; this itself means the degree of closeness of the items of that cluster to the user's taste. The similarity of the items scored by the active user to the cluster head of the  $j$ -th cluster is calculated through Equation 3.5.

$$Similarity_j = \sum_{i \in AU_{Items}} PC(i, ClusterHead_j) \quad (3.5)$$

In the above equation,  $AU_{Items}$  is the set of items scored by the active user, and  $ClusterHead_j$  is the cluster head of the  $j$ -th cluster. Now, the cluster the cluster head of which has the most similarity with the items scored by the active user is selected as the candidate cluster in accordance with Equation 3.6.

$$CandidateCluster = \max_{1 \leq j \leq S} \{Similarity_j\} \quad (3.6)$$

In the above equation,  $S$  is the number of clusters in the system.

### 3.2 Phase II: Ranking of Users

Table 2 shows the pseudocode of the second phase of the proposed approach.

**Table 2.** Pseudocode of the second phase of the proposed approach

Second phase pseudo-code (Create the Rank array)		
1.	<b>BEGIN</b>	
2.	$CU = F\_users(BC);$	// Find the users that evaluate items
3.	$Rank = D\_sort(CU);$	// Create the Rank array
4.	<b>END</b>	

At this stage, the member users of the candidate cluster are arranged in a list based on the amount of similarity to the active user. Any active user in the system who has scored at least one of the items of the candidate cluster is considered to be a member of that cluster, and in a collaborative mode, his interests can be used to generate a recommendation to an active user. However, these users are ranked based on the similarity to the active user from the most similar to the lowest in a list called *Rank*. In fact, each member user of the selected cluster will be given a weight based on his similarity to the active user, and these users will be arranged from a higher to lower weight in the *Rank* array.

### 3.3 Phase III: Generation of Recommendations

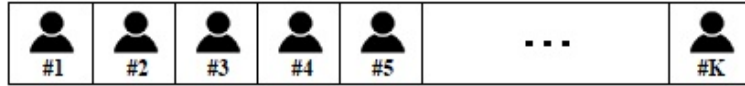
Table 3 demonstrates the pseudocode of the proposed approach in the third phase.

**Table 3.** Pseudocode of the third phase of the proposed approach

Third phase pseudo-code (Offer the set of items to active user)		
1.	<b>BEGIN</b>	
2.	$t = 1;$	// Start with an initial time
3.	$P(t) = \text{Init\_pop}(Rank);$	// Create the initial population of NSGA-II algorithm
4.	$\text{Fit\_evaluate}(P(t));$	// Chromosomes' fitness appointment
5.	$\text{Fronting}(P(t));$	// Determine the number of pareto front
6.	$\text{Crowding\_distance}(P(t));$	// Calculate crowding distance
7.	<b>REPEAT</b>	
8.	$P'(t) = \text{S\_parents}(P(t));$	// Select the pairs of chromosomes to reproduce
9.	$P''(t) = \text{Crossover}(P'(t));$	// Produce off-springs population
10.	$\text{Mutation}(P''(t));$	// Produce mutated off-springs
11.	$\text{Fit\_evaluate}(P''(t));$	// Off-springs' fitness appointment
12.	$H(t) = \text{Compound\_pop}(P(t), P''(t));$	// Create hybrid population
13.	$\text{Fronting}(H(t));$	
14.	$\text{Crowding\_distance}(H(t));$	
15.	$P(t+1) = \text{Next\_pop}(H(t));$	// Determine the population of next generation
16.	$Best = \text{Find\_chrom}(P(t+1));$	// Find the chromosome with the best fitness
17.	$Item\_set[ ] = \text{Chrom\_item}(Best);$	// Find the set of items in the best chromosome
18.	$WM[ ] = \text{Forecast}(Item\_set[ ]);$	// Forecast the neighbors' weighted mean on each item
19.	$\text{Recommend\_item}(WM[ ]);$	// Offer the set of items with the most weighted mean
20.	$t = t + 1;$	// Increase the time (iteration) counter
21.	<b>UNTIL</b> terminate iteration condition;	
22.	<b>END</b>	

#### 3.3.1 Creation of the Initial Population

In the third phase of the proposed approach, initially, the initial population of the NSGA-II algorithm is generated randomly to the number of  $N$  based on the Rank array, so that each chromosome of the initial population will contain  $K$  initial users of the *Rank* array as the order within the array, with the  $K$  values being randomly determined for the chromosomes. Figure 2 illustrates a sample chromosome in the initial population of the NSGA-II algorithm.



**Figure 2.** A sample chromosome in the initial population of the NSGA-II algorithm

### 3.3.2 Calculation of the Chromosome Fitness

The fitness level of each of the chromosomes in the population relies on the two conflicting objectives considered in the proposed approach, namely, the accuracy of the recommendation and its generation time. So, if the recommendations are generated with higher accuracy and in a less time for the active user in a chromosome, then that chromosome will have a higher fitness. In order to measure the accuracy of the recommendations generated in a chromosome, first, the suitability of each item in that chromosome ( $neighbors_{ItemSet}$ ) scored by its users ( $K$  neighbor users) should be evaluated to be recommended to the active user. That is, based on the scores given by the neighbors' weighted combination, a prediction is made about the suitability of each item to be recommended to the active user. This prediction is calculated as the neighbors weighted mean for the active user ( $AU$ ) on the item in a chromosome as Equation 3.7 [18].

$$P(AU, i) = \overline{R_{AU}} + \frac{\sum_{u \in K} (R_{u,i} - \overline{R_u}) * PC(AU, u)}{\sum_{u \in K} PC(AU, u)} \quad (3.7)$$

In the above equation,  $\overline{R_{AU}}$ ,  $K$ ,  $PC(AU, u)$ ,  $R_{u,i}$  and  $\overline{R_u}$  are the mean score given by the active user, active user's radius of neighborhood (the number of users in the chromosome), similarity between the active user and the user  $u$  available in the chromosome, score given by user  $u$  to item  $i$ , and mean score given by user  $u$ , respectively. Now, based on the prediction made for each of the items in the chromosome, some of these items with the highest prediction (suitability for the active user) should be offered to the active user by the RS as Equation 3.8 in the form of a recommended set of items.

$$RecommendedItems = \{The\ set\ of\ items\ with\ the\ most\ P(AU, i)\} \quad (3.8)$$

In the proposed approach, the accuracy of the items proposed to the active user in each chromosome is obtained based on the average of the two MAE and  $F_{measure}$  criteria, which are calculated according to Equations 4.2 and 4.8, respectively. The values obtained will be for accuracy in the interval  $[0, 1]$ . To calculate the accuracy of each chromosome as Equation 3.9:

$$Accuracy = \frac{(1 - MAE) + F_{measure}}{2} \quad (3.9)$$

As the computation time to generate the proposed set of items for the active user in the chromosome increases with the increased  $K$  value, and in contrast, the computation time decreases with its decrease as well, so based on Equation 3.10, the time complexity of the computations in each chromosome is directly related to the value of  $K$  or the same number of neighboring users in the chromosome.

$$TimeComplexity = K \quad (3.10)$$

The fitness of each chromosome in the proposed approach depends on both objects of accuracy and time of the calculations, so that these two objects are in conflict with each other, and a balance should be established between them ultimately in order to obtain the best solution. Equation 3.11 indicates the fitness of each chromosome in the proposed approach.

$$ChromosomeFitness = \alpha(Accuracy) + \beta(1/TimeComplexity) + \gamma \quad (3.11)$$

In the above equation,  $\gamma > 0$  is a very small constant value. The coefficients  $\alpha$  and  $\beta$  are also used to create the final balance between the two conflicting objects of accuracy and time.

### 3.3.3 Fronting and Crowding Distance

At this stage, the chromosomes must first be fronted on the basis of fitness, and then calculate the crowding distance. So that first, the chromosomes in the population are divided into  $p$  Pareto front using the non-dominated

sorting method, in which the best and the worst population solutions will belong to the fronts  $P_1$  (the closest front to the optimal Pareto edge) and  $P_p$ , respectively. Then, the crowding distance of the solutions of each Pareto front is calculated separately, which for each solution, is equal to the normalized sum of the distance created by the solutions before and after that solution. In the proposed approach, in order to prioritize the initial and final solutions of each front, the infinity crowding distance is allocated to them. To calculate the crowding distance of the other intermediate solutions (e.g., the  $i$ -th chromosome) in each front, Equation 3.12 is exploited [21].

$$CrowdingDistance_i = \sum_{j=1}^M \frac{|f_j^{(i-1)} - f_j^{(i+1)}|}{f_j^{Max} - f_j^{Min}} \quad (3.12)$$

In the above equation,  $M$  is the number of objective functions of the problem, which here are the accuracy and time objects.  $f_j^{(i-1)}$  and  $f_j^{(i+1)}$  are equal to the values of the previous and next solutions of the  $i$ -th solution in the  $j$ -th objective function, respectively, and  $f_j^{Min}$  and  $f_j^{Max}$  respectively are the minimum and maximum values of the  $j$ -th objective function among the solutions of the desired front.

### 3.3.4 Selection

After allocating the front number and the crowding distance to each chromosome, at this stage, the chromosomes should be selected to perform the crossover operation. To do this, the binary crowded tournament selection operator is utilized as follows [21]:

- Two chromosomes are randomly selected.
- The winner of these two chromosomes, as determined as follows, is selected for the crossover:
  1. The chromosome with a smaller front number.
  2. In case of the equal numbers of fronts, the chromosome with a larger crowding distance.

### 3.3.5 Crossover and mutation

To perform the crossover in the proposed approach, if the length of the two parent chromosomes is  $K_1$  and  $K_2$ , respectively, the bracket of one third of the difference between the lengths ( $\lfloor |K_1 - K_2| / 3 \rfloor$ ) is considered to be  $K'$  and using the *Rank* array, the offspring chromosomes are generated with lengths calculated according to Equation 3.13.

$$\begin{cases} K_1 + K' & , & K_1 + 2K' & & K_1 < K_2 \\ K_1 - 1 & , & K_2 + 1 & & K_1 = K_2 \\ K_1 - K' & , & K_1 - 2K' & & K_1 > K_2 \end{cases} \quad (3.13)$$

In the above equation, if  $K' < 1$ , then  $K_1 = K_2$ . Now a percentage of the offspring chromosomes is randomly selected to perform the mutation, and taking into account the *Rank* array, their length is increased by one unit.

### 3.3.6 Population of the Next Generation

At this stage, the fitness of each offspring chromosome is first calculated according to 3.11, and then the parent population and the offspring population are combined, hence a population of size  $2N$  is achieved. In the next step, the non-dominated sorting operation is performed on the new population and new fronts are formed. Now a population of size  $N$  must be selected and moved to the next generation. For this purpose, first, the fronts with a smaller number capable of transferring to the next generation are selected in the order of priority. Then, regarding the remaining capacity of the next generation, from among the front that cannot be completely transferred, chromosomes with a larger crowding distance are moved to the next generation in the order of priority until the capacity is filled [21].

### 3.3.7 Generation of Recommendation

Now, given the new generation, a set of items must be created to be offered to the active user. Therefore in the new generation, the chromosome with the best fitness is selected. Then, taking into account the number of neighboring users in this chromosome ( $K$ ), a prediction is performed regarding the suitability of each item of that chromosome to be offered to the active user according to Equation 3.7. After that, based on Equation 3.8, a set of items is selected



to recommend to the active user and present to him/her. In the next step, the proposed approach calls the NSGA-II algorithm with the new generation and will try again to improve its solutions.

The above process is performed in the third phase of the proposed approach in each round of the iteration loop for the active user, and in each loop, it is tried to improve the solutions of that generation using the NSGA-II algorithm so that at the end of that loop, by achieving the best  $K$  value, a set of items with higher accuracy can be created for the active user. This process continues until a suitable balance is achieved between the time and accuracy of the produced recommendations. Therefore, it can be declared that in the proposed approach, gradually with increasing waiting time, the user will receive the set of the recommended items from the system with higher accuracy and quality. As this approach can practically produce the entire Pareto front solutions for the active user to establish a balance between the objects of time and accuracy of the recommendations.

## 4 Experimental Results

### 4.1 Experimental Setup

At first, the simulation platform, data explanation and algorithm initializing for evaluating the proposed approach are introduced.

#### 4.1.1 Simulation Platform

For simulating the proposed approach, the MATLAB R2017a software was exploited on a computer with 2.3 GHz Intel Core i7 microprocessor, 8 GB RAM, 500 GB hard drive and 64-bit Windows 10 Operating System.

#### 4.1.2 Data Explanation

The proposed multi-objective approach is evaluated with four standard datasets of the MovieLens 100K, MovieLens 1M, Netflix and FilmTrust. In two MovieLens datasets of 100K and 1M, each rating is made up of four fields of User-ID, Item-ID, Rating value and Timestamp, respectively. Each evaluation in the Netflix dataset consists of four fields of respectively the number of the movie, Customer-ID, Rating and Evaluation date. Each rating in the FilmTrust dataset consists of three fields of User-ID, Item-ID and Rating value, respectively. Table 4 demonstrates the general specifications of the datasets utilized.

**Table 4.** General specifications of the datasets used to evaluate the proposed approach

Dataset	#Users	#Movies	#Ratings	Min-Max Values
MovieLens 100K	943	1682	100000	1 - 5
MovieLens 1M	6040	3900	1000209	1 - 5
Netflix	480189	17770	100480507	1 - 5
FilmTrust	1508	2071	35497	0.5 - 4

#### 4.1.3 Algorithm Initializing

In each iteration of the loop in the third phase of the proposed approach, after the implementation of the NSGA-II algorithm, by determining the most suitable value for the number of neighbors ( $K$ ), a triple set of the most appropriate items is suggested to the active user. The values of some other parameters in the third phase of the proposed approach are indicated in Table 5.

**Table 5.** Values of some parameters of the proposed approach in the third phase

Parameter	Value
Initial Population (No. of Chromosomes)	200
Crossover Rate	100%
Mutation Rate	5% of Childs
No. of Iteration Loop	500

## 4.2 Measurement Criteria

For testing the proposed multi-objective approach, six measures of Mean Absolute Error (MAE), Root Mean Square Error (RMSE), Coverage, Precision, Recall and  $F_{measure}$  are investigated.

MAE measures the mean error in the predicted rating ( $p_{u,i}$ ) by the system vs. the true rating ( $r_{u,i}$ ) by the user. This measure for the active user  $u$  is equal to Equation 4.1 [2, 9, 20, 27, 28, 29].

$$MAE_u = \frac{1}{n_u} \sum_{i=1}^{n_u} |p_{u,i} - r_{u,i}| \quad (4.1)$$

$n_u$ , is the cardinality of the test ratings set. Then, the total MAE is equal to Equation 4.2.

$$MAE = \frac{1}{\#U} \sum_{u=1}^{\#U} MAE_u \quad (4.2)$$

In the above equation,  $\#U$  is the total number of active users. RMSE for the active user  $u$  is equal to Equation 4.3 [2, 20].

$$RMSE_u = \sqrt{\frac{1}{n_u} \sum_{i=1}^{n_u} (p_{u,i} - r_{u,i})^2} \quad (4.3)$$

Thus, the total RMSE is computed by Equation 4.4.

$$RMSE = \frac{1}{\#U} \sum_{u=1}^{\#U} RMSE_u \quad (4.4)$$

Coverage illustrates the percentage of items that the RS was able to recommend. This measure is equal to Equation 4.5 [2, 9, 20, 28].

$$Coverage = \frac{\sum_{u=1}^{\#U} q_u}{\sum_{u=1}^{\#U} n_u} \quad (4.5)$$

In the above equation,  $q_u$  is the total number of predicted items for active user  $u$ .

Precision indicates that how much of the recommended set is really interested by the active user and correct. This criterion is calculated as Equation 4.6 [29, 30, 31].

$$Precision = \frac{|Recommended\ Items \cap Favorite\ Items|}{|Recommended\ Items|} \quad (4.6)$$

Recall shows how much of the items that are actually of interest to the user are recommended by the system. This criterion is calculated based on Equation 4.7 [29, 30, 31].

$$Recall = \frac{|Recommended\ Items \cap Favorite\ Items|}{|Favorite\ Items|} \quad (4.7)$$

Due to the parallel performance of Precision and Recall criteria, they should be integrated. This was performed using a criterion called  $F_{measure}$ , which was the harmonic mean of the two criteria of Precision and Recall. To calculate this criterion based on Equation 4.8, we have [30, 31]:

$$F_{measure} = \frac{2 * (Precision * Recall)}{Precision + Recall} \quad (4.8)$$

## 4.3 Proposed Approach Performance

Now, the simulation results of the proposed MOEA approach have been presented. Moreover, the performance of the proposed approach has been compared with another collaborative filtering methods. The results presented have been obtained for 30 simulations on the standard datasets employed.

### 4.3.1 First Experiment

In this experiment, the performance of the proposed MOEA approach has been assessed in comparison with the Pearson correlation (PC) [18, 27], PMOEA [16] and MapReduce framework (MRF) [35] methods. Therefore, these methods were first evaluated in each of the MAE, RMSE, Coverage, Precision, Recall and  $F_{measure}$  criteria on MovieLens 100K, MovieLens 1M, Netflix and FilmTrust standard datasets. Table 6 represents the minimum and maximum values obtained for each of the above criteria in the methods mentioned on the datasets used.

**Table 6.** Minimum and maximum values for each of the criteria evaluated in the PC, PMOEA and MRF methods

Dataset/Method		MAE		RMSE		Coverage(%)		Precision		Recall		$F_{measure}$	
		Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
MovieLens100K	PC	0.7100	0.9100	0.9100	1.1660	48.13	93.27	0.3541	0.4029	0.1237	0.6761	0.1833	0.5049
	PMOEA	0.6955	0.8464	0.8911	1.0846	83.54	95.11	0.3621	0.4966	0.2033	0.7483	0.2603	0.5970
	MRF	0.6829	0.7596	0.8752	0.9738	91.69	98.06	0.3682	0.5484	0.2317	0.7724	0.2844	0.6414
MovieLens 1M	PC	0.7527	0.9534	0.9649	1.2218	29.17	84.51	0.3214	0.3552	0.2008	0.7016	0.2471	0.4716
	PMOEA	0.7341	0.8817	0.9409	1.1296	41.73	92.33	0.3304	0.4621	0.2807	0.7942	0.3035	0.5842
	MRF	0.7237	0.7908	0.9274	1.0137	50.18	96.82	0.3361	0.5045	0.3166	0.8292	0.3260	0.6273
Netflix	PC	0.8620	0.9870	1.1050	1.2653	08.92	39.11	0.2900	0.3391	0.3041	0.7082	0.2968	0.4586
	PMOEA	0.7943	0.9063	1.0179	1.1615	13.61	48.90	0.3032	0.4394	0.4109	0.8257	0.3489	0.5735
	MRF	0.7756	0.8189	0.9939	1.0496	17.15	55.27	0.3111	0.4655	0.4329	0.8543	0.3620	0.6026
FilmTrust	PC	0.7300	0.9280	0.9358	1.1896	43.86	90.69	0.3417	0.3932	0.1316	0.6827	0.1900	0.4990
	PMOEA	0.7135	0.8513	0.9142	1.0909	82.22	93.71	0.3503	0.4873	0.2201	0.7546	0.2703	0.5921
	MRF	0.6992	0.7663	0.8961	0.9821	90.47	96.92	0.3592	0.5412	0.2491	0.7886	0.2941	0.6418

Then, the performance of the proposed MOEA was measured in each of the MAE, RMSE, Coverage, Precision, Recall and  $F_{measure}$  criteria on the standard datasets utilized in this study. Table 7 indicates the minimum and maximum values obtained for each of the above criteria in the proposed MOEA on the standard datasets used.

**Table 7.** Minimum and maximum values for each of the criteria evaluated in the proposed approach

Dataset	MAE		RMSE		Coverage(%)		Precision		Recall		$F_{measure}$	
	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max	Min	Max
MovieLens100K	0.6783	0.7361	0.8695	0.9436	94.68	100	0.3760	0.5641	0.2565	0.8091	0.3526	0.5510
MovieLens 1M	0.7219	0.7632	0.9254	0.9784	52.21	99.33	0.3419	0.5187	0.3426	0.8653	0.4126	0.5401
Netflix	0.7688	0.7908	0.9856	1.0138	20.85	59.32	0.3246	0.4869	0.4587	0.8872	0.4723	0.5288
FilmTrust	0.6924	0.7479	0.8876	0.9588	93.07	99.92	0.3655	0.5572	0.2714	0.8168	0.3650	0.5460

In the next step, in order to compare the performance of the proposed MOEA approach with the methods mentioned in above, the values obtained from the criteria assessed on each of the datasets in each of the four methods have been depicted in Figures 3–6. In these figures, in the MAE, RMSE and Coverage criteria, the horizontal axis indicates the number of neighbors ( $K$ ) with the values ranging from 50 to 800. In addition, in the Precision, Recall and  $F_{measure}$  criteria, the horizontal axis shows the number of recommendations suggested by the system, with values ranging from 2 to 20.

As demonstrated in Figures 3–6, the proposed MOEA approach has improved all the criteria evaluated in all datasets used in comparison to the Pearson correlation, PMOEA and MapReduce framework methods. According to the findings, in the MAE, RMSE and Coverage criteria, increase in the number of neighborhoods of the active user in all datasets has led to better results for these criteria in the proposed approach toward another methods mentioned. So, the best value for the above criteria has been obtained in the neighborhood number of  $K=800$ . The highest improvement of MAE and RMSE criteria in the proposed approach toward the methods mentioned was achieved in the neighborhood number of  $K=50$ . As, the improvement rate in these criteria has diminished with increase in the amount of  $K$ . This is due to the fact that as  $K$  increases, the neighbors become more and more similar in the four methods, and hence the trend of improvement of the above criteria in the four methods becomes similar. The improvement rate of the Coverage criterion in the MovieLens 100K and FilmTrust datasets is low by increasing the number of neighbors, however this rate is high in the MovieLens 1M and Netflix datasets with the increase in the number of neighbors. The reason is that the number of items in the MovieLens 1M and Netflix datasets is larger than that of the MovieLens 100K and FilmTrust datasets, making it easier to recommend items that are not evaluated by the active user with increasing the number of neighbors.

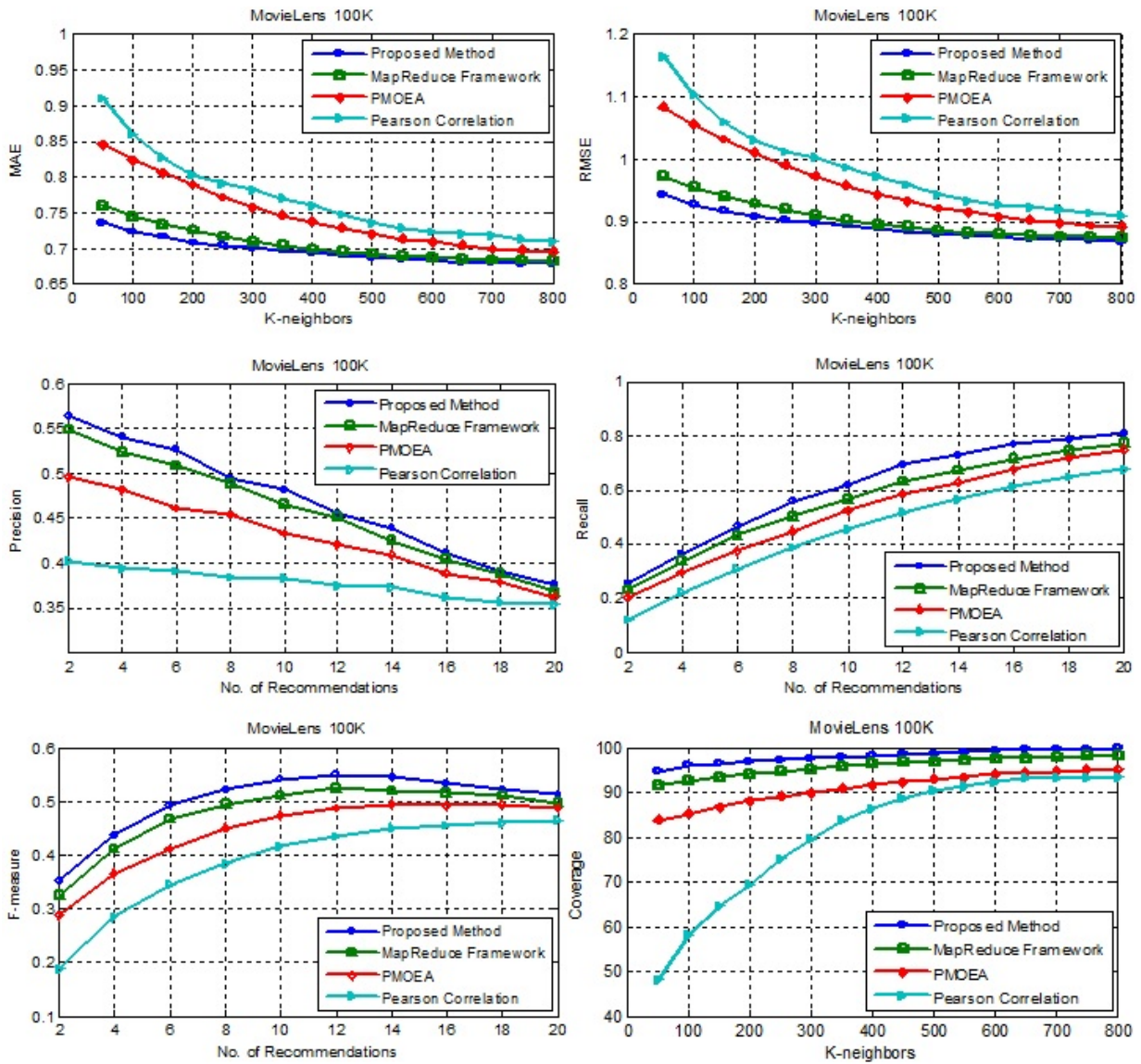


Figure 3. Values of the criteria evaluated in the proposed approach compared to the MapReduce framework, PMOEA and Pearson correlation methods on the MovieLens 100K dataset

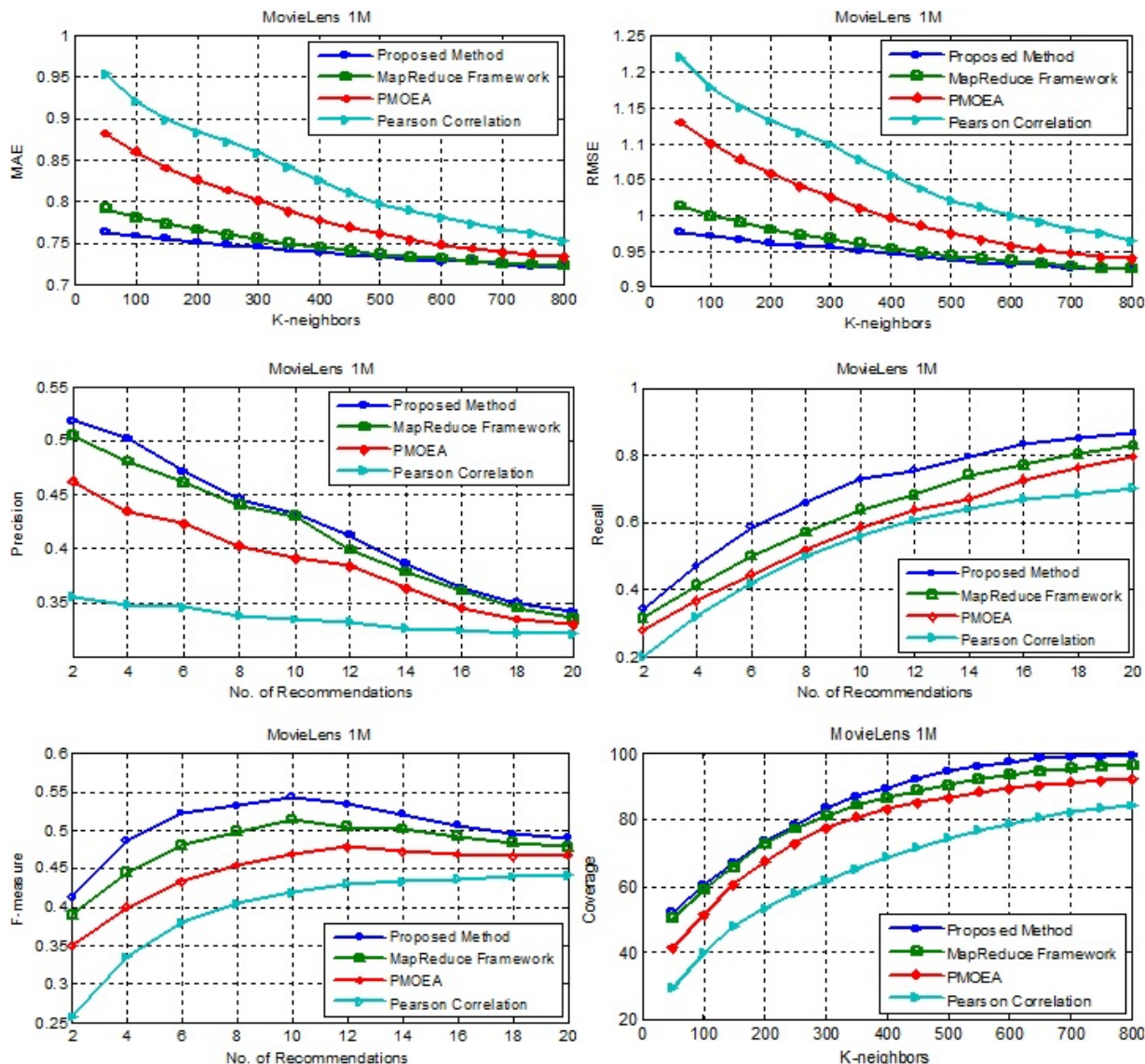


Figure 4. Values of the criteria evaluated in the proposed approach compared to the MapReduce framework, PMOEA and Pearson correlation methods on the MovieLens 1M dataset

Based on the results of the proposed approach and the methods mentioned in Figures 3-6, in the Precision criterion, with increasing the number of recommendations, the values obtained for this criterion have been declined in all datasets, however in the Recall criterion, the increase in the number of recommendations has led to an increase in the values obtained for this criterion in all datasets. The improvement of the proposed approach in the Precision and Recall criteria in all datasets and in all the number of recommendations is more than the mentioned methods. In the  $F_{measure}$  criterion, which is equivalent to the harmonic mean of the two criteria of Precision and Recall, the values obtained for the proposed approach in all datasets first increases and then decreases with increasing the number of recommendations. In addition, the improvement of the proposed approach in the  $F_{measure}$  criterion in all datasets and in all the number of recommendations is more than the mentioned methods.

In this step of the first experiment, minimum value, median, dispersion of quantities and maximum value for the MAE, RMSE, Coverage, Precision, Recall and  $F_{measure}$  criteria of the proposed MOEA approach on each of the MovieLens 100K, MovieLens 1M, Netflix and FilmTrust standard datasets are displayed by the box plot. This graph consists of a box and two bar getting out of it, and shows information such as the minimum value, the median, the dispersion of the values and the maximum amount of data obtained from the experiments. In the following, the values obtained for each of the criteria mentioned over the 30 times of simulation of the proposed approach on each of the standard datasets used are shown in Figure 7.

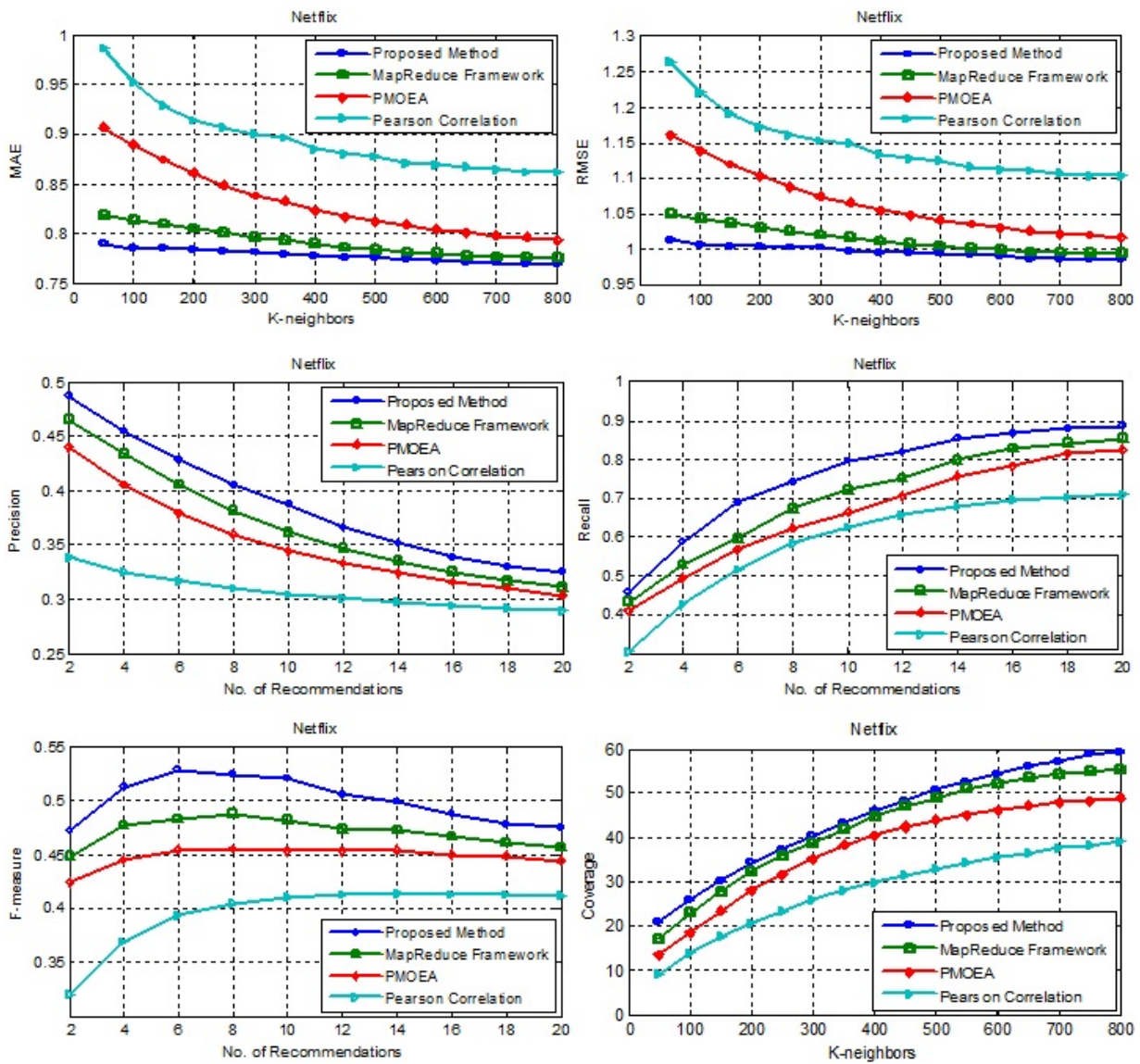


Figure 5. Values of the criteria evaluated in the proposed approach compared to the MapReduce framework, PMOEA and Pearson correlation methods on the Netflix dataset

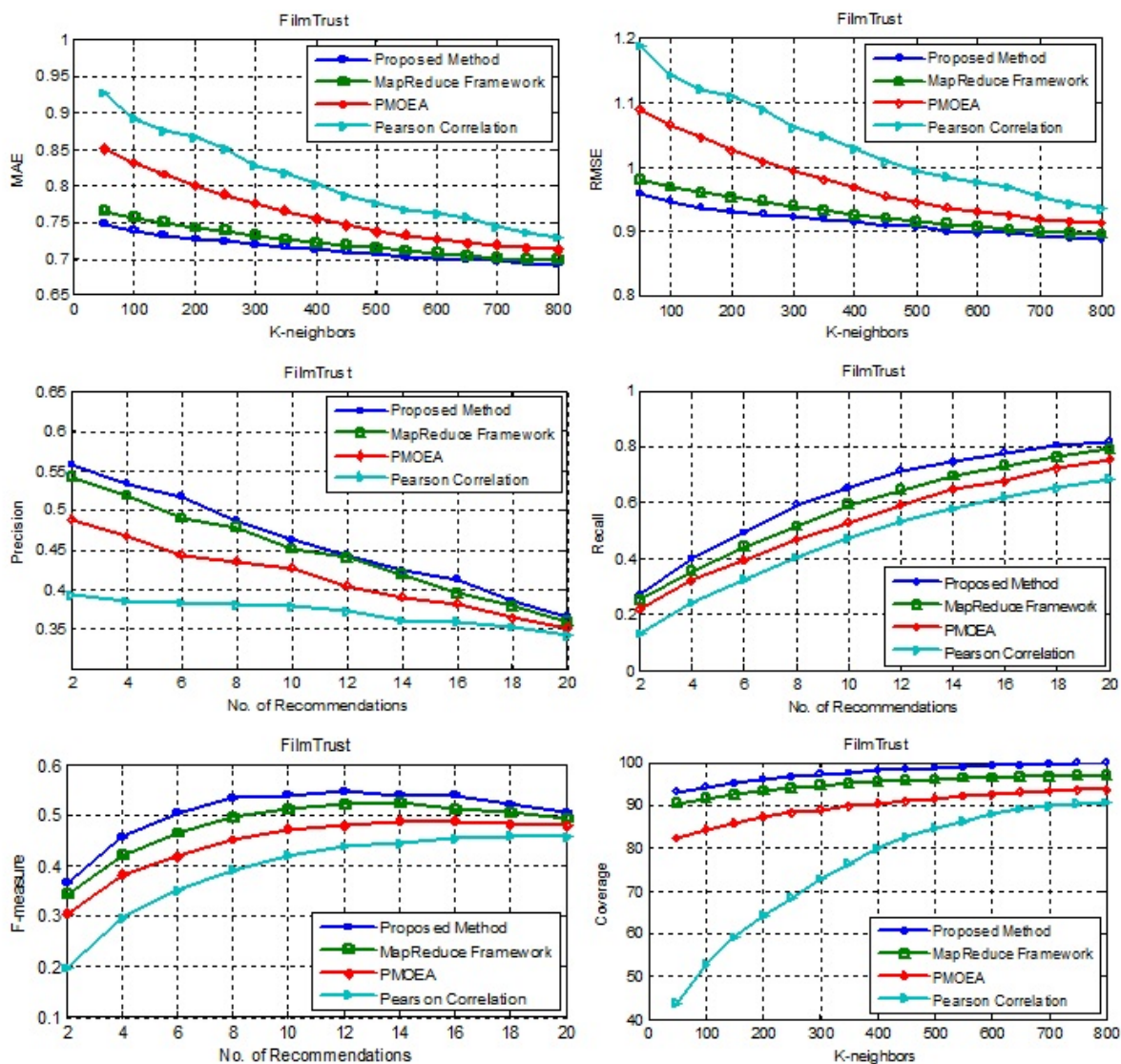


Figure 6. Values of the criteria evaluated in the proposed approach compared to the MapReduce framework, PMOEA and Pearson correlation methods on the FilmTrust dataset

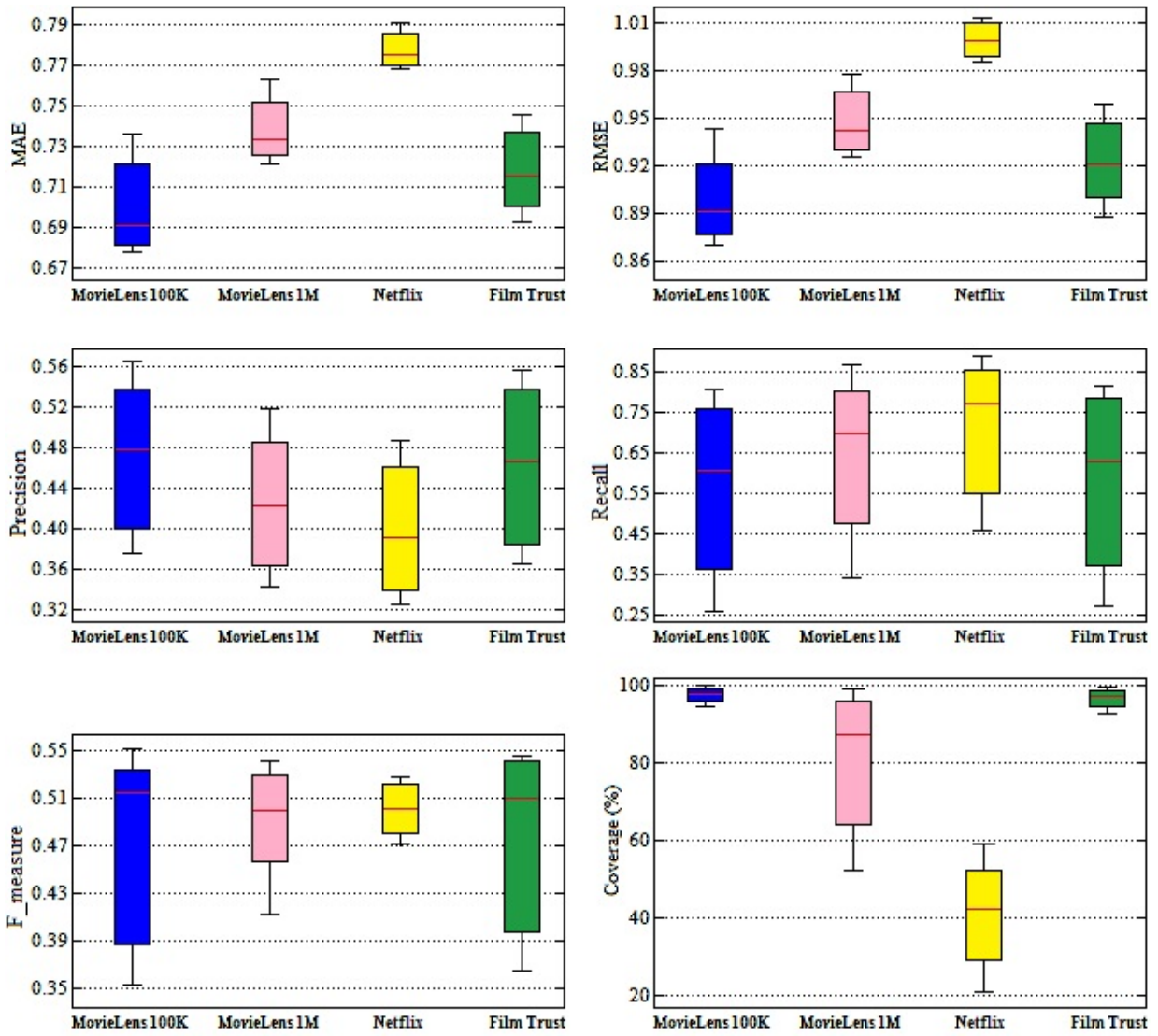


Figure 7. Minimum values, median, dispersion of quantities and maximum values of the criteria obtained in the proposed approach on the datasets



### 4.3.2 Second Experiment

The aim of the proposed MOEA approach in the current study was to balance between the two conflicting objects of the accuracy of the recommendations and the time of their presentation by the system to the active user. Therefore, in this experiment, two objects of accuracy and time of each of the recommendations suggested by the system to the active user were measured during the implementation of the proposed approach on each of the MovieLens 100K, MovieLens 1M, Netflix and FilmTrust datasets. Afterwards, the fitness of each of the solutions was assessed by the values obtained from these two objects for each solution. Then, by fronting the solutions and calculating their crowding distance, the results for the two objects of accuracy and time were demonstrated in the form of the Pareto front diagrams on each of the datasets applied. For this purpose, at first, the minimum and maximum values obtained for each of the two objects of accuracy and the time of recommendations in each of the standard datasets used were presented in Tables 8 and 9.

**Table 8.** Minimum and maximum values of the accuracy object of recommendations in the proposed approach

Dataset	Accuracy	
	Min	Max
MovieLens100K	0.3082	0.4363
MovieLens 1M	0.3247	0.4091
Netflix	0.3417	0.3854
FilmTrust	0.3085	0.4268

**Table 9.** Minimum and maximum values of the time object of recommendations in the proposed approach

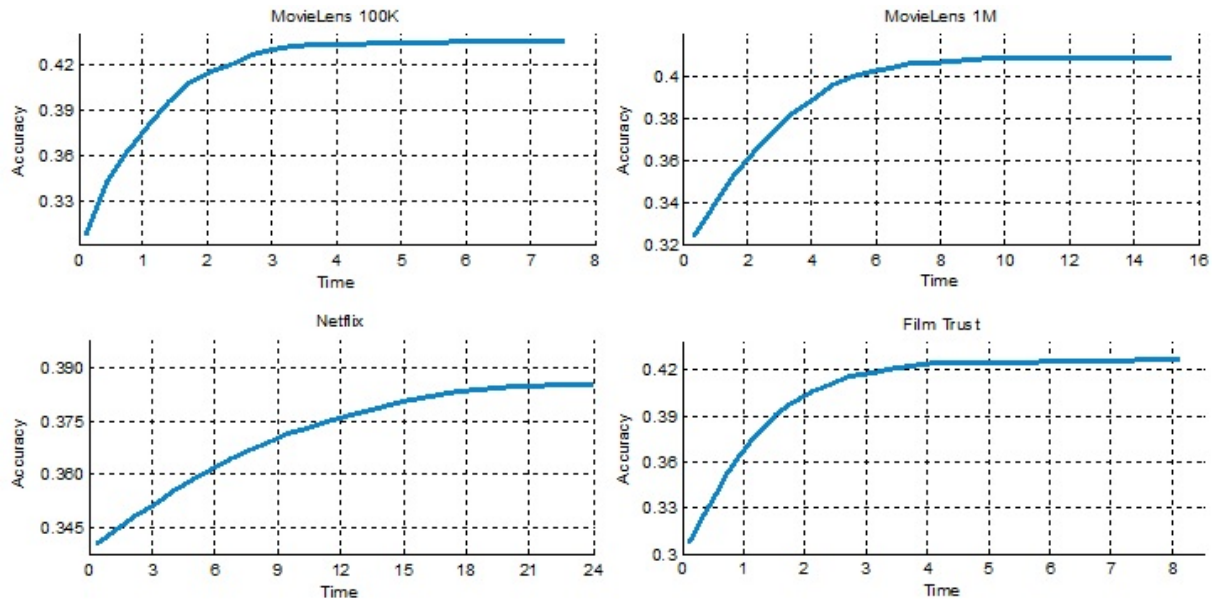
Dataset	Recommendation Time (s)	
	Min	Max
MovieLens100K	0.129	7.829
MovieLens 1M	0.342	15.731
Netflix	0.623	24.274
FilmTrust	0.151	8.293

After that, the Pareto front of the solutions provided by the system to the active user during the implementation of the proposed approach on each of the standard datasets utilized have been depicted in Figures 8 for the two objects of the accuracy and time of the recommendations.

As demonstrated in Figures 8, over time, the growth rate of the accuracy criterion of the solutions is higher in the early stages of the implementation of the proposed approach in the MovieLens 100K and FilmTrust datasets compared to the MovieLens 1M and Netflix datasets. Moreover, over time, the drop in the growth rate of the accuracy criterion in the MovieLens 100K and FilmTrust datasets will also be larger in comparison to the MovieLens 1M and Netflix datasets. This indicates that the speed of improvement in the early stages of the implementation of the proposed multi-objective evolutionary approach is higher in the MovieLens 100K and FilmTrust datasets, and in the same way and continuing the run, these two datasets show more significant drop in the speed of improvement of the solutions over time. Of course, this may be due to much lower data included in the MovieLens 100K and FilmTrust datasets compared to the MovieLens 1M and Netflix datasets. This item is also observed in the MovieLens 1M dataset as compared to the Netflix one. So that the speed of improvement of the solutions in the early stages of the implementation of the proposed multi-objective evolutionary approach in the MovieLens 1M dataset is greater than the Netflix dataset.

## 5 Conclusion

In this study, a Multi-Objective Evolutionary Algorithm (MOEA) was presented to provide the most suitable recommendations to the active user in the collaborative filtering Recommender System (RS). The proposed MOEA approach is based on the Non-dominated Sorting Genetic Algorithm II (NSGA-II). The purpose of this approach was to establish a balance between the two conflicting objects of the accuracy of the recommendations and the time of their proposal by the system, so that the recommendations can be presented to the active user with the highest accuracy and in the shortest time. In order to measure the performance of the proposed approach, the quality measures including MAE, RMSE, Coverage, Precision, Recall and  $F_{measure}$  were evaluated on the standard datasets of MovieLens 100K, MovieLens 1M, Netflix and FilmTrust. First, in a test, the performance of the proposed MOEA approach was compared with Pearson correlation, PMOEA and MapReduce framework methods in all of the desired



**Figure 8.** The Pareto front of the solutions provided to the active user on the datasets used

criteria for each of the standard datasets. The results of this test indicated that the proposed approach improved all the criteria evaluated in all datasets toward the methods mentioned. Furthermore, the improvement of the proposed approach in the MAE, RMSE and Coverage criteria in all datasets and in all the number of active user's neighborhoods is more than the mentioned methods. On the other hand, the speed of improvement is low in the Coverage criterion in MovieLens 100K and FilmTrust datasets, and is high in the MovieLens 1M and Netflix datasets. Besides, in all the number of recommendations, the improvement of the proposed approach in the Precision, Recall and  $F_{measure}$  criteria in all datasets is more than the mentioned methods. In another test, two objects of accuracy and time of each of the recommendations provided by the system to the active user during the implementation of the proposed approach were evaluated on each of the standard datasets. The results of this test, presented as Pareto front diagrams, revealed that the speed of improvement of the solutions in the early stages of implementation of the proposed multi-objective evolutionary approach in the MovieLens 100K and FilmTrust datasets is higher than that of the MovieLens 1M and Netflix datasets, and in the same way and with continuation of the run, the MovieLens 100K and FilmTrust datasets exhibit more sensible drop in the speed of improvement of the solutions over time. It should be noted that the proposed MOEA approach is independent of the data type and can be used for all types of data and have no specific limitation.

## References

- [1] F. Zhang, T. Gong, V.E. Lee, G. Zhao, C. Rong and G. Qu, *Fast algorithms to evaluate collaborative filtering recommender systems*, Knowledge-Based Syst. **96** (2016), 96–103.
- [2] J. Bobadilla, F. Ortega, A. Hernando and A. Gutiérrez, *Recommender systems survey*, Knowledge-based Syst. **46** (2013), 109–132.
- [3] L. Lü, M. Medo, C.H. Yeung, Y.-C. Zhang, Z.-K. Zhang and T. Zhou, *Recommender systems*, Phys. Rep. **519** (2012), no. 1, 1–49.
- [4] J. Lu, D. Wu, M. Mao, W. Wang and G. Zhang, *Recommender system application developments: A survey*, Decision Support Syst. **74** (2015), 12–32.
- [5] P.G. Campos, F. Díez and I. Cantador, *Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols*, User Model. User-Adapted Interact. **24** (2014), no. 1-2, 67–119.
- [6] X. Yang, Y. Guo, Y. Liu and H. Steck, *A survey of collaborative filtering based social recommender systems*, Comput. Commun. **41** (2014), 1–10.
- [7] M. Elahi, F. Ricci and N. Rubens, *A survey of active learning in collaborative filtering recommender systems*, Comput. Sci. Rev. **20** (2016), 29–50.

- [8] M. Mao, J. Lu, G. Zhang and J. Zhang, *Multirelational social recommendations via multigraph ranking*, IEEE Trans. Cybernet. **47** (2017), no. 12, 4049–4061.
- [9] J.L. Herlocker, J.A. Konstan, L.G. Terveen and J.T. Riedl, *Evaluating collaborative filtering recommender systems*, ACM Trans. Inf. Syst. **22** (2004), no. 1, 5–53.
- [10] J. Wei, J. He, K. Chen, Y. Zhou and Z. Tang, *Collaborative filtering and deep learning based recommendation system for cold start items*, Expert Syst. Appl. **69** (2017), 29–39.
- [11] A.H. Celdrán, M.G. Pérez, F.J.G. Clemente and G.M. Pérez, *Design of a recommender system based on users' behavior and collaborative location and tracking*, J. Comput. Sci. **12** (2016), 83–94.
- [12] D.H. Park, H.K. Kim, I.Y. Choi and J.K. Kim, *A literature review and classification of recommender systems research*, Expert Syst. Appl. **39** (2012), no. 11, 10059–10072.
- [13] R. Pan, P. Dolog and G. Xu, *KNN-based clustering for improving social recommender systems*, Agents Data Mining Interact. **7607** (2013), 115–125.
- [14] M. Sadeghi and S.A. Asghari, *Recommender systems based on evolutionary computing: a survey*, J. Software Engin. Appl. **10** (2017), 407–421.
- [15] M.T. Ribeiro, A. Lacerda, A. Veloso and N. Ziviani, *Pareto-efficient hybridization for multi-objective recommender systems*, Proc. 6th ACM Conf. Recommender Syst., 2012, pp. 19–26.
- [16] L. Cui, P. Ou, X. Fu, Z. Wen and N. Lu, *A novel multi-objective evolutionary algorithm for recommendation systems*, J. Paral. Distrib. Comput. **103** (2017), 53–63.
- [17] Y. Zuo, M. Gong, J. Zeng, L. Ma and L. Jiao, *Personalized recommendation based on evolutionary multi-objective optimization [research frontier]*, IEEE Computational Intel. Mag. **10** (2015), no. 1, 52–62.
- [18] F. Ortega, J.-L. SáNchez, J. Bobadilla and A. Gutiérrez, *Improving collaborative filtering-based recommender systems results using Pareto dominance*, Inf. Sci. **239** (2013), 50–61.
- [19] B. Geng, L. Li, L. Jiao, M. Gong, Q. Cai and Y. Wu, *NNIA-RS: A multi-objective optimization based recommender system*, Phys. A: Statist. Mech. Appl. **424** (2015), 383–397.
- [20] T. Mohammadpour, A.M. Bidgoli, R. Enayatifar and H.H.S. Javadi, *Efficient clustering in collaborative filtering recommender system: hybrid method based on genetic algorithm and gravitational emulation local search algorithm*, Genomics **111** (2019), no. 6, 1902–1912.
- [21] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*, Int. Conf. Paral. Prob. Solving from Nature **1917** (2000), 849–858.
- [22] S. Kannan, S. Baskar, J.D. McCalley, P. Murugan, *Application of NSGA-II algorithm to generation expansion planning*, IEEE Trans. Power Syst. **24** (2009), no. 1, 454–461.
- [23] P. Murugan, S. Kannan and S. Baskar, *NSGA-II algorithm for multi-objective generation expansion planning problem*, Electric Power Syst. Res. **79** (2009), no. 4, 622–628.
- [24] H. Li and Q. Zhang, *Multiobjective optimization problems with complicated pareto sets, MOEA/D and NSGA-II*, IEEE Trans. evolut. Comput. **13** (2009), no. 2, 284–302.
- [25] K. Deb and H. Jain, *Handling many-objective problems using an improved NSGA-II procedure*, IEEE Cong. Evolut. Comput., 2012, pp. 1–8.
- [26] C. Von Lüken, B. Barán and C. Brizuela, *A survey on multi-objective evolutionary algorithms for many-objective problems*, Comput. Optim. Appl. **58** (2014), no. 3, 707–756.
- [27] J. Bobadilla, F. Ortega, A. Hernando and J. Alcalá, *Improving collaborative filtering recommender system results and performance using genetic algorithms*, Knowledge-Based Syst. **24** (2011), no. 8, 1310–1316.
- [28] M.Y.H. Al-Shamri and K.K. Bharadwaj, *Fuzzy-genetic approach to recommender systems based on a novel hybrid user model*, Expert Syst. Appl. **35** (2008), no. 3, 1386–1399.
- [29] J. Zhang, Y. Lin, M. Lin, J. Liu, *An effective collaborative filtering algorithm based on user preference clustering*, Appl. Intell. **45** (2016), no. 2, 230–240.

- [30] M. Nilashi, K. Bagherifard, M. Rahmani and V. Rafe, *A recommender system for tourism industry using cluster ensemble and prediction machine learning techniques*, *Comput. Ind. Engin.* **109** (2017), 357–368.
- [31] M. Nilashi, O. Ibrahim and K. Bagherifard, *A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques*, *Expert Syst. Appl.* **92** (2018), 507–520.
- [32] M.Y.H. Al-Shamri, A.F. Al-Juniad, T.S. Qaid, M.H.A. Ahmed and A.A. Raweh, *Cascaded feature selection for enhancing the performance of collaborative recommender system*, *Adv. Electric. Comput. Engin.* **18** (2018), no. 4, 23–34.
- [33] Y. Bai, S. Jia, S. Wang, B. Tan, *Customer loyalty improves the effectiveness of recommender systems based on complex network*, *Inf.* **11** (2020), no. 3, 171–186.
- [34] N. Tohidi and C. Dadkhah, *Improving the performance of video collaborative filtering recommender systems using optimization algorithm*, *Int. J. Nonlinear Anal. Appl.* **11** (2020), no. 1, 483–495.
- [35] C. Xu, *A big-data oriented recommendation method based on multi-objective optimization*, *Knowledge-Based Syst.* **177** (2019), 11–21.