Information Processing and Management 57 (2020) 102310

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

A collaborative filtering recommender system using genetic algorithm

Bushra Alhijawi*, Yousef Kilani

Hashemite University, Jordan

ARTICLE INFO

Keywords:
Collaborative filtering
Recommender system
Genetic algorithms
Similarity functions
Hybrid recommender system
Semantic information

ABSTRACT

This paper presents a novel genetic-based recommender system (BLI_{GA}) that depends on the semantic information and historical rating data. The main contribution of this research lies in evaluating the possible recommendation lists instead of evaluating items then forming the recommendation list. BLI_{GA} utilizes the genetic algorithm to find the best list of items to the active user. Thus, each individual represents a candidate recommendation list. BLI_{GA} hierarchically evaluates the individuals using three fitness functions. The first function uses semantic information about items to estimates the strength of the semantic similarity between items. The second function estimates the similarity in satisfaction level between users. The third function depends on the predicted ratings to select the best recommendation list.

BLI_{GA} results have been compared against recommendation results from alternative collaborative filtering methods. The results demonstrate the superiority of BLI_{GA} and its capability to achieve more accurate predictions than the alternative methods regardless of the number of K-neighbors.

1. Introduction

Today, e-business websites such that Amazon, eBay, and Alibaba offer a large number of products to their customers. This makes the customers' experience when manually searching for target products harder. The Recommender System (RS) acts as a purchase decision support system that generates recommendation lists for the customers to alleviate this problem and increase the profit of companies (Alhijawi & Kilani, 2016; Lu et al., 2015). The RS techniques have been employed in eight applications, e-government, e-business, e-commerce/e-shopping, e-library, e-learning, e-tourism, e-resource services, and e-group activities (Lu et al., 2015). Those applications distributed into four environments, mobile, cloud, social network and Traditional (PC) environments (Alhijawi & Kilani, 2020).

RS is a filtering system that predicts the probability of a given item to be preferred by a particular user (Ricci, Rokach, & Shapira, 2011; Sharma & Gera, 2013). The recommendation methods can be classified into four types according to how it generates the recommendation: Collaborative Filtering (CF), demographic filtering, content-based filtering, and hybrid systems (Bobadilla, Ortega, Hernando, & Gutiérrez, 2013). The content-based RS builds a user profile by analyzing certain items' characteristics to predict and generate the recommendation (Alhijawi, Obeid, Awajan, & Tedmori, 2018; Chen, Chen, & Wang, 2015). Thus, the recommended items are usually similar to items the user liked in the past. The Demographic-based method assumes that users who have common demographic profiles (e.g. age, gender, and country) will also have the same interests. Thus, this recommendation method predicts different items for different demographic statuses. The CF method has become one of the most popular recommendation methods and

* Corresponding author.
E-mail addresses: bushra.hijawi@gmail.com (B. Alhijawi), ymkilani@hu.edu.jo (Y. Kilani).

https://doi.org/10.1016/j.ipm.2020.102310
Received 17 October 2019; Received in revised form 19 April 2020; Accepted 19 May 2020
0306-4573/

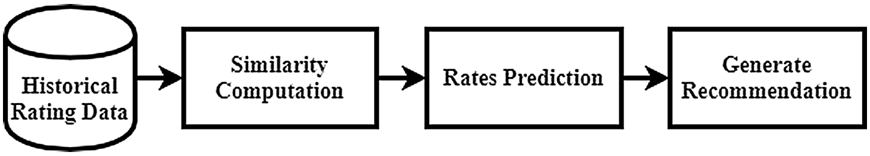

Fig. 1. The main architecture of CF.

achieved remarkable success in terms of accuracy. The principle of CF methods depends on analyzing a historical user's ordinal feedback information to predict the recommendation. In other words, the CF recommends items to a given user that are similar to what similar users preferred regardless of the item's features (Alhijawi et al., 2018). CF applies two different mechanisms to generate the recommendation, memory-based CF and model-based CF (Alhijawi & Kilani, 2020). Memory-based CF uses the historical rating data to locate the *K*-similar users to the Active User (*AU*) (Alhijawi, 2019a; Chen et al., 2015). The historical rating data about the *K*-similar users are utilized to predict the recommendation (Bobadilla et al., 2013). Model-based CF uses the historical rating data to train a model such as Bayesian networks, fuzzy algorithm, clustering models and Genetic Algorithm (GA) to be used in the prediction process of the recommendation (Alhijawi, 2019b; Bobadilla, Ortega, Hernando, & Gutiérrez, 2013). Combining any two or more recommendation techniques results in a hybrid RS (Alhijawi et al., 2018; Gao & Li, 2008; Ho, Fong, & Yan, 2007; Nilashi, Ibrahim, & Bagherifard, 2018; Nilashi, bin Ibrahim, Ithnin, & Sarmin, 2015a).

The recommendation problem is a minimization problem. All CF recommendation approaches aims to achieve as least prediction errors as possible. However, the prediction step depends mainly on the similarity computation step between users/items. Figure 1 shows the main architecture of CF. Two main issues, cold-start, and sparsity affect the quality of the recommendation and prediction accuracy. Those two main issues face the CF because of the full dependency on the historical user-item rating matrix. The sparsity problem is a result of that most users do not rate most of the items or only rate a few items (Alhijawi, Al-Naymat, Obeid, & Awajan, 2019). Thus, the recommendation is generated using a sparse matrix. Cold-start issue faces the CF when there is not enough information available about the AU as he/she is a new user or when a new item is introduced (Alhijawi et al., 2018).

In that context, our interest in this research is the employment of an optimization algorithm in CF to optimize the mentioned objective. A novel genetic-based CF RS called (BLI_{GA}) referring to the Best List of Items using GA is presented. The core idea is to hierarchically evaluate the whole recommendation list to ensure the quality of the entire list instead of selecting individual items to form a list. BLI_{GA} generates the recommendation using the item's semantic features and the user's historical rating data. The selected recommendation list passe three filtering levels to be considered as the Best List of Items (*BLI*) to *AU*. Thus, BLI_{GA} has three fitness functions each one related to a filtering level (i.e criterion). The first filtering level depends on the semantic correlation between items within the list. The second filtering level depends on the satisfaction-based similarity between *AU* and other users who rated at least one item in the list. The third filtering level depends on the predicted ratings of the items in the list. Those filtering levels guaranty that *BLI* has three important characteristics:

1. The items belong to *BLI* have a high correlation in terms of the semantic features.
2. The neighbors of *AU* satisfied with some of the items belong to *BLI*.
3. The items belong to *BLI* have high predicted ratings.

The contribution of this paper includes developing a novel genetic-based CF that aims to select *BLI* that meets *AU*'s interests based on multi-filtering criteria. The GA has been utilized as an optimization tool to find the optimal recommendation list for a specific user since *BLI* is represented as a vector. However, any optimization algorithm can be used instead of GA. The proposed recommendation technique adopts the idea of evaluating the possible recommendation lists instead of evaluating items then forming the recommendation list. The authors expect that BLI_{GA} will alleviate the main challenges face the CF: cold-start and sparsity problems. We adopt a GA with n-fitness functions, each one related to an objective function. Those n-filtering levels are ordered based on the domain problem. Each individual should pass one filtering level to reach another. Also, BLI_{GA} adopts the multi-ratings criteria idea in a new aspect to evaluate (i.e. give rates) the candidate recommendation lists in the population. In other words, each individual has n-fitness values that are considered as ratings gave by the GA to measure the fit degree of this individual. The filtering levels are ordered based on priority as well as on the domain problem. The term "Filtering Level" has been used instead of the term "Objective" since the individuals that score the least fitness ratings in one level will be filtered and removed from the population.

The rest of this paper is organized as follows. In Section 2, the background and related works are covered. Section 3 shows and details the proposed recommendation approach, BLI_{GA}. Section 4 focuses on the testing of BLI_{GA} and discussing the gathered results. Finally, section 5 concludes the paper and presents future work.

2. Literature review

This research deals primarily with GA-based RS, multi-criteria CFRS, and CFRS that uses semantic information. The purpose of this literature review is to present an introduction to the work that already been done in the relevant fields.

2.1. Using genetic algorithms in the recommender system

In RS, the GAs have been utilized in three aspects: clustering (KIM & AHN, 2008; Zhang & you Chang, 2006), hybrid models (Gao & Li, 2008; Ho et al., 2007) and using the GAs without requiring the additional information provided by the hybrid model (Alhijawi & Kilani, 2016; Bobadilla, Ortega, Hernando, & Alcalá, 2011). Usually, the GA used to find the optimal similarity measure. Xiao, Luo, Chen, and Li (2015) proposed an item-based CF system named $ItemCF_{GA}$. This system involves a novel similarity function that uses the average rating for each user instead of the overall average rating for all users. Bobadilla et al. (2011) presented a new RS that employed the GA to find the optimal similarity function. The similarity function that is selected by the GA provides better quality and quicker results than the other traditional metrics. Another GA-based RS called SimGen developed by Alhijawi and Kilani (2016). SimGen computes the user-user similarity matrix using the GA. At the same time, it does not use any of the common similarity measures such as Pearson correlation and vector cosine-based similarity. Each individual in the population represents a user-user similarity matrix that contains similarities values in the range of [0-1].

Gupta, Shivhare, and Sharma (2015) proposed a fuzzy c-means clustering technique and used the GA weighted similarity of Bobadilla et al. (2011). In their system, the generated clusters values of fuzzy c-means are passed to the GA to find the similarity between the clustered values and to obtain optimal similarity measures. KIM and AHN (2008) proposed RS that utilizes the GA to select optimal or sub-optimal initial seeds for the K-means clustering technique. Besides, Wang, Yu, Feng, Wang, 2014 utilized an improved K-means clustering method combined with the GA to build a hybrid model-based movie recommendation system. The GA optimizes the vector of the features' weights to calculate the similarity between users.

Salehi (2014) proposed a latent feature-based recommendation approach that makes improvements in the quality of recommendation and addresses the sparsity problem. They used the GA as a supervised learning task to optimize latent features' weights for each learner based on the historical rating data. A novel hybrid RS for learning materials is developed by Gao and Li (2008) consists of multiple RSs. Their system uses the GA to forecast the performances of each RS. Lv, Hu, and Chen (2016a) developed item-based RS that depends on the items' features with corresponding weights. Kim, Kim, Lee, and Ahn (2010) developed a new music RS that combines the content-based filtering technique with the interactive GA methodologies. The proposed RS has explored the preferences of users. They used the GA to obtain the appropriate music tracks to recommend it to *AU*. Jia, Ding, Liu, Zhang, and Zhang (2014) proposed a CF recommendation algorithm that improved the accuracy of the traditional algorithm. They used the GA to select weights and threshold values quickly to calculate the similarity measure. The proposed algorithm used two measures to generate the recommendation: similarity and trust metric.

Mostly, the GA has been utilized to find the optimal similarity metric or to find the features' optimal weights that lead to the optimal similarity metric (Bobadilla et al., 2011; Gupta et al., 2015; Lv et al., 2016a; Salehi, 2014; Xiao et al., 2015). Also, the GA has been employed to choose the most appropriate initial seeds (K) for the K-mean clustering algorithm (KIM & AHN, 2008). Besides, the GA has been applied to find the optimal user-user similarity matrix (Alhijawi & Kilani, 2016) and to forecasting performances of multiple RSs (Gao & Li, 2008). The recent contributions to the genetic-based RS research area aim to find the appropriate items to form the recommendation list. Table 1 summarizes the reviewed contributions.

2.2. Using multi-criteria and semantic information in the recommender system

The traditional RSs make a recommendation for a user based only on one criterion or one utility function. The common assumption of CF is that two users u_1 and u_2 are similar if both users gave quite similar ratings for particular items. Thus, the RS can predict the unrated items for user u_1 based on the ratings of user u_2 to these items and vice versa. However, user u_1 may give a rating based on particular criteria that differs from the criteria that user u_2 adopted when rating this item. For example, user u_1 and user u_2 both gave the rating of 4 to a book B^*. User u_1 liked B^* because the author $Author_1$ wrote this book while user u_2 enjoined B^* since this book told an action story. Whereas, n-criteria item RSs allows users to give ratings for n semantic features of items. Parveen,

Table 1
Summarization of contributions.

Contribution	Domain	Method	GA Usage	Handled Issues
Alhijawi and Kilani (2016)	Movie	GA	Find the optimal similarity matrix.	-Cold-Start. -Sparsity.
Gao and Li (2008)	Movie	GA	Find the optimal RSs' weights	-Sparsity.
KIM and AHN (2008)	Online diet portal site	k-means Clustering	Select optimal or sub-optimal initial seeds for K-means	-Sparsity.
Bobadilla et al. (2011)	Movie	GA	Find the optimal similarity measure.	-Sparsity.
Xiao et al. (2015)	Movie	GA	Find the optimal similarity measure.	-
Gupta et al. (2015)	Movie	GA, c-means Clustering	Find the similarity between the clustered values.	-Sparsity.
Wang, Yu, Feng, Wang, 2014	Movie	GA, k-means Clustering	Find the optimal features' weights	-Cold-Start. -Sparsity.
Salehi (2014)	e-Learning	GA, Matrix Factorization	Find the optimal latent features' weights for learners	-Sparsity.
Lv et al. (2016a)	Movie	GA	Find the optimal features' weights	-Cold-Start. -Sparsity.
Kim et al. (2010)	Music	GA	Find the optimal music tracks	-Sparsity.
Jia et al. (2014)	Movie	GA	Find the optimal weights and threshold values	-Cold-Start.

Ashraf, and Parveen (2015) designed a fuzzy multi-criteria RS that generates more accurate and precise results. Liu, Mehandjiev, and Xu (2011) developed a novel multi-criteria recommendation technique that clusters users according to their criteria preferences. They presented three different methods for predictions. Jannach, Karakaya, and Gedikli (2012) presented a multi-criteria RS that improves the accuracy using the information derived from multi-dimensional ratings. Their system uses support vector regression to determine the relative importance of the individual criteria ratings. Nilashi et al. (2015) developed a new recommendation method using classification, regression tree and expectation-maximization for accuracy improvement of multi-criteria RSs.

In RS, the information at the semantic level of items includes features (i.e. attribute) and the relationship between items (Nilashi et al., 2018). Recently, the semantic information involves in the RS to alleviate the weaknesses of these systems and to reasoning which item meets the user's needs. Indeed, social experts assume that people tend to buy items related to what they have bought. Many RS applications have been utilized the semantic information such as e-learning (Vesin, Ivanović, Klašnja-Milićević, & Budimac, 2012), movie recommendation (Lv et al., 2016; Nilashi et al.,2018) and tourism recommendation (Lee, Hsia, Hsu, & Lin, 2017). Lv et al. (2016) developed a recommendation technique that combines genetic-based RS with domain ontology to overcome the cold-start problem. Their system generates the recommendation depending on the item's weighted semantic features. The weights of the features are obtained using the genetic algorithm. Nilashi et al. (2018) presented a new hybrid- RS that combines CF, ontology and Singular Value Decomposition (SVD). The domain ontology and SVD are adopted to solve sparsity and scalability problems. Alhijawi et al. (2018) presented a new hybrid RS method depends on the semantic information about the items to alleviate the cold-start and sparsity problems of CF. At the same time, the use of the historical users-item rating matrix helps in alleviating the overspecialization problem of knowledge-based RS. The item's semantic information is exploited to find the users who have the same interests as *AU*. A novel implicit semantic trust-based CF was proposed by Gohari, Haghighi, and Aliee (2016). The proposed RS uses the ant colony optimization technique to find the neighbors of *AU*. The authors used the item's semantic data to compute the semantic similarity between the items. Later, the semantic similarity has been utilized for clustering the items into groups.

Besides, various methods are developed to use both semantic and multi-criteria rating data. Shambour and Lu (2011) proposed a hybrid Multi-Criteria Semantic-enhanced CF algorithm (MC-SeCF). MC-SeCF combines multi-criteria item-based CF with a semantic item-based filtering approach. The semantic filtering approach applies to avoid sparsity and cold-start item problems. A hybrid citation recommendation system was proposed by Zarrinkalam and Kahani (2012) that combines content-based with multi-criteria CF. Table 2 summarizes the reviewed contributions.

2.3. Distinctions

This section presents the differences between the proposed recommendation method (BLI_{GA}) and other recommendation approaches (i.e. genetic-based and multi-criteria-based).

The proposed work employs the GA to find the appropriate recommendation list instead of finding the items to compose the recommendation list. BLI_{GA} utilizes the GA to select the items list that meets the *AU*'s interests. Thus, each individual in the population represents a candidate recommendation list to *AU*. The GA selects one individual to recommend its genes to *AU* depending on hierarchically multi-filtering levels. To our best knowledge, we have not found in the literature any work in the RS research field that focuses on the quality of the whole recommendation list.

Besides, BLI_{GA} adopts the multi-criteria idea in two different aspects:

- BLI_{GA} utilizes the idea of Shambour and Lu (2011) for computing the item-based semantic similarity value of a given target item against other items. The semantic similarity has been computed based on the common semantic features. To our best knowledge, we have not found any work that adopts the idea of item-based semantic similarity in the genetic-based RS.
- BLI_{GA} generates the recommendation based on multi-filtering criteria each one related to a different criterion. Thus, each

Table 2
Summarization of contributions.

Contribution	Domain	Method	Used Data	Handled Issues
Alhijawi et al. (2018)	Movie	Statistics	-Rating. -Semantic features	-Cold-Start. -Sparsity.
Nilashi et al. (2018)	Movie	Clustering	- Semantic features	-Sparsity. -Scalability.
Parveen et al. (2015)	Movie	Fuzzy logic	- Multi-criteria ratings.	-
Liu et al. (2011)	Hotel	Clustering	- Multi-criteria ratings.	-Sparsity.
Jannach et al. (2012)	Hotel, Movie	Support vector regression	- Multi-criteria ratings.	-Sparsity.
Nilashi et al. (2015)	Tourism	Hybrid	- Multi-criteria ratings.	-Sparsity.
Lv et al. (2016)	Movie	Hybrid	- Semantic features	-Cold-Start. -Sparsity.
Gohari et al. (2016)	Movie	Hybrid	-Rating. -Semantic features	-Cold-Start. -Sparsity.
Shambour and Lu (2011)	Movie	Hybrid	- Multi-criteria ratings. - Semantic features	-Cold-Start. -Sparsity.
Zarrinkalam and Kahani (2012)	Citation	Statistics	- Multi-criteria ratings. - Semantic features	-

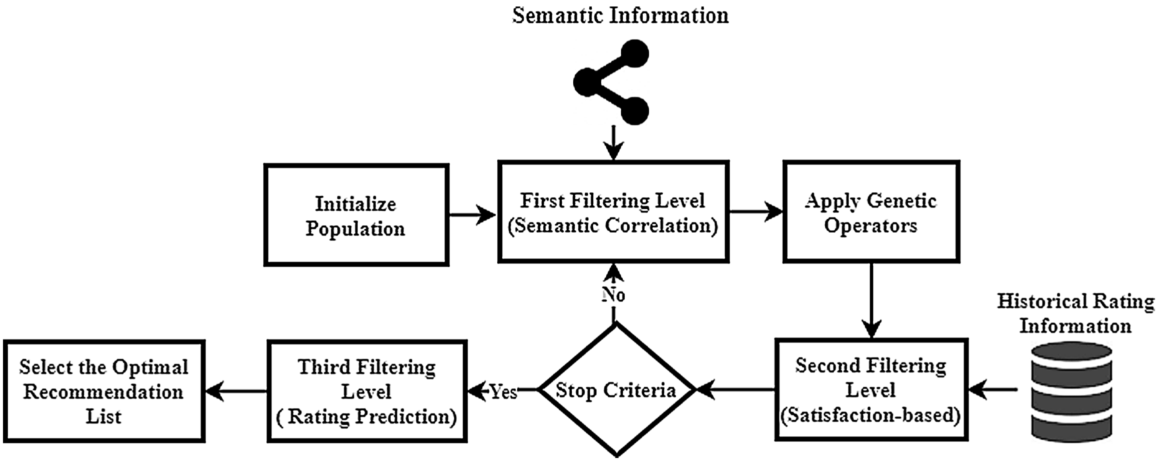

Fig. 2. The main architecture of BLI_{GA}.

individual has multi-fitness values that have been considered as ratings gave by the GA to measure the individual's fit degree. All the reviewed literature works in the genetic-based methods use one filtering technique (i.e. one fitness function) to select the best individual. BLI_{GA} hierarchically filters the individuals to select the best one based on multiple fitness functions.

3. The proposed genetic-based recommender system

BLI_{GA} focuses on the quality of the entire recommendation list instead of selecting individual items to form a recommendation list. The core idea is to hierarchically evaluate the whole recommendation list based on computed multi-criteria. At each evaluation level, the fittest recommendation lists pass to the next evaluation levels. Thus, each evaluation level has been considered as a filtering level. Fig. 2 and Algorithm 1 present the architecture and the main procedure of BLI_{GA}, respectively.

1. **Initialize population (Line 1).** Initially, BLI_{GA} fills the population with M randomly generated recommendation list, *iniPop*. Each individual contains random items that have not previously rated by *AU*.
2. **First filtering level (Lines 3-4).** First, the BLI_{GA} evaluates the individuals based on the semantic correlation criterion to select the top best individuals (*bestMem*) at this step.
3. **Apply genetic operators (Lines 5-7).** From *bestMem*, a set of parents are chosen to produce children by applying crossover and mutation operators. Each produced children will be evaluated in terms of the semantic correlation. All children that have a semantic correlation value greater than the worst individual belongs to *bestMem* are added to the current population.
4. **Second filtering level (Lines 8-9).** Next, the GA evaluates each individual based on the user's satisfaction criterion. The top best individuals (*bestMem*) are captured depending on the satisfaction-based filtering level.
5. **Check the stop criteria (Line 2).** The current generation *bestMem* is transferred to the next generation *nextGen*. BLI_{GA} repeats these steps for a predefined number of generations, *maxGen*.
6. **Third filtering level (Line 11).** In the last generation, the GA evaluates the individuals based on the predicted rating filtering criterion. The predicted rating filtering level is the dominant filtering level. Therefore, the best individual is selected to be *BLI* based on the predicted rating filtering criterion.
7. **Select the optimal recommendation list (Line 12).** BLI_{GA} captures the items of the selected individual and recommends it to *AU*.

Note that the population size mostly decreases from generation to another (Fig. 3). Whereas, the initial population contains M individuals and the next generation has *bestMem* in which $|bestMem| < M$.

Section 3.1 presents details related to the initial population and genetic representation. Section 3.2 illustrates the evaluation process and levels; semantic filtering criteria, satisfaction-based filtering, and predicted rating filtering criteria. Furthermore, in this section, details related to the process of selecting the recommendation list are presented. Section 3.3 shows the genetic operators, termination and replacement strategies.

3.1. Genetic representation and population initialization

BLI_{GA} employs the GA as an optimization tool to select the desired recommendation list for *AU*. Thus, each individual represents a vector of possible items ($\overrightarrow{RecL}$). Each gene (i_g) is a possible item randomly selected from the set of items that are not previously rated by *AU*. BLI_{GA} temporally removes the selected item from the set of items that are not previously rated by *AU* (i.e. while initializing the current individual). This step ensures that the individual does not include duplicate items. The gene value refers to the item's ID.

Input: user-item matrix $U_I(u, i)$ and node item-category matrix $I_C(i, c)$.
Output: Recommendation list.
Generate the initial population (*iniPop*) that contains M individuals.
for $currentGen \neq maxGen$ **do**
    Calculate the semantic correlation ($SemCorrRate(ind)$) of each individual within *iniPop*.
    Select the top (*topX*) best individuals (*bestMem*).
    Parents Selection.
    Apply crossover operator with *CrossOverP* probability.
    Apply mutation operator with *mutP* probability.
    Calculate the satisfaction-based similarity ($Similarity(ind)$).
    Select the top (*topX*) best individuals (*bestMem*).
end
Evaluate each individual within *bestMem* based on the predicted ratings ($predictSatD(ind, x)$).
Select the best individual (*recommendMem*) and recommended its items to *AU*.

ALGORITHM 1. The Main Procedure of BLI_{GA}

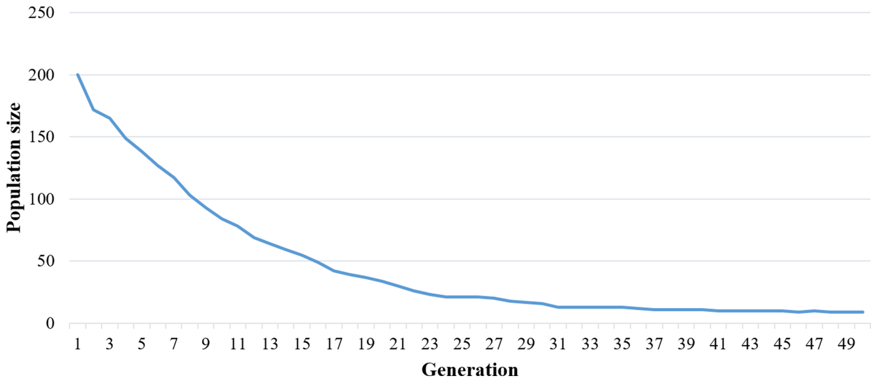

Fig. 3. The population size per generation.

BLI_{GA} initializes a population contains M possible different candidate recommendation list to AU. Formally, the population and individuals are defined as follows:

$$Population = \begin{bmatrix} \overrightarrow{RecL_1} \\ \overrightarrow{RecL_2} \\ \vdots \\ \overrightarrow{RecL_M} \end{bmatrix} = \begin{bmatrix} i_1^1, i_2^1, i_3^1, \ldots, i_N^1 \\ i_1^2, i_2^2, i_3^2, \ldots, i_N^2 \\ \vdots \\ i_1^M, i_2^M, i_3^M. \ldots, i_N^M \end{bmatrix} \quad (1)$$

BLI_{GA} ensures that the population contains a unique recommendation list using a validation mechanism that depends on the summation of genes. The genes of each new produced individual are compared to the individuals which have similar genes summation. For example, assume that the population contains the following four individuals and a new individual $ind_n(I_9, I_3, I_2, I_1)$ was produced. BLI_{GA} will compare the new individual with only ind_1 and ind_3 since the three individuals have the same genes summation of 15. In case the new individual contains at least one unique item then it will be added to the population.

$$\begin{array}{c} ind_1 \\ ind_2 \\ ind_3 \\ ind_4 \end{array} \begin{pmatrix} I_2 & I_7 & I_5 \\ I_{10} & I_5 & I_3 \\ I_2 & I_1 & I_4 \\ I_{10} & I_7 & I_8 \end{pmatrix}$$

3.2. The filtering levels

In BLI_{GA}, the GA hierarchically evaluates the individuals based on three filtering criteria to ensure that the selected recommendation list (*BLI*) meets three main characteristics:

1. The recommended items have common semantic features.
2. The neighbors of *AU* satisfied with some of the recommended items.
3. The predicted ratings of the recommended items are as high as possible.

Each filtering criterion associated with a fitness function. Hence, three fitness functions have been considered in the hierarchical evaluation of BLI_{GA}. The GA give ratings to each individual according to those filtering criteria to select *BLI* of *AU*. The first filtering criterion depends on the item-item semantic correlation ($SemSimI(p, q)$) where $p, q \in ind$. The item-based semantic similarity of a given item against other items is computed based on the common semantic features. The second filtering level measures the harmony degree (i.e. satisfaction-based similarity) between *AU* and other users who gave a rating to at least one item belongs to individual *ind*. The satisfaction-based similarity depends on the user-item rating matrix. The third filtering criterion aims to find the potential favorite items of high predicted ratings to be selected as *BLI*.

3.2.1. The semantic correlation filtering criterion

This filtering level has been chosen motivated by the assumption that people tend to buy items related to what they have bought. Thus, people buy items that have common semantic features. Essentially, *BLI* should contain a set of items with a high semantic correlation degree. BLI_{GA} adopts the proposed idea by Shambour and Lu (2011) for computing the item-based semantic similarity of a given item against other items. Each item represents by a vector of semantic features (Eq. (2)). Each semantic feature represents by a vector of integers ($\overrightarrow{V_c}$). For instance, let V_{c1} represents the genre feature and $V_{c1}^i = (1, 10, 15, 20)$ then this item belongs to the categories: 1, 10, 15 and 20. The Jaccard coefficient (Eq. (3)) has been used to measure the semantic correlation between the items. The semantic correlation value lies in the range [0,1]. Whereas, $SemSimI(p, q) = 1$ indicates that item p and item q belong to the same semantic features. While, $SemSimI(p, q) = 0$ indicates that item p and item q absolutely belong to different semantic features.

$$\vec{V_i} = (V_{c1}, V_{c2}, V_{c3}, \ldots, V_{cn}) \tag{2}$$

$$SemSimI(p, q) = \frac{F_{11}}{F_{10} + F_{01} + F_{11}} \tag{3}$$

where,

- F_{11} refers to the total number of features in which both items, p and q, belong to it. Formally,

$$F_{11} = \sum_{f=1}^{n} |V_{cf}^{p} \wedge V_{cf}^{q}| \tag{4}$$

- F_{10} refers to the total number of features in which p belongs to it and q does not belong to it. Formally,

$$F_{10} = \sum_{f=1}^{n} |V_{cf}^{p} - V_{cf}^{q}| \tag{5}$$

- F_{01} refers to the total number of features in which q belongs to it and p does not belong to it. Formally,

$$F_{01} = \sum_{f=1}^{n} |V_{cf}^{q} - V_{cf}^{p}| \tag{6}$$

BLI_{GA} aggregates the item-item semantic similarity between every two items belong to the individual *ind* to compute the first fitness value (i.e. semantic-based rating (Eq. (7))) of *ind*. The higher the semantic similarity degree between items, the higher the semantic-based rating of the individual. BLI_{GA} computes the semantic correlation between items in offline mode to reduce the needed time for these computations. While in online processing, only the individual's fitness value is calculated. The following example details the process.

$$SemCorrRating(ind) = \sum_{p,q \in ind} SemSimI(p, q) \tag{7}$$

Example 3.1. In this example, two movies and seven genres (action, comedy, adventure, Sci-Fi, crime, horror, war) and three actors (i.e. Act1, Act2, Act3) have been considered. For simplicity, only two semantic feature (i.e. genre, actor) is used in computation. The vectors of both items are defined as follows:

$$\begin{aligned} V_M &= (Genre, Actor) \\ V_A &= (\{1, 0, 1, 1, 0, 0, 0\}, \{1, 1, 0\}) \\ V_B &= (\{0, 0, 1, 1, 1, 0, 1\}, \{0, 1, 0\}) \\ V_C &= (\{0, 1, 0, 1, 0, 1, 1\}, \{0, 1, 1\}) \end{aligned}$$

Whereas, the value of 1 indicates that the movie belongs to this genre. For example, movie A belongs to action, adventure and Sci-Fi genres, thus the first, third and fourth values in the vector V_A have been set to 1.

The semantic correlation, $SemSimI(A, B)$, between the two movies is given by the formula:

$SemSimI(A, B) = \frac{F_{11}}{F_{10} + F_{01} + F_{11}} =$
$\frac{|\{Adventure, Sci-Fi, Act2\}|}{|\{Action, Act1\}| + |\{Crime, War\}| + |\{Adventure, Sci-Fi, Act2\}|} = \frac{3}{2+2+3} = 0.429$

$SemSimI(A, C) = \frac{2}{3+4+2} = 0.223$

$SemSimI(B, C) = \frac{3}{2+3+3} = 0.375$

Assume the ind_1 contains the movies *A, B, C*. The semantic correlation rating of ind_1 is computed as follows:

$SemCorrRating(ind_1) = SemSimI(A, B) + SemSimI(A, C) + SemSimI(B, C)$
$= 0.429 + 0.223 + 0.375 = 1.027$

3.2.2. The satisfaction-based filtering criterion

The satisfaction-based filtering level has been chosen motivated by the assumption that people who liked the same items are likely to feel similarly towards other items. Thus, BLI_{GA} searches for a list contains high semantically correlated items in which the like-minded users as *AU* satisfied with some of those items.

The satisfaction-based similarity is a user-based similarity measure the degree of harmony between *AU* and other users who gave ratings on at least one item belong to the individual *ind*. For computation purposes, each user is represented as a vector of integers (Eq. (8)) in which the value of v_{ik}^{u} refers to the given rating by user u on item *ik*. BLI_{GA} computes the satisfaction-based rating (Eq. (9))

using a linear combination that consists of Pearson correlation measure (Eq. (10)) and binary Jaccard (Eq. (12)). The Pearson correlation measure depends on the commonly rated items between two users regardless of the number of given ratings by each user. Consequently, this may contribute to computing an inequitable and unreliable high similarity between two users in case they have a small proportion of commonly rated items. For instance, suppose user u_a gave ratings on a total of 266 items, user u_b gave ratings on a total of 32 items and both users gave those 32 items the same ratings. Then, it is unreliable to consider that user u_a is 100% similar to user u_b since the percentage of commonly rated items between both users is 12.1%. Thus, the *Psim*((*AU, u*) is tuned by the binary Jaccard coefficient to consider the proportion of commonly rated items between users (i.e. $v_{ik}^{ux} > 0 \wedge v_{ik}^{uz} > 0$). Note that BLI_{GA} computes both Pearson correlation and Jaccard coefficient values in offline mode and only the individual's fitness value is computed in the online processing.

$$\overrightarrow{V_u} = (v_{i1}^{u}, v_{i2}^{u}, \cdots, v_{ig}^{u}) \tag{8}$$

$$SatRating(ind) = \sum_{u \in U} SatSimU(AU, u) \tag{9}$$

$$SatSimU(AU, u) = Psim(AU, u)^{*}Jaccard(AU, u) \tag{10}$$

$$Psim(AU, u) = \frac{\sum_{i \in I} (r_i^{AU} - \bar{r_{AU}})(r_i^{u} - \bar{r_u})}{\sqrt{\sum_{i \in I} (r_i^{AU} - \bar{r_{AU}})^2}\sqrt{\sum_{i \in I} (r_i^{u} - \bar{r_u})^2}} \tag{11}$$

$$Jaccard(AU, u) = \frac{|\overrightarrow{V_{AU}} \cap \overrightarrow{V_u}|}{|\overrightarrow{V_{AU}} \cup \overrightarrow{V_u}|} \tag{12}$$

where,

- I is the group of rated items by both users AU and u.
- $r_{AU,i}$ is the rate of user AU on item i.
- $\bar{r_{AU}}$ is the mean rating value of user AU.
- $r_{u,i}$ is the rating of user u on item i.
- $\bar{r_u}$ is the mean rating value of user u.
- U refers to the set of users who gave ratings on at least one item belong to *ind*.

3.2.3. Selecting the best list of items

In the last generation, the population contains a number of the best individuals. Each individual consists of items that satisfy two conditions:

1. Semantically correlated.
2. The *AU*'s neighbors gave ratings on some of those items.

However, both conditions do not guarantee that *AU* may prefer those items. Therefore, BLI_{GA} predicts the *AU*'s satisfaction degree with each candidate recommendation list. The Resnick's adjusted weighted sum (Eq. (14)) has been utilized to predict the ratings of the items. BLI_{GA} utilizes the predicted ratings to compute the prediction-based rating, third fitness value, (Eq. (13)) of individual *ind*.

$$predictSatRatng(ind, AU) = \sum_{i \in ind} P_{AU,i} \tag{13}$$

$$P_{AU,i} = \bar{r_{AU}} + \frac{\sum_{u \in U^+} (r_i^{u} - \bar{r_u})^{*}Psim(AU, u)}{\sum_{u \in U^+} Psim(AU, u)}, \tag{14}$$

BLI_{GA} considered the Top-*k* similar users to *AU* (i.e. U^+) who gave ratings on at least one item belong to the individual in the prediction process. Thus, for each candidate recommendation list, a different set of neighbors is utilized in the prediction step.

BLI refers to the individual with the highest predicted-based rating. BLI_{GA} faces a challenge when multiple individuals have the same highest prediction-based rating. To handle such a challenge, BLI_{GA} returns the individual in which the satisfaction-based rating is the highest. In case multiple individuals have the same highest satisfaction-based rating, BLI_{GA} returns the individual which has the highest semantic-based rating. In case multiple individuals have the same highest semantic-based rating, BLI_{GA} randomly selects one individual and returns it. Example 3.2 illustrates the process of selecting *BLI*:

Example 3.2. Suppose that:

- There are 10 items: I_1, I_2, $\cdots$, I_{10} and 4 users: u_1, u_2, u_3, u_4. Fig. 4a presents the items and users by columns and rows, respectively. The dot in Fig. 4a indicates that the user did not give rating on this item. For instance, u_1 gave ratings on items: I_1 (3), I_4 (4), and I_6 (1) and did not give rating on items: I_2, I_3, I_5, I_7, I_8, I_9, and I_{10}. In this example, each user gave ratings on at least 3 items.
- *AU* is u_1. Thus, the recommendation list is a subset of { I_2, I_3, I_5, I_7, I_8, I_9, I_{10} } (i.e. the items that are not previously rated by u_1).

$$\begin{pmatrix} u_1: & 3 & . & . & 4 & . & 1 & . & . & . & . \\ u_2: & 2 & . & 4 & . & 3 & 2 & . & 4 & 5 & . \\ u_3: & . & 1 & . & 1 & . & 3 & 1 & 3 & . & 2 \\ u_4: & 4 & . & 2 & . & 5 & . & 4 & . & 3 & 5 \end{pmatrix}$$

(a) $user - item$ matrix

$$\begin{array}{c} ind_1 \\ ind_2 \\ ind_3 \\ ind_4 \end{array} \begin{pmatrix} I_2 & I_7 & I_5 \\ I_{10} & I_5 & I_3 \\ I_3 & I_8 & I_9 \\ I_{10} & I_7 & I_8 \end{pmatrix}$$

(b) Individuals

Fig. 4. User-Item rating matrix and the Population of Example 3.2.

- In the last generation of GA, the population contains 4 individuals (Fig. 4b).

The prediction-based rating of each individual is computed using Eq. (13) to choose the recommendation list (*BLI*). The following steps detail the process. Then, based on the predicted ratings, the predicted satisfaction degree of each individual is computed.

1. The prediction rating of items process requires that the satisfaction-based similarity between *AU* and other users is computed. BLI_{GA} adopts the Pearson correlation measure for this purpose.

$$\begin{aligned} Psim(u_1, u_2) &= 0.55 \\ Psim(u_1, u_3) &= -0.99 \\ Psim(u_1, u_4) &= 0.99 \end{aligned}$$

2. Compute the predicted rating using Eq. (14).

$$\begin{array}{ll} P_{u_1,I_2} = 3.5 & P_{u_1,I_3} = 1.72 \\ P_{u_1,I_5} = 3.3 & P_{u_1,I_7} = 3.16 \\ P_{u_1,I_8} = 2.16 & P_{u_1,I_9} = 1.55 \\ P_{u_1,I_{10}} = 3.17 & \end{array}$$

3. Compute the prediction-based rating using Eq. (13).

$$\begin{aligned} predictSatRatng(ind_1, u_1) &= 9.96. \\ predictSatRatng(ind_2, u_1) &= 8.18. \\ predictSatRatng(ind_3, u_1) &= 5.43. \\ predictSatRatng(ind_4, u_1) &= 8.49. \end{aligned}$$

For instance, the prediction-based rating of ind_1 for u_1 is 9.96 which is given by:$predictSatRatng(ind_1, u_1) = P_{u_1,I_2} + P_{u_1,I_7} + P_{u_1,I_5} = 3.5 + 3.16 + 3.3$

Thus, the items of ind_1 (i.e. I_2, I_7, I_5) are recommended to *AU* (i.e. u_1) since it has the highest prediction-based rating. ⋄

3.3. The genetic operators and termination criteria

BLI_{GA} applies the common genetic operators: selection, crossover, and mutation. The GA terminates when reaching the maximum number of generations (*maxGen*). The features of used genetic operators are:

- Selection. The roulette-wheel selection method (RW) has been used to select the parents for mating (*ParentPool*). RW gives high selection probability for the fitter individuals and low selection probability for bad individuals. However, both good and bad individuals have chances to be selected but the good individuals have higher probability.

```
Input: An individual ind
Output: Boolean decision
key = ind[0]
for (gene1 = 1; gene1 < |ind|; gene1 + +) do
    for (gene2 = gene1; gene2 < |ind|; gene2 + +) do
        if key == ind[gene2] then
            return True
        end
    end
    key = ind[gene1]
end
return False
```

ALGORITHM 2. BLI_{GA}'s Duplication-Check-Crossover Procedure

- Crossover. One-point crossover technique has been applied to generate two new children after each round of crossover. The crossover probability (*CrossOverP*) is experimentally tuned. Whereas, a random number between [1, (0.8 × |*ParentPool*|] is selected to determine the number of corssover operations per generation. The crossover technique is fed with two individuals in which; (1) the first individual has *CrossOverP* probability to be selected form *ParentPool*, (2) the second individual is randomly selected from *ParentPool*. BLI_{GA} utilizes Algorithm 2 to ensure that the individual does not include duplicate items. The individual with duplicate items has been excluded.
- Mutation. BLI_{GA} utilizes the Uniform mutation technique. This type of mutation can only be used for integer and float genes. Simply, the child is produced by replacing the value of the chosen gene from the parent with a uniform random value selected between specified upper and lower bounds for that gene. The new gene's value is randomly selected from a pool of items that contains the items that are not previously seen by *AU* and not included in the parent individual. This step ensures that the individual does not include duplicate items. The mutation probability (*mutP*) is experimentally tuned. Each individual in the *ParentPool* has *mutP* probability to be selected as an input for the mutation operation.

The best individuals of the parents and children (*bestMem*) at the current generation are picked out to prepare the population of the next generation. Because of the applied hierarchically filtering method, the number of individuals in the population is not fixed from generation to another.

4. Evaluation and results

This section provides detailed of how the proposed RS, BLI_{GA}, was tested. Various experiments were conducted for purposes of comparing the proposed RS with alternative recommendation techniques:

- Pearson-based CF (PRC). The Pearson-based CF depends mainly on Pearson correlation similarity metric which is one of the component of BLI_{GA}.
- Cosine-based CF (COS). This recommendation technique generate the recommendation based on the cosine similarity metric.
- Bobadilla et al.'s (2011) (GA1). GA1 is a genetic-based CF method the uses the GA to find the optimal similarity function. The selected function is used in recommendation generation process.
- Xiao et al.'s (2015) (ItemCFGA). ItemCFGA is an item-based CF depends on a novel similarity function that is selected by the GA.
- Alhijawi and Kilani (2016) (SimGen). SimGen is a genetic-based CF that uses GA to find the optimal user-user similarity matrix. The selected similarity matrix is used in the recommendation generation process.
- Shambour and Lu (2011) (Hybrid MC-SeCF). Hybrid MC-SeCF is a hybrid recommendation method that combines item-based CF with a semantic item-based filtering approach. BLI_{GA} adopts the item's semantic similarity used by Shambour and Lu (2011).

Details related to the datasets that have been used in the experiments is provided in section 4.1. Section 4.2 provides detailed related to the followed experimental methodology and the used evaluation metrics. Finally, Section 4.3 presents and discusses the results.

4.1. Experimental data

Two benchmark datasets and synthetic data have been used in all experiments. Table 3 shows the statistics of the experimental data.

- Movielens 100K dataset. This dataset consists of 100,000 ratings (1-5) collected from 943 users on 1682 movies. Each user has rated at least 20 movies and each movie belongs to at least one category from the 18 categories. The sparsity level of this dataset is 93.7% (sparsity level $=1-(100000/(943*1682))=0.937$).
- HetRec 2011 (MovieLens + IMDb/Rotten Tomatoes) dataset. This dataset consists of 855598 rating (1-5) collected from 2113 users on 10197 movies, 95321 actors, 4060 directors and 20 genres. It is an extension of the MovieLens10M dataset, published by GroupLeans research group. Each user gave ratings at least to 20 items. The sparsity level of this dataset is 96.1%(sparsity level =

Table 3
Specifications of the used datasets.

	Movielens	HetRec 2011	Synthetic data
Number of users	943	1682	300
Number of movies	2113	10197	300
Number of genres	18	20	18
Number of actors	0	95321	0
Number of directors	0	4060	0
Number of ratings	100000	855598	3940
Rating scale	1-5	1-5	1-5
Sparsity level	93.7%	96.1%	95.6%

$1 - (855598/(2113*10197)) = 0.961$).

- Synthetic data (SD). We built a random biased data by applying the Marmanis and Babenko (2009) approach. This data consists of 3940 rating (1-5) made by 300 users, $u_1, u_2, \cdots, u_{300}$ on 300 item, $i_1, i_2, \cdots, i_{300}$. Each item belongs to at least one category from the 18 categories generated randomly. The sparsity level of this dataset is 95.6% (sparsity level $=1 - (3940/(300*300)) = 0.956$). This dataset is considered as a small one. The goal of building SD is to simulate the case when the system is new and the available information is small size and sparse. The users and items in SD are split into groups as follows:
  - $U_{40} = \{u_1, \cdots, u_{120}\}$, contains 40% of the total users.
  - $U_{60} = \{u_{41}, \cdots, u_{300}\}$, contains the remaining 60% of the users.
  - $I_{40} = \{I_1, \cdots, I_{120}\}$, contains 40% of the total items.
  - $I_{60} = \{I_{41}, \cdots, I_{300}\}$, contains the remaining 60% of the items.

  Six constraints control the building process to facilitate the validation process. These constraints have been summarized below. The result of running any algorithm using the synthetic random data determines whether the RS generates a high-quality recommendation. For example, the recommended items for the users belong to U_{40} should belong to I_{40} set.
  1. Each user has given ratings on n movies, where n is a random number between 10 and 20.
  2. Each movie belongs to m category from the 18 categories, where m is a random number between 1 and 4.
  3. n items from I_{40} randomly assigned to each user from U_{40} with a random rating of 4 or 5.
  4. n items from I_{60} randomly assigned to each user from U_{60} with a random rating of 1, 2 or 3.
  5. Each item of I_{40} was randomly added to m categories of the first 7 categories.
  6. Each item of I_{60} was randomly added to m categories of the remaining 11 categories.

4.2. Experiments design

The BLI_{GA}'s evaluation process consists of four experiments that were conducted under 2 GB RAM Dual CPU 2.16 GHz PC. The first experiment is a special experiment aims to evaluate the selected recommendation list by BLI_{GA}. In this experiment, we consider the obtained results using the Movielens and HetRec 2011 datasets. Five statistical values have been captured and retained for the selected recommendation list (i.e. *BLI*). Those values are mentioned below:

1. The minimum predicted item's rating in the selected recommendation list.
2. The maximum predicted item's rating in the selected recommendation list.
3. The average predicted item's rating in the selected recommendation list.
4. The percentage of the recommended hidden items ratio for all recommended items.

The second experiment aims to compare the running time performance of BLI_{GA} with other genetic-based RS, specifically GA1 (Bobadilla et al., 2011), ItemCFGA (Xiao et al., 2015) and SimGen (Alhijawi & Kilani, 2016). The needed CPU time by BLI_{GA} to predict the recommendation is compared with the needed CPU time by other techniques. The average needed time to produce recommendation for a user has been considered in the comparisons The third experiment examines the prediction accuracy of BLI_{GA} and compared it with other baseline methods. The Mean Absolute Error (MAE) has been utilized to measure the absolute difference between the predicted ratings and the real ratings. Formally, the MAE is computed using Eq. (15). The number of selected neighbors (k) has a significant impact on MAE results. Thus, the MAE is computed using a variable number of neighbors.

$$MAE = \frac{1}{\#U} \sum_{u=1}^{U} \frac{\sum_{i=1}^{I_u} |p_{u,i} - r_{u,i}|}{\#I_u} \tag{15}$$

The fourth experiment evaluates the BLI_{GA} in terms of recommendation quality using recall, precision, and F1-measure. The recall (Eq. (16)) is the percentage of the favorite recommended items to the total favorite items of *AU*. The precision (Eq. (17)) is the fraction of the interesting recommended items to the total number of recommended items. F1-measure (Eq. (18)) is the harmonic mean of precision and recall. Both recall and precision measures are sensitive to the number of recommended items (*N*). Thus, the recommendation quality measures have calculated using a different number of recommended items.

$$Recall = \frac{TP}{TP + FN} \tag{16}$$

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$F1 - measure = \frac{2^* \ Precision \ ^* \ Recall}{Precision \ + \ Recall} \tag{18}$$

Table 4 presents the main parameters used in the experiments. 20% of the users were randomly selected as testing users. For testing purposes, 40% of the rated items by each of the testing users have been hidden. This will help in examining the effect of the cold-start problem on the RSs performance. The percentage of the achieved improvements by BLI_{GA} is computed using Eq. (19).

$$Improvement = \frac{Result\ of\ Algorithm\ A - Result\ of\ BLI_{GA}}{Result\ of\ Algorithm\ A} \tag{19}$$

Table 4
Main parameters used in the experiments.

	k	N	Test users %	Runs
Movielens	20-200	2-20	20%	51
HetRec 2011	20-200	2-20	20%	51
Synthetic data	20-20	2-20	20%	51

A sensitivity analysis has been conducted to determine the suitable values that lead to the best GA performance. Table 5 presents the selected values of the parameters.

4.3. Results

This section exhibits the obtained results of all experiments are presented and discussed. Section 4.3.1 demonstrates the gathered statistics of the selected recommendation list. Section 4.3.2 presents the needed CPU time to generate the recommendation by genetic-based methods. Section 4.3.3 presents the prediction accuracy results of all recommendation methods. Section 4.3.4 shows the gathered recommendation quality results of the recommendation techniques.

4.3.1. Recommendation list evaluation: a special experiment

The main goal is to examine the selected recommendation list by BLI_{GA}. Mainly, this experiment focuses on the quality of the recommended items (*BLI*). Four statistical values are captured for each selected recommendation list for each testing user:

1. The minimum predicted item's rating in the selected recommendation list.
2. The maximum predicted item's rating in the selected recommendation list.
3. The average predicted item's rating in the selected recommendation list.
4. The percentage of the recommended hidden items ratio for all recommended items.

Table 6 illustrates the results of this experiment. The predicted ratings of recommended items range between [3.5,5] and [3.3,5] when using Movielens and HetRec 2011, respectively. Note that 40% of the rated items by *AU* have been hidden. The authors expect that BLI_{GA} will recommend to *AU* a set of items that satisfy two constraints:

1. The recommended items should belong to the hidden items set.
2. The recommended items should have at least a rating of 3.

In general, the results indicate that BLI_{GA} is able to recommend 7 out of 10 items that are truly favorite and belong to the hidden items set. On average, when uses the Movielens dataset, 69.5% of the recommended items satisfy the constraints. While the gathered results when using HetRec 2011 dataset show that 70.1% of the selected recommendation items by BLI_{GA} satisfy the constraints.

4.3.2. Running-time performance

The objective of the second experiment is to compare BLI_{GA} with the alternative genetic-based RS techniques in terms of required for generating the recommendation. Table 7 presents the average required CPU time in seconds to generate the recommendations by the four techniques using the experimental datasets. BLI_{GA} needed 12.3 s using Movielens to generate a recommendation list while GA1, ItemCFGA, and SimGen needed 15, 21.2 and 14.9 s, respectively. By meaning, BLI_{GA} is 18%, 41.9%, and 17.5% faster than GA1, ItemCFGA and SimGen, respectively. In the case of HetRec 2011, SimGen outperforms BLI_{GA} by 15% this is because the number of users and items has more impact on the performance of BLI_{GA} than SimGen. However, BLI_{GA} generates the recommendations 3.2% and 26.4% faster than GA1 and ItemCFGA, respectively. The needed time to predict the recommendation using SD by BLI_{GA} is 29.4%, 50.7%, and 12.2% lower than the needed time by GA1, ItemCFGA, and SimGen, respectively.

4.3.3. Prediction accuracy results

The third experiment examines the performance of BLI_{GA} in terms of prediction accuracy. The BLI_{GA}'s prediction accuracy is

Table 5
The used GA parameters in the experiments.

Parameter	Value
Number of generations	50
Population size	200
Mutation probability	20%
Crossover probability	60%
Top best individual percentage	75%

Table 6
The statistics of BLI_{GA}.

Dataset	Average predicted rating	Minimum predicted rating	Maximum predicted rating	Average Percentage of recommended hidden items
Movielens	3.9	3.5	5	69.5%
HetRec 2011	3.7	3.3	5	70.1%

Table 7
The needed CPU time that to generate the recommendation in seconds.

Dataset		GA1	ItemCFGA	SimGen	BLI_{GA}
Movielens	Time(second)	15	21.2	14.9	12.3
	Improvments	18%	41.9%	17.5%	
HetRec 2011	Time(second)	19	25	16	18.4
	Improvments	3.2%	26.4%	-15%	
SD	Time(second)	5.1	7.3	4.1	3.6
	Improvments	29.4%	50.7%	12.2%	

compared with the one obtained by the alternative methods; PRC, COS, GA1, ItemCFGA, SimGen, and Hybrid MC-SeCF. Table 8 shows the obtained prediction accuracy results by applying all recommendation methods using all datasets.

BLI_{GA} achieves significantly fewer errors than other methods for any selected number of neighbors using all datasets. The reader can observe that BLI_{GA}'s and Hybrid MC-SeCF's accuracy when using Movielens and HetRec 2011 are close when the number of the selected neighbor is below 60. However, BLI_{GA} achieved better performance than Hybrid MC-SeCF when the number of the selected neighbor is increased. In general, there is a positive correlation between the accuracy improvement and the number of neighbors. The obtained accuracy by applying BLI_{GA} when using SD is worst than the one obtained using other datasets. Note that SD is a small and sparse dataset. However, the prediction accuracy of BLI_{GA} is improved in the whole k values. On average, BLI_{GA} outperforms other recommendation methods by 15.4%, 13.6% and 13.3% using Movielens, HetRec 2011 and SD.

4.3.4. Recommendation quality results

The objective of the fourth experiment is to evaluate the performance of BLI_{GA} in terms of recommendation quality. Therefore, the recommendation quality of BLI_{GA} is compared against the recommendation quality of the other alternative recommendation approaches; PRC, COS, GA1, ItemCFGA, SimGen, and Hybrid MC-SeCF.

Table 9 presents a comparison between RSs based on recall results using the datasets. From this table, the authors can conclude that the performance of all recommendation methods with the increasing of the number of Top-N recommendation is improved. This is because the recall represents the percentage of the favorite recommended items to all favorite items in the collection. Thus, increasing the number of recommended items leads to an increase in the probability of recommending interesting items for AU. Clearly, the recall results of BLI_{GA} and Hybrid MC-SeCF are close when $N = 4$ and $N \geq 18$. However, on average, BLI_{GA} outperforms Hybrid MC-SeCF by 3.21%. The percentage of the achieved improvement by BLI_{GA} in terms of recall using Movielens is 44.4%. With respect to gathered results using HetRec 2011, BLI_{GA} outperforms other methods by 43.6%. You can observe that the performance of Hybrid MC-SeCF is close to the performance of BLI_{GA} when $N < 10$. While the recall results of BLI_{GA} when using SD is close to Hybrid MC-SeCF and SimGen. Moreover, the recommendation quality of all methods is less than the one achieved when using the Movielens and HetRec 2011 datasets. Remember that SD is a small and sparse dataset that represents the case when the system is new. Thus, the RS faces a challenge in generating a high-quality recommendation. Anywise, BLI_{GA} made a significant improvement when comparing it with other recommendation methods. On average, applying BLI_{GA} using SD improves the recall results by 46.98%.

Table 10 shows the gathered precision results. The reader can observe that BLI_{GA} has remarkable improvements when compared with other recommendation approaches regardless of the used dataset. The results illustrate the negative correlation between the precision of all methods and the number of the recommended items. This results of the positive correlation between the probability of recommending uninteresting items and the number of recommended items. The precision of BLI_{GA} has reached the best case at $Top - 2$ then begins to decrease. However, there is a big gap in performance between BLI_{GA} and other techniques when using Movielens and HetRec 2011. The performance of all techniques using SD was decreased when compared to their performance using other datasets. BLI_{GA} achieved precision results close to the ones of Hybrid MC-SeCF. However, BLI_{GA}'s precision results exceed Hybrid MC-SeCF's precision results by 8.1%. On average, BLI_{GA} made 71.2% improvements in terms of precision.

Table 11 presents the F1-measure results of the recommendation methods. The obtained results demonstrate the superiority of the proposed method. On average, BLI_{GA} achieved 76% better F1-measure results than other baseline recommendation methods. Note that F1-measure takes both false positives and false negatives into account.

4.4. Discussion

The main idea that has been adopted in this research is to evaluate the possible recommendation lists instead of evaluating items then forming the recommendation list. Thus, BLI_{GA} searches for the recommendation list that meets three main important features:

Table 8
Prediction accuracy of the recommendation methods.

Dataset	Recommendation method	K			
		20	40	60	80
Movielens	COS	0.9025	0.8901	0.8805	0.8687
	PRC	0.8732	0.8578	0.8523	0.8549
	Hybrid MC-SeCF	**0.719**	**0.712**	0.7	0.7115
	SimGen	0.7513 ± 0.04	0.75 ± 0.041	0.7401 ± 0.031	0.7325 ± 0.041
	GA1	0.812 ± 0.04	0.785 ± 0.042	0.77 ± 0.04	0.768 ± 0.041
	ItemCFGA	0.815 ± 0.051	0.8 ± 0.048	0.789 ± 0.05	0.778 ± 0.05
	BLI_{GA}	0.726 ± 0.049	0.713 ± 0.05	**0.689** ± 0.048	**0.6723** ± 0.048
HetRec 2011	COS	0.885	0.88	0.8596	0.8495
	PRC	0.852	0.8456	0.84	0.8339
	Hybrid MC-SeCF	0.715	0.711	0.7	0.7056
	SimGen	0.764 ± 0.035	0.754 ± 0.038	0.7315 ± 0.0349	0.7174 ± 0.039
	GA1	0.7789 ± 0.008	0.7685 ± 0.0091	0.7542 ± 0.008	0.7425 ± 0.0085
	ItemCFGA	0.8 ± 0.041	0.798 ± 0.04	0.78512 ± 0.045	0.7795 ± 0.039
	BLI_{GA}	**0.703** ± 0.051	**0.7** ± 0.045	**0.6942** ± 0.0482	**0.672** ± 0.048
SD	COS	0.915	0.901	0.9	0.891
	PRC	0.89	0.88	0.859	0.8454
	Hybrid MC-SeCF	0.72	0.723	0.716	0.712
	SimGen	0.74 ± 0.039	0.735 ± 0.042	0.718 ± 0.039	0.709 ± 0.039
	GA1	0.768 ± 0.043	0.76 ± 0.0425	0.7523 ± 0.04	0.7444 ± 0.04
	ItemCFGA	0.8 ± 0.051	0.801 ± 0.051	0.793 ± 0.05	0.781 ± 0.045
	BLI_{GA}	**0.71** ± 0.04	**0.698** ± 0.039	**0.678** ± 0.04	**0.66** ± 0.04

Dataset	K					
	100	120	140	160	180	200
Movielens	0.86	0.8502	0.8432	0.843	0.831	0.83
	0.8361	0.8201	0.81	0.7856	0.7729	0.77
	0.719	0.717	0.7106	0.71	0.7164	0.71
	0.683 ± 0.042	0.681 ± 0.051	0.667 ± 0.05	0.655 ± 0.044	0.635 ± 0.043	0.629 ± 0.045
	0.75 ± 0.041	0.741 ± 0.04	0.71 ± 0.039	0.69 ± 0.04	0.68 ± 0.0395	0.67 ± 0.04
	0.76 ± 0.0496	0.751 ± 0.0489	0.75 ± 0.049	0.749 ± 0.05	0.751 ± 0.048	0.747 ± 0.049
	0.6529 ± 0.0485	**0.6344** ± 0.049	**0.6159** ± 0.05	**0.5982** ± 0.051	**0.5789** ± 0.0495	**0.556** ± 0.049
HetRec 2011	0.8368	0.8241	0.8114	0.79867	0.78597	0.7733
	0.815	0.8	0.7953	0.7751	0.7588	0.80346
	0.6981	0.7	0.703	0.69612	0.693975	0.69184
	0.7011 ± 0.04	0.6985 ± 0.0355	0.6777 ± 0.041	0.6634 ± 0.039	0.6491 ± 0.041	0.6349 ± 0.036
	0.73015 ± 0.008	0.7178 ± 0.009	0.70545 ± 0.0075	0.6931 ± 0.008	0.68075 ± 0.009	0.6684 ± 0.008
	0.77206 ± 0.04	0.76462 ± 0.04	0.7572 ± 0.039	0.7498 ± 0.0421	0.7423 ± 0.0425	0.7349 ± 0.038
	0.6576 ± 0.032	**0.6378** ± 0.035	**0.6299** ± 0.036	**0.6163** ± 0.0385	**0.6027** ± 0.0386	**0.5891** ± 0.04

(*continued on next page*)

Table 8 (*continued*)

Dataset	K					
	100	120	140	160	180	200
SD	0.869	0.86	0.851	0.84	0.829	0.8085
	0.82984	0.815	0.7989	0.784	0.7679	0.7524
	0.71	0.71	0.7	0.705	0.703	0.7
	0.698 ± 0.039	0.68 ± 0.041	0.676 ± 0.039	0.665 ± 0.038	0.652 ± 0.039	0.633 ± 0.039
	0.73655 ± 0.04	0.7287 ± 0.042	0.721 ± 0.039	0.7 ± 0.0425	0.6911 ± 0.041	0.6873 ± 0.04
	0.77 ± 0.0455	0.765 ± 0.048	0.757 ± 0.0385	0.749 ± 0.04	0.741 ± 0.045	0.733 ± 0.042
	0.654 ± 0.0385	**0.65** ± 0.038	**0.6447** ± 0.039	**0.6397** ± 0.04	**0.6347** ± 0.035	**0.62** ± 0.036

Table 9
Recall results of the recommendation methods.

Dataset	Recommendation method	N									
		Top-2	Top-4	Top-6	Top-8	Top-10	Top-12	Top-14	Top-16	Top-18	Top-20
Movielens	COS	0.232	0.275	0.29	0.301	0.323	0.35	0.3697	0.391	0.413	0.494
	PRC	0.275	0.342	0.368	0.4	0.413	0.441	0.46	0.47	0.516	0.557
	SimGen	0.43	0.527	0.57	0.612	0.631	0.665	0.67	0.7	0.71	0.751
	GA1	0.31	0.422	0.468	0.5	0.54	0.574	0.609	0.665	0.7	0.74
	ItemCFGA	0.295	0.331	0.4	0.468	0.55	0.59	0.62	0.67	0.697	0.721
	Hybrid MC-SeCF	**0.579**	0.58	0.626	0.659	0.678	0.711	0.745	0.776	0.808	0.839
	BLI_{GA}	0.524	**0.586**	**0.658**	**0.741**	**0.75**	**0.761**	**0.754**	**0.792**	**0.815**	**0.845**
HetRec 2011	COS	0.24	0.25	0.278	0.3	0.313	0.328	0.357	0.385	0.41	0.452
	PRC	0.281	0.334	0.358	0.4	0.42	0.453	0.458	0.471	0.506	0.537
	SimGen	0.44	0.501	0.543	0.578	0.599	0.642	0.652	0.68	0.699	0.715
	GA1	0.33	0.354	0.4	0.485	0.54	0.584	0.61	0.64	0.69	0.7
	ItemCFGA	0.3	0.325	0.384	0.452	0.5	0.541	0.584	0.621	0.65	0.691
	Hybrid MC-SeCF	0.538	0.566	0.619	0.648	0.66	0.689	0.71	0.725	0.76	0.778
	BLI_{GA}	**0.55**	**0.568**	**0.62**	**0.66**	**0.685**	**0.73**	**0.756**	**0.785**	**0.79**	**0.81**
SD	COS	0.15	0.16	0.16	0.2	0.22	0.25	0.3	0.33	0.35	0.41
	PRC	0.19	0.202	0.226	0.252	0.3	0.334	0.355	0.384	0.414	0.444
	SimGen	0.4	0.43	0.48	0.51	0.53	0.54	0.57	0.58	0.59	0.61
	GA1	0.31	0.33	0.34	0.4	0.46	0.45	0.46	0.49	0.5	0.51
	ItemCFGA	0.27	0.31	0.32	0.38	0.4	0.42	0.44	0.43	0.45	0.47
	Hybrid MC-SeCF	0.413	0.413	0.424	0.459	0.505	0.52	0.561	0.594	0.628	0.641
	BLI_{GA}	**0.43**	**0.45**	**0.5**	**0.55**	**0.57**	**0.56**	**0.6**	**0.6**	**0.62**	**0.65**

Table 10
Precision results of the recommendation methods.

Dataset	Recommendation method	N									
		Top-2	Top-4	Top-6	Top-8	Top-10	Top-12	Top-14	Top-16	Top-18	Top-20
Movielens	COS	0.351	0.34	0.335	0.331	0.33	0.323	0.328	0.32	0.315	0.31
	PRC	0.359	0.349	0.338	0.338	0.335	0.331	0.33	0.329	0.324	0.32
	SimGen	0.612	0.595	0.591	0.587	0.581	0.578	0.574	0.57	0.565	0.561
	GA1	0.485	0.472	0.436	0.39	0.38	0.37	0.362	0.363	0.36	0.34
	ItemCFGA	0.413	0.385	0.358	0.35	0.355	0.351	0.35	0.349	0.349	0.344
	Hybrid MC-SeCF	0.735	0.711	0.7	0.691	0.682	0.677	0.665	0.653	0.652	0.649
	BLI_{GA}	**0.985**	**0.95**	**0.946**	**0.942**	**0.94**	**0.931**	**0.924**	**0.9088**	**0.899**	**0.8893**
HetRec 2011	COS	0.361	0.351	0.321	0.315	0.31	0.2902	0.3	0.2926	0.2888	0.29
	PRC	0.37	0.369	0.351	0.3214	0.32	0.315	0.31	0.31	0.308	0.3
	SimGen	0.605	0.59	0.592	0.577	0.57	0.565	0.5541	0.55	0.5477	0.5395
	GA1	0.452	0.451	0.431	0.381	0.378	0.373	0.36	0.354	0.344	0.349
	ItemCFGA	0.4	0.369	0.36	0.35	0.352	0.3465	0.342	0.349	0.338	0.3396
	Hybrid MC-SeCF	0.725	0.713	0.702	0.69	0.688	0.677	0.665	0.653	0.652	0.649
	BLI_{GA}	**0.982**	**0.97**	**0.958**	**0.946**	**0.934**	**0.922**	**0.916**	**0.89**	**0.886**	**0.874**
SD	COS	0.3	0.31	0.29	0.295	0.29	0.281	0.28	0.28	0.27	0.254
	PRC	0.323	0.312	0.301	0.302	0.298	0.3	0.29	0.289	0.286	0.283
	SimGen	0.61	0.59	0.587	0.58	0.587	0.59	0.587	0.58	0.55	0.55
	GA1	0.41	0.42	0.4	0.405	0.39	0.392	0.391	0.39	0.38	0.38
	ItemCFGA	0.37	0.358	0.35	0.329	0.322	0.324	0.317	0.311	0.311	0.31
	Hybrid MC-SeCF	0.653	0.651	0.65	0.638	0.637	0.623	0.618	0.624	0.619	0.605
	BLI_{GA}	**0.71**	**0.69**	**0.687**	**0.68**	**0.687**	**0.69**	**0.687**	**0.68**	**0.665**	**0.65**

1. The recommended items are semantically correlated.
2. The neighbors of *AU* satisfied with the recommended items.
3. The predicted ratings of the recommended items are as high as possible.

To find the list that meets those features, BLI_{GA} adopts multi-filtering levels, each one related to a specific feature. At each level, the GA discards worst lists. Thus, the number of choices that BLI_{GA} should search on it is decreased from generation to another.

The authors can conclude that the performance of BLI_{GA} is affected with the dataset size, the selected number of *AU*'s neighbors and the number of recommended items (i.e. size of the individual). Thus, the *k*-neighbors and $Top - N$ should be chosen carefully and experimentally with a feasible dataset size to achieve the best possible performance. The results showed the remarkable improvements that have been made by BLI_{GA} in terms of prediction accuracy and recommendation quality. One of the fundamental differences between BLI_{GA} and other methods are in the mechanism of selecting the k-neighbors. BLI_{GA} select a different set of k-neighbors

Table 11
F1-measure results of the recommendation methods.

Dataset	Recommendation method	N									
		Top-2	Top-4	Top-6	Top-8	Top-10	Top-12	Top-14	Top-16	Top-18	Top-20
Movielens	COS	0.2694	0.305	0.3109	0.3253	0.3265	0.336	0.348	0.352	0.35741	0.381
	PRC	0.312	0.3455	0.3528	0.3599	0.3699	0.3782	0.3844	0.3871	0.3981	0.4069
	SimGen	0.5051	0.5589	0.5803	0.5992	0.605	0.6185	0.6183	0.6283	0.6293	0.6422
	GA1	0.3782	0.4456	0.4514	0.4382	0.4461	0.45	0.4541	0.4696	0.4755	0.4659
	ItemCFGA	0.3442	0.356	0.3778	0.4005	0.4315	0.4401	0.4421	0.4517	0.4651	0.4658
	Hybrid MC-SeCF	0.6477	0.6389	0.6609	0.6746	0.68	0.6936	0.7027	0.7092	0.7217	0.7319
	BLI_{GA}	**0.6841**	**0.7249**	**0.7761**	**0.8295**	**0.8343**	**0.8375**	**0.8304**	**0.8464**	**0.8549**	**0.8666**
HetRec 2011	COS	0.2883	0.292	0.298	0.3073	0.3115	0.3079	0.326	0.3325	0.3389	0.3533
	PRC	0.3194	0.3506	0.3545	0.3564	0.3632	0.3716	0.3697	0.3739	0.3829	0.3849
	SimGen	0.5095	0.5419	0.5664	0.5775	0.5841	0.601	0.5991	0.6081	0.6142	0.615
	GA1	0.3815	0.3967	0.4149	0.4268	0.4447	0.4552	0.4528	0.4559	0.4591	0.4658
	ItemCFGA	0.3429	0.3456	0.3716	0.3945	0.4131	0.4224	0.4314	0.4469	0.4447	0.4554
	Hybrid MC-SeCF	0.6177	0.6311	0.6579	0.6683	0.6737	0.6829	0.6868	0.6871	0.7019	0.7077
	BLI_{GA}	**0.7051**	**0.7165**	**0.7528**	**0.7775**	**0.7904**	**0.8148**	**0.8283**	**0.8342**	**0.8353**	**0.8408**
SD	COS	0.2	0.2111	0.2062	0.2384	0.2502	0.2646	0.2897	0.303	0.3048	0.3137
	PRC	0.2393	0.2452	0.2582	0.2747	0.299	0.3161	0.3192	0.3298	0.3383	0.3457
	SimGen	0.4832	0.4975	0.5281	0.5428	0.557	0.5639	0.5784	0.58	0.5693	0.5784
	GA1	0.3531	0.3696	0.3676	0.4025	0.4221	0.419	0.4227	0.4343	0.4318	0.4355
	ItemCFGA	0.3122	0.3323	0.3343	0.3527	0.3568	0.3658	0.3685	0.3609	0.3678	0.3736
	Hybrid MC-SeCF	0.506	0.5054	0.5132	0.5339	0.5634	0.5669	0.5881	0.6086	0.6235	0.6225
	BLI_{GA}	**0.5356**	**0.5447**	**0.5788**	**0.6081**	**0.6231**	**0.6182**	**0.6406**	**0.6375**	**0.6417**	**0.65**

for each candidate recommendation list. Thus, the rating of a particular item may be predicted based on different k-neighbors. In other words, the neighbors of *AU* are selected to be the Top-k similar users who have given ratings to at least one item belong to the recommendation list. While other methods predict the item's ratings using the same neighbors set.

Cold-start and sparsity are two main issues in the face CFRS. Accordingly, BLI_{GA} aims to alleviate cold start and sparsity problems to improve the recommendation quality and RS's accuracy. BLI_{GA} is evaluated on three datasets with different sizes and sparsity levels (i.e. sparsity level of Movielens, HetRec 2011 and SD are 93.7%, 96.1%, and 95.6%, respectively). Nonetheless, BLI_{GA} achieved fewer prediction errors (i.e. [0.556, 0.726]) than other recommendation methods using Movielens. Besides, BLI_{GA} achieved [0.589, 0.703] MAE results using HetRec 2011. When using SD, the BLI_{GA}'s MAE results (i.e. [0.6, 0.7]) are better than other recommendation techniques. The improvements percentage achieved by BLI_{GA} when using Movielens is higher than the one made using SD and HetRec 2011. Note that the size and a density level of SD are lower than MovieLens's size and sparsity level. However, BLI_{GA} makes a more accurate prediction than other CF methods regardless of the dataset's size and sparsity level.

With regard to the cold-start problem, the BLI_{GA}'s performance in terms of accuracy and recommendation quality outperforms the other RSs regardless of the available amount of the historical information about *AU*, the selected *k*-neighbors and the number of recommended items. In Movielens, 32.4% of the users gave rates to 20 - 40 items which are a small number of rates compared with remaining users. This means there is a 32.4% probability that the selected *AU* gave rates to 20-40 items which is the highest selection probability. For HetRec 2011, 24% of the users gave ratings to 20 - 100 items, with the average number of rates being 405. As with Movielens, 24% of users have the highest probability to be selected as the AU. In SD, 105 users gave rates to 10-15 items. This means there is a 35% probability that the selected *AU* gave rates to 10-15 items which are the highest selection probability. Moreover, for testing purposes, 40% of the items which were rated by *AU* have been hidden. Besides, BLI_{GA} addresses the cold-start problem at the item level. By meaning, the new items enter the system have the same probability to be recommended to *AU* as the old items. This is because BLI_{GA} fills the individuals (i.e. candidate recommendation lists) with random items. Overall, BLI_{GA} made is a significant improvement with respect to the prediction accuracy and recommendation quality. This demonstrates its effectiveness in alleviating the sparsity and cold-start problems of CF.

5. Conclusion and future work

This research work presented a multi-filtering level CF technique, BLI_{GA}. BLI_{GA} produces recommendations to *AU* using the GAs approach. The main idea is to search for a list contains high correlated items in terms of semantic and those items have two characteristics: rated by the neighbors of *AU* and potential favorite items of high predicted values of ratings. BLI_{GA} searches for the best recommendation list instead of searching for the best items to form the recommendation list. Thus, the initial population contains a set of individuals, each represents a list of random potential recommendation lists. Those individual subjects to three filtering levels:

1. The semantic correlation filtering criterion. The GA gives a semantic-based rating to each individual based on the semantic similarity degree between every two items belong to the individual. The higher the semantic similarity degree between items, the higher the semantic-based rating of this individual.

2. The satisfaction-based filtering criterion. The GA gives a satisfaction-based rating to each individual based on the degree of harmony between *AU* and the users who were rated at least one item belongs to the individual (i.e. satisfaction-based similarity).
3. The predicted satisfaction degree filtering criterion. The GA predicts the satisfaction degree of *AU* about each individual. The rating prediction process of the items involves different k-neighbors for each list. Since the k-neighbors set is a subset of the users' set who gave a rating to at least one item belonging to the candidate list. BLI_{GA} computes the predicted satisfaction degree based on the predicted rating of each item belong to the individual. The *BLI* (i.e. selected recommendation list) is the individual that has the highest predicted satisfaction degree.

The multi-criteria idea has been adopted in BLI_{GA} in two different aspects. The first aspect is adopting the idea to compute the item-based semantic similarity between every two items belong to the individual. The semantic similarity is computed based on the common semantic features between the items. The second aspect is that BLI_{GA} generates the recommendation based on multi-filtering criteria. Each individual has multi-fitness values that have been considered as ratings gave by the GA to measure the fit degree of this individual.

BLI_{GA} alleviate the cold start and sparsity in a novel way that is considering the individual in the population as a potential recommendation list. Consequently, the proposed method does not depend on any similarity metric to produce the recommendation list but to evaluate a potential recommendation list. Moreover, the usage of item-based semantic correlation as the first filtering level contributes to producing recommendations without the full dependency on users' historical rating data. Also, these individuals are filled randomly with the items. Therefore, the new item that enters the system has the same probability as the old one to be selected and added to the individual. Due to this, the item cold-start issue is alleviated. With regard to the sparsity problem, BLI_{GA} is evaluated on three datasets that have different size and sparsity level. The proposed RS has been compared to alternative techniques in terms of prediction accuracy and recommendation quality. The results obtained prove that BLI_{GA} has better performance than other techniques.

There are two limitations to this research work that could be addressed in future research. The first limitation stems from the specifications of the used machine in the experiments. This contributed to testing the method on small and medium sized datasets which is not feasible for a realistic scenario. The third limitation arises from the use of only the genre feature of the items as a semantic feature. In the future, more features may be considered and tested.

CRediT authorship contribution statement

Bushra Alhijawi: Conceptualization, Data curation, Formal analysis, Investigation, Writing - original draft, Writing - review & editing. **Yousef Kilani:** Conceptualization, Data curation, Formal analysis, Investigation.

References

Alhijawi, B. (2019a). Improving collaborative filtering recommender system results and performance using satisfaction degree and emotions of users. *Web Intelligence, 17*(3), 229241. https://doi.org/10.3233/WEB-190415.

Alhijawi, B. (2019b). *Improving collaborative filtering recommender system results using optimization technique. Proceedings of the 3rd international conference on advances in artificial intelligence*. ACMhttps://doi.org/10.1145/3369114.3369126.

Alhijawi, B., Al-Naymat, G., Obeid, N., & Awajan, A. (2019). *Mitigating the effect of data sparsity: A case study on collaborative filtering recommender system. Proceedings of the 2nd international conference on new trends in computing sciences (ICTCS)*. IEEE1–6. https://doi.org/10.1109/ictcs.2019.8923064.

Alhijawi, B., & Kilani, Y. (2016). *Using genetic algorithms for measuring the similarity values between users in collaborative filtering recommender systems. Proceedings of the IEEE/ACIS 15th international conference on computer and information science (ICIS)*. IEEEhttps://doi.org/10.1109/icis.2016.7550751.

Alhijawi, B., & Kilani, Y. (2020). The recommender system: A survey. *International Journal of Advanced Intelligence Paradigms, 15*(3), 229–251. https://doi.org/10.1504/IJAIP.2020.105815.

Alhijawi, B., Obeid, N., Awajan, A., & Tedmori, S. (2018). *Improving collaborative filtering recommender systems using semantic information. Proceedings of the 9th international conference on information and communication systems (ICICS)*. IEEEhttps://doi.org/10.1109/iacs.2018.8355454.

Bobadilla, J., Ortega, F., Hernando, A., & Alcalá, J. (2011). Improving collaborative filtering recommender system results and performance using genetic algorithms. *Knowledge-Based Systems, 24*(8), 1310–1316. https://doi.org/10.1016/j.knosys.2011.06.005.

Bobadilla, J., Ortega, F., Hernando, A., & Gutiérrez, A. (2013). Recommender systems survey. *Knowledge-Based Systems, 46*, 109–132. https://doi.org/10.1016/j.knosys.2013.03.012.

Chen, L., Chen, G., & Wang, F. (2015). Recommender systems based on user reviews: the state of the art. *User Modeling and User-Adapted Interaction, 25*(2), 99–154. https://doi.org/10.1007/s11257-015-9155-5.

Gao, L., & Li, C. (2008). *Hybrid personalized recommended model based on genetic algorithm. Proceedings of the 4th International conference on wireless communications, networking and mobile computing*. IEEEhttps://doi.org/10.1109/wicom.2008.2152.

Gohari, F. S., Haghighi, H., & Aliee, F. S. (2016). A semantic-enhanced trust based recommender system using ant colony optimization. *Applied Intelligence, 46*(2), 328–364. https://doi.org/10.1007/s10489-016-0830-y.

Gupta, A., Shivhare, H., & Sharma, S. (2015). *Recommender system using fuzzy c-means clustering and genetic algorithm based weighted similarity measure. Proceedings of the international conference on computer, communication and control (IC4)*. IEEEhttps://doi.org/10.1109/ic4.2015.7375707.

Ho, Y., Fong, S., & Yan, Z. (2007). A hybrid ga-based collaborative filtering model for online recommenders.,. (pp. 200–203).

Jannach, D., Karakaya, Z., & Gedikli, F. (2012). *Accuracy improvements for multi-criteria recommender systems. Proceedings of the 13th ACM conference on electronic commerce - EC '12*. ACM Presshttps://doi.org/10.1145/2229012.2229065.

Jia, Y. B., Ding, Q. Q., Liu, D. L., Zhang, J. F., & Zhang, Y. L. (2014). *Collaborative filtering recommendation technology based on genetic algorithm. Applied mechanics and materials599*. *Applied mechanics and materials* Trans Tech Publ1446–1452. https://doi.org/10.4028/www.scientific.net/AMM.599-601.1446.

Kim, H.-T., Kim, E., Lee, J.-H., & Ahn, C. W. (2010). *A recommender system based on genetic algorithm for music data. Proceedings of the 2nd International conference on computer engineering and technology*. IEEEhttps://doi.org/10.1109/iccet.2010.5486161.

KIM, K., & AHN, H. (2008). A recommender system using GA k-means clustering in an online shopping market. *Expert Systems with Applications, 34*(2), 1200–1209. https://doi.org/10.1016/j.eswa.2006.12.025.

Lee, C.-I., Hsia, T.-C., Hsu, H.-C., & Lin, J.-Y. (2017). *Ontology-based tourism recommendation system. Proceedings of the 4th International conference on industrial engineering and applications (ICIEA)*. IEEEhttps://doi.org/10.1109/iea.2017.7939242.

Liu, L., Mehandjiev, N., & Xu, D.-L. (2011). *Multi-criteria service recommendation based on user criteria preferences. Proceedings of the fifth ACM conference on recommender systems - RecSys '11*. ACM Presshttps://doi.org/10.1145/2043932.2043950.

Lu, J., Wu, D., Mao, M., Wang, W., & Zhang, G. (2015). Recommender system application developments: A survey. *Decision Support Systems, 74*, 12–32. https://doi.org/10.1016/j.dss.2015.03.008.

Lv, G., Hu, C., & Chen, S. (Hu, Chen, 2016a). Research on recommender system based on ontology and genetic algorithm. *Neurocomputing, 187*, 92–97. https://doi.org/10.1016/j.neucom.2015.09.113.

Marmanis, H., & Babenko, D. (2009). *Algorithms of the intelligent web minning.* Manning Publications.

Nilashi, M., Ibrahim, O., & Bagherifard, K. (2018). A recommender system based on collaborative filtering using ontology and dimensionality reduction techniques. *Expert Systems with Applications, 92*, 507–520. https://doi.org/10.1016/j.eswa.2017.09.058.

Nilashi, M., bin Ibrahim, O., Ithnin, N., & Sarmin, N. H. (bin Ibrahim, Ithnin, Sarmin, 2015a). A multi-criteria collaborative filtering recommender system for the tourism domain using expectation maximization (EM) and PCA–ANFIS. *Electronic Commerce Research and Applications, 14*(6), 542–562. https://doi.org/10.1016/j.elerap.2015.08.004.

Parveen, H., Ashraf, M., & Parveen, R. (2015). *Improving the performance of multi-criteria recommendation system using fuzzy integrated meta heuristic. International conference on computing, communication & automation*. IEEEhttps://doi.org/10.1109/ccaa.2015.7148394.

Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to recommender systems handbook. Recommender systems handbook*. Springer1–35.

Salehi, M. (2014). Latent feature based recommender system for learning materials using genetic algorithm. *Information Systems & Telecommunication, 137*.

Shambour, Q., & Lu, J. (2011). *A hybrid multi-criteria semantic-enhanced collaborative filtering approach for personalized recommendations. Proceedings of the IEEE/WIC/ACM International conferences on web intelligence and intelligent agent technology*. IEEEhttps://doi.org/10.1109/wi-iat.2011.109.

Sharma, L., & Gera, A. (2013). A survey of recommendation system: Research challenges. *International Journal of Engineering Trends and Technology (IJETT), 4*(5), 1989–1992.

Vesin, B., Ivanović, M., Klašnja-Milićević, A., & Budimac, Z. (2012). Protus 2.0: Ontology-based semantic recommendation in programming tutoring system. *Expert Systems with Applications, 39*(15), 12229–12246. https://doi.org/10.1016/j.eswa.2012.04.052.

Wang, Z., Yu, X., Feng, N., & Wang, Z. (2014). An improved collaborative movie recommendation system using computational intelligence. *Journal of Visual Languages & Computing, 25*(6), 667–675. https://doi.org/10.1016/j.jvlc.2014.09.011.

Xiao, J., Luo, M., Chen, J.-M., & Li, J.-J. (2015). *An item based collaborative filtering system combined with genetic algorithms using rating behavior. Lecture notes in computer science*. Springer International Publishing453–460. https://doi.org/10.1007/978-3-319-22053-6_48.

Zarrinkalam, F., & Kahani, M. (2012). *A multi-criteria hybrid citation recommendation system based on linked data. Proceedings of the 2nd International eConference on computer and knowledge engineering (ICCKE)*. IEEEhttps://doi.org/10.1109/iccke.2012.6395393.

Zhang, F., & you Chang, H. (2006). *A collaborative filtering algorithm employing genetic clustering to ameliorate the scalability issue. Proceedings of the IEEE international conference on e-business engineering (ICEBE'06)*. IEEEhttps://doi.org/10.1109/icebe.2006.2.