

## An Effective Algorithm in a Recommender System Based on a Combination of Imperialist Competitive and Firefly Algorithms

Zahra Khorsand · Reza Mortazavi\*

Received: 13 June 2016 / Accepted: 18 October 2016

**Abstract** With the rapid expansion of the information on the Internet, recommender systems play an important role in terms of trade and research. Recommender systems try to guess the user's way of thinking, using the information of user's behavior or similar users and their views, to discover and then propose a product which is the most appropriate and closest product of user's interest. In the past decades, many studies have been done in the field of recommender systems, most of which have focused on designing new recommender algorithms based on computational intelligence algorithms. The success of a recommender system besides the quality of the algorithm depends on other factors such as: Sparsity, Cold start and Scalability in the performance of a recommender system, which can affect the quality of the recommendation. Consequently, the main motivation for this research is to providing an effective meta heuristic algorithm based on a combination of imperialist competitive and firefly algorithms using clustering technique. The simulation results of the proposed algorithm on real data sets Move Lens and Film Trust have shown better forecast accuracy in the item recommendation to users than other algorithms presented in subject literature. Also the proposed algorithm can choose appropriate items among the wide range of data and give it to output in a reasonable time.

---

Z. Khorsand  
Islamic Azad University, Damghan Branch, Damghan, Iran  
Tel.: +98-9359326915  
E-mail: khorsand87z@gmail.com

\*Corresponding author  
R. Mortazavi  
School of Engineering, Damghan University, Damghan, Iran  
Tel.: +98-233 522 0081-6 (438)  
E-mail: r.mortazavi@du.ac.ir

**Keywords** recommender systems · computational intelligence · clustering · imperialist competitive algorithm · firefly algorithm.

**Mathematics Subject Classification (2010)** 8T20 · 90C59 · 68P20.

## 1 Introduction

In recent years we have witnessed an explosion of information with an exponential growth in the Internet. The internet is facing a lot of overhead information and there are difficulties for users for facing a large amount of information. It seems that a solution which helps users find their favorite items should be sought. One of the solutions is providing a recommender system which offers favorite items to users [?]. Almost in the mid 1990's, studies on the recommender system were raised as an independent branch of research. The reason for this special attention was that researchers tend to solve the problem of recommender methods which were utilized in an initial approach to a query problem in a large amount of information. Such systems provide a list of items which might be preferred by the user or they do a prediction in the case of user interest in any items. Different definitions have been proposed for recommender systems, including a holistic and summarized definition of Mr. TP Ling in 2007, which defines a recommender system as information systems that can analyze past behaviors and provide recommendations for current problems [?]. In simple terms, recommender systems try to guess the user's way of thinking, using the information of user's behavior or similar users and their views, to discover and then to propose a product which is the most appropriate and the closest product to user's interest. The recommendations of recommender systems generally contain two consequences: [?]

- (1) Help users to decide among several items.
- (2) Increase user's knowledge in his favorite field.

In a relatively short time, a variety of recommender methods has been created and a wide variety of these systems are available that all use the benefits of a particular set of techniques of artificial intelligence, including clustering information methods which are effective techniques for providing an accurate recommendation to help these systems. So these methods cluster information based on the required factors and give the information to recommender systems to provide better recommendations for users. The success of a recommender system does not depend only on the quality of an algorithm, but there are certain challenges in the performance of a recommender system that affect the quality of the recommendation [?]. The first challenge is Sparsity problem, meaning that, however, there is information in a system, the information is sparse; therefore it is not possible to realize correctly and firmly whether the item is more acceptable [?]. Another challenge in recommender systems named Cold start. Cold start occurs for new users or items just entered the system. Also, it occurs when there is not enough information [?]. Scalability problem refers to the ability for handling a large amount of information in an efficient and effective manner [?]. According to the research conducted so far

and important applications of a recommender system, researchers always have noticed providing an efficient algorithm to improve decision making in complex environments. So, the main motivation for this research is to provide an effective meta heuristic algorithm based on a combination of imperialist competitive and firefly algorithms so that recommendation methods try to improve prediction accuracy. Also, it tries to choose appropriate items among the wide range of data; then it gives the selected items to output at a reasonable time. The structure of this paper is as follows: Section 2 presents the related works. In Section 3, the proposed algorithm will be detailed. The simulation results are shown in Section 4. Finally, Section 5 concludes the paper.

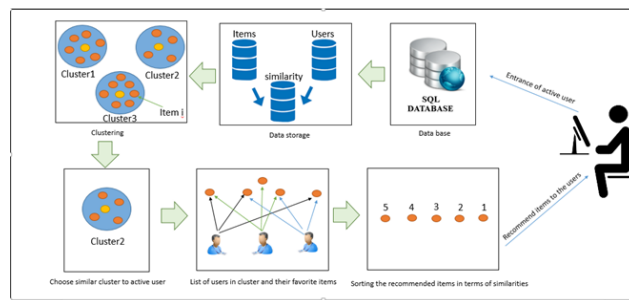
## 2 Related works

In the past decade, many studies have been done in the field of recommender systems that most of the studies have focused on designing new recommendation algorithms based on computational intelligence algorithms. The algorithms used in computational intelligence are often mathematical tools which have been somehow inspired by nature and the world around. In general, most of the algorithms have used three approaches for recommendation: Content-based filtering [?], Collaborative filtering [?], and Knowledge-based [?]. In a content-based filter for providing a recommendation, choices and experiences of active users in the past are used. In Collaborative filters which are one of the most frequently used existing techniques, for providing a recommendation, comments and ratings done by neighbor users is used. In the Knowledge-based methods, the recommendation is provided based on the perception that a system has received about users and features of items. The general purpose of each of these methods is choosing the best items from a large collection of available items, so that always leads to the satisfaction of the users. In the following, we will explain some of the algorithms proposed by researchers. Ujjin and Bentley in [?] explained recommender systems implemented by the particle optimization algorithms. They combined the recommender system with an idea based on Pearson and achieved a predictive algorithm with higher accuracy than systems based on genetic algorithm and Pearson. Lorenzi et al. [?] presented a system called CASIS which modeled the social behavior of a group of insects. This model was inspired by a bee dance and used in recommender systems within the recommendation step. Sobecki [?] presented a recommender system based on ants algorithm in 2008. He utilized all three techniques of filtering information. Mohammed yaha et al. [?] presented a recommender system using a combination idea of genetic and fuzzy algorithms. The presented model is a memory based method. However, this method is flexible and scalable; it may have some problem with time, because it does the customizing process in an online phase. So it is possible that the user exits the website, before providing customized environment. Handel and Meyer [?] categorized ant-based clustering methods into two main groups: methods that directly follow the clustering behavior of real ant colonies, methods that in-

directly inspired by the nature. Jesus and Fernando [?] improved results and performance of the collaborative filtering recommender system by using genetic algorithm. A metric to measure the similarity between users was shown that is executable in the processes of collaborative filtering of recommendation system. This metric was formulated using a simple linear combination of value and weight. Salehi et al. [?] presented a feature-based hybrid recommender system for worksheets using genetic algorithm and multidimensional data model. Davynam and Divya [?] proposed a recommender system for music data using a genetic algorithm. They have tried to improve a recommender system for music data that predicts electronic user's music according to some information filtering algorithms. Animesh and sink [?] presented an implementation of a hybrid genetic algorithm for clustering-based data for a recommender system. They have proposed a new hybrid clustering algorithm and executed it using genetic and K-NN algorithms. According to the results, the proposed hybrid method is effective and efficient for the given program. A hybrid approach was shown in [?] to dealing with issues uncertainty and data sparsity in the customer and product data for products and communication services. This method utilized the fuzzy set techniques in connection with user-based and item-based filtering to deal with the similarity of the fuzzy products. Liu shambour et al. [?] proposed a method for personalized business partner recommendation in small and medium businesses. To reduce the sparsity and cold start a product similar analysis for the development of the product semantic communication method was combined. A Fuzzy modeling based recommendation framework is presented by Chang Cheng, L. C., & Wang, H. A. [?] which uses objective and subjective information. However, subjective information includes expert opinions; objective information is based on the similar users' preferences and their past experiences. Hsu et al. [?] presented a personal and scalable system to recommend teaching aids on Facebook using an artificial bee colony algorithm. This method provided teaching aids, based on the difficulty level of teaching aids, the number of likes of a special education case, and unique teaching style and subject.

### 3 The proposed recommender system

The proposed clustering-based recommender system is based on imperialist competitive [?] and firefly algorithms [?] that contains online and offline phases. Firstly, in the offline phase, the clustering algorithm tries to classify items into several groups so that the items in each group have maximum similarity to one another. The online phase is a recommender phase. In other words, in the online phase, favorite items of users who have logged into the site or system are predicted and recommended. Figure 1 shows the schema of the proposed system. In the proposed method in the offline phase first a list of all items is fetched from the system and located in the data warehouse. Then the similarity between items is calculated based on the Pearson correlation coefficient. Afterwards the proposed algorithm clusters the items. Clustering



**Fig. 1** an overview of the proposed method in the recommender system

means that the items that are in a cluster are more similar and there is a cluster head within each cluster that has the most similarities with other members of its own cluster. All these stages are done in the offline phase. So it can be said that the criterion of clustering quality has greater priority in the online phase and the clustering algorithm should always do the best classification. A high quality clustering means that the items inside a cluster are similar as much as possible and the items corresponding to different clusters are different to each other as much as possible. Then, in the online phase for the active user logged into the system, the best cluster in terms of the highest similarity is selected among the items that the active user didn't see before. Rating items are predicted based on the  $k$  nearest neighbors within each cluster. Among the predicted items, the item that has the highest predicted value is recommended to the user.

Figure 1::: active user login- database- Data Warehouse-clustering- selecting a similar cluster active user- List of users in a cluster and their favorite items- Sorting the recommended items based on the similarity- Recommend items to the user.

In the following, firstly, we describe the online phase, which contains: (1) fetch data from the data warehouse, (2) calculating the Pearson correlation coefficient between data and (3) clustering.

### 3.1 Fetch data from the data warehouse

At this stage of the offline phase, favorite items to all users and the corresponding rates for items will be fetched from the database and then placed in a two-dimensional matrix in the data warehouse.

In Table 1, a unique number per row shows a user in the database and each column determines items or products on the system. The content of each cell of the matrix specifies rating to item by corresponding user. Also a sample matrix for active users is fetched when logging into the system. This matrix contains one row and  $M$  columns that the contents of each cell show the rate of each item. When an active user logs into the system, a matrix is fetched

**Table 1** a matrix of rating items by users

Users	Items				
	Item 1	Item 2	Item 3	...	Item M
User 1	3	5	?	...	?
User 2	5	?	4	...	1
User 3	3	5	?	...	?
...	..	...	...	...	-
User N	5	1	3	-	?

**Table 2** a matrix of rating items by active users

User	Items				
	Item 1	Item 2	Item 3	...	Item M
Active User	?	?	2	...	1

and placed in the data warehouse. Table 2 tabulates rating items by active users.

According to Table 2, the recommender system should recommend items for the active user among the items that the user didn't predict; the recommended items should most likely be his favorite. So after fetching information, the user similarity to one another as well as the similarity of each item to one another should be calculated. Then we predict the rates of the items which were not previously rated by the user.

### 3.2 The calculation of similarities

To calculate the similarity of the items in the system relative to each other as well as system users based on the table, the matrix is fetched and the Pearson correlation coefficient was used.

$$PC(a, b) = \frac{\sum_{i=1}^m (R_{ai} - \bar{R}_a) \times (R_{bi} - \bar{R}_b)}{\sqrt{\sum_{i=1}^n (R_{ai} - \bar{R}_a)^2} \times \sqrt{\sum_{i=1}^l (R_{bi} - \bar{R}_b)^2}} \quad (1)$$

where  $PC(a,b)$  denotes the correlation-based similarity between items a and b. m denotes the number of users who have rated to each item a and b. n indicates the number of users who have rated to each item a. l indicates the number of users who have rated to each item b.  $R_{ai}$  denotes the rating that the user i gave for item a and  $R_{bi}$  denotes the rating that the user i gave for item b.  $\bar{R}_a$  shows the average rating that item a has received and  $\bar{R}_b$  shows the average rating that item b has received. All these amounts are calculated based on the matrix 1. In the following, we measure the similarity between two users in the system. This similarity is determined based on the equation 2.

$$PC(U, V) = \frac{\sum_{i \in I_{uv}} (R_{ui} - \bar{R}_u) \times (R_{vi} - \bar{R}_v)}{\sqrt{\sum_{i \in I_{uv}} (R_{ui} - \bar{R}_u)^2} \times \sqrt{\sum_{i \in I_{uv}} (R_{vi} - \bar{R}_v)^2}} \quad (2)$$

where  $PC(U, V)$  denotes the similarity between users  $U$  and  $V$ .  $I_{uv}$  contains items which were rated by both users.  $R_{ui}$  shows the rating that the user  $U$  gave for item  $i$  and  $R_{vi}$  shows the rating that the user  $V$  gave for item  $i$ .  $\bar{R}_v$  and  $\bar{R}_u$  denote the average ratings that each user gave for items that will be calculated as (3) and (4) respectively.

$$\bar{R}_u = \frac{1}{|I_{uv}|} \times \sum_{i \in I_{uv}} R_{ui} \quad (3)$$

$$\bar{R}_v = \frac{1}{|I_{uv}|} \times \sum_{i \in I_{uv}} R_{vi} \quad (4)$$

The similarities obtained from Pearson correlation coefficient 2 and 3, will be in the range of -1 to 1. The more the two users are similar the value is closer to 1.

### 3.3 The Proposed Clustering Algorithm

One major issue in introducing the recommender system in the proposed method is the clustering step. The more the clustering algorithm can cluster data accurately, the more the interesting items of the user are utilized. Although this step is used in the offline phase, but the algorithm should have the adequate speed and classification accuracy. To obtain these purposes, a combination of imperialist competitive and firefly algorithms are used in the proposed method. The goal of this combination is using the global and local search benefits of both algorithms. The Pseudo code of the proposed algorithm which combines these two algorithms is represented in Algorithm1. In the proposed method, the firefly algorithm is implemented inside the imperialist competitive algorithm trying to improve the solutions. In the pseudo code of the proposed method, at first the parameters of the problem as well as the imperialist competitive and firefly algorithms are received, then the imperialist competitive algorithm generates population from the country (solution) randomly, then revolution and absorption operations are applied on each imperialist with a specific rate. Then, to improve the population of the imperialist, the firefly algorithm runs inside each imperialist. After improving the population using firefly algorithm, the imperialists compete and those with no colonies are eliminated. This procedure proceeds until the termination condition. The termination condition is considered as achieving to a specific iteration number or a single imperialist. The details of these steps will explain in the following.

**Algorithm 1** Pseudo code of the proposed algorithm for clustering the items

---

```

1: Function ICAFA (problemsize)
2: Input populationsize, problemsize, NumberEmpires, PR, PA, RateUpdate, Minsize, M, Dmax
3: Output Sbest
4: Population  $\leftarrow \emptyset$ ;
5: EmpiresPopulation  $\leftarrow \emptyset$ ;
6: for  $i = 1$  to Populationsize do
7:    $C_{i\text{position}} \leftarrow \text{RandomPosition}(\text{Problem}_{\text{size}})$ ;
8:   Population  $\leftarrow C_{i\text{position}}$ ;
9: end
10: InitialEmpiresPopulation (EmpiresPopulation, Population, NumberEmpires);
11: while  $\neg \text{StopCondition}(\text{ICA})$  do
12:   for  $i = \text{Empire} \in \text{EmpiresPopulation}$  do
13:     for  $j = 1$  to RateUpdate do
14:        $C_{i\text{Candidate}} \leftarrow \text{GetCandidateColony}(\text{Empires}_i)$ 
15:        $C_{i\text{Candidateposition}} \leftarrow \text{Absorption}(C_{i\text{imperialisti}}, C_{i\text{Candidate}}, P_A)$ ;
16:        $C_{i\text{Candidateposition}} \leftarrow \text{Revolution}(C_{i\text{Candidate}}, P_R)$ ;
17:       if ( $\text{Fitness}(C_{i\text{Candidate}}) > \text{Fitness}(C_{i\text{Candidateposition}})$ ) then
18:          $C_{i\text{Candidateposition}}, C_{i\text{Candidate}}, \text{Empires}_i(\text{Placement})$ 
19:         if ( $\text{Fitness}(C_{i\text{empire}}) > \text{Fitness}(C_{i\text{Candidateposition}})$ ) then
20:            $\text{ExchangeRoles}(C_{i\text{empire}}, C_{i\text{Candidateposition}}, \text{Empires}_i)$ 
21:         end
22:       end
23:     end
24:     while  $\neg \text{StopCondition}(FA)$  do
25:       for  $i = 1$  to Empiresize do
26:         for  $j = i$  to Empiresize do
27:            $\text{UpdatePosition}(F_i, F_i, 0)$ ;
28:         end
29:       end
30:     end
31:      $\text{EvaluateEmpiresPopulation}(\text{Empires}_i)$ ;
32:   end
33:    $\text{ImperialisticCompetition}(\text{EmpiresPopulation})$ ;
34:    $\text{EliminateWeakEmpires}(\text{EmpiresPopulation})$ ;
35: end
36:  $S_{\text{best}} \leftarrow \text{GetBestSolution}(\text{EmpiresPopulation})$ ;
37: return  $S_{\text{best}}$ ;

```

---

### 3.3.1 Generating the initial population

One major issue in each meta-heuristic algorithm is the manner of coding the solutions. In the imperialist algorithm the coding for representing a solution is known as a country and each country is represent a solution in the problem space. In the proposed algorithm for representing each county in the clustering step, a one dimensional array with  $n$  columns is used. Where  $n$  determines the number of items that must be classified. For each cell of the array, the index number determines the number of the item and the value of the cell determines that this item is a cluster head or not. In the other words, in the proposed method, for representing each county at first it is determined that which items can be selected as a cluster head, in this step it is randomly determined. The 1 value in the array identifies that the corresponding item is a cluster head



1	0	1	0	0	0	1	0	1
---	---	---	---	---	---	---	---	---

**Fig. 2** An example to show how a solution is represent in the proposed algorithm.

and the zero value identifies that the corresponding item is not a cluster head. For clarifying, Figure 2 shows a sample of how a solution is represented in the proposed method.

As shown in Figure2, the clustering should be done on 9 items. The items 1, 3, 7 and 9 are randomly identified as a cluster head. Hence, in this solution sample there are 4 clusters that the items 1, 3, 7 and 9 are cluster head of each cluster and other items must migrate to one of these clusters. The manner of migration is that each item is transferred to the cluster which is the most similar one (according to equation1) to the cluster head item in the corresponding cluster. After migration of all items to the clusters, we will achieve an instance of clustered items, in this step a specified number of populations of these countries will be generated, randomly.

### 3.3.2 Calculating the power of a country

The power of a country in the proposed algorithm represents the fitness and eligibility of a solution. The more the solution is eligible, the more the corresponding solution leads to an optimal solution. As mentioned in the previous sections, a clustering algorithm must partition the data into clusters in a way that members of each cluster have most similarity and the elements of two different clusters have most differences. The eligibility of a solution is a measurement to represent the clustering accuracy in the proposed method which is calculated according to equation 5.

$$EF = \sum_{i=1}^k \sum_{x \in c_i} \frac{p}{Distance(X, \mu(c_i))} + \sum_{i=1}^{k-1} \sum_{j=i+1}^k Distance(\mu(c_i), \mu(c_j)) \quad (5)$$

As shown in equation 5, the clustering accuracy is calculated according to the summation of internal and external distance. Distance in the proposed method means the similarity between two items that is calculated based on the equation 2. More internal similarity of members of a cluster and less external similarity between members of different clusters will lead to better clustering.

In the above equation,  $c_i$  identifies the  $i$ th cluster and  $\mu(c_i)$  identifies the center of cluster  $i$ .  $Distance(X, \mu(c_i))$  is the distance between an object and its center and  $Distance(\mu(c_i), \mu(c_j))$  is the distance between the center of clusters  $i$  and  $j$ . Also,  $p$  is an input parameter which identifies the importance of considering the internal distance of objects. Reduction of EF leads to more accurate clustering and optimal solution.

Imperialist	1	0	0	1	0	0	1	0	0
Colony	1	0	1	0	0	1	0	0	0
New Country	1	0	0	1	0	1	0	0	0

**Fig. 3** The process of the absorption operation with the one-point technique.

### 3.3.3 Empire generation

In this step of the imperialist competitive algorithm,  $m$  countries with better objective function (power) are selected as imperialist country and rest of the countries are colonies as colonies. Then, colonies will randomly and equally be allocated to each of the imperialist countries. In addition, (The remainder of integer division of  $p - m$  by  $m$ ) because of the inherent feature of correct component function, rest of the countries will be allocated to the most powerful empire. In the following in this stage, the power of each empire must be measured. In the proposed algorithm, equation 6 is used to calculate the power of each empire:

$$W_j = F_j + \mathcal{L}(S_j) \quad j = 1, 2, 3, \dots, n \quad (6)$$

where  $W_j$  is the overall power of  $j$ th empire,  $F_j$  is the eligibility of imperialist country in the empire  $j$  and  $S_j$  is the mean of colonies' eligibility in empire  $j$  and  $\mathcal{L}$  is a parameter between 0 and 1 which determines the effect of objective function of imperialist country in proportion to the mean of the objective function of colonies on power of the empire.

### 3.3.4 Absorption policy

In this stage of the imperialist competitive, the colonies must be absorbed towards imperialists in each empire, to update the solution and search the problem space. One-point and two-points techniques are used for applying the absorption operation. Figures 3 and 4 show these two techniques.

**One-point:** In the one-point technique, at first a colony of the empire is selected using roulette wheel, then a random point is selected in the imperialist and the first contents of the imperialist are transferred to the new country and the rest of the contents are selected from the colony.

**Two-point:** In this technique instead of one point, two points are selected in the imperialist and the contents in the range of these two points are transferred to the new country and the rest of the contents are received from imperialist country.

After applying the absorption operation, finally the eligibility power of the new country is calculated and if the new country is better than the colony, the colony will be replaced with the new country and the previous colony will be eliminated. Also if the new colony has more power or better eligibility in comparison with the imperialist, the new colony will be considered as the

Imperialist	1	0	0	1	0	0	1	0	0
Colony	1	0	1	0	0	1	0	0	0
New Country	1	0	0	0	0	1	1	0	0

**Fig. 4** The process of the absorption operation with the two-point technique.

Colony	1	0	0	1	0	0	1	0	0
New Country	1	0	0	1	1	0	1	0	0

**Fig. 5** The process of revolution operator in the proposed algorithm.

imperialist and the previous imperialist will be considered as colony, in the empire. Finally, the power of the empire will be calculated again. In this step the colonies in each empire are absorbed towards their imperialist based on the specified number.

### 3.3.5 Revolution policy

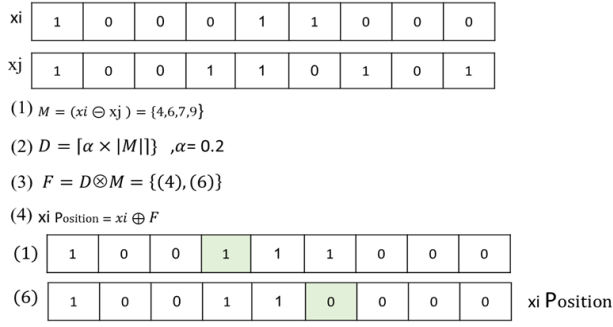
In this stage, according to the entry parameter,  $p$  percent of colonies are selected using roulette wheel and they make a revolution. Applying the revolution operator in the proposed algorithm is in such a way that at first a random point of colony is selected and its content is reversed. The goal of this operation is finding the best cluster head for the items.

### 3.3.6 Improving the solutions using firefly algorithm

Since the population of countries in each empire converges rapidly after a specific number of iterations, the firefly algorithm must be run to improve the solutions and increase the diversity, so that, by improving the solutions, a more diverse space of the problem be explored. For this purpose, the firefly algorithm receives the empire population in each iteration, from the imperialist competition algorithm, and tries to update countries of the received empire. A new method in the discrete space is used for applying the updating operator to the firefly. For this purpose, we found the equation of the absorption of fireflies relative to each other with a little change. Equation 7 represents the fireflies' absorption to each other in a discrete space in the proposed algorithm.

$$X_i = X_i \oplus (X_i \ominus X_j) \otimes \alpha \quad (7)$$

where  $X_i$  represents solution  $i$  in the firefly algorithm which is supposed to be absorbed towards solution  $X_j$ .  $X_i \oplus X_j$  represents the difference of two solutions. The value of  $\alpha$  is between 0 and 1; and the value is selected randomly. For clarifying, consider figure 6 for absorption operation of two fireflies in the discrete space. At first, two fireflies are selected; it is supposed that the firefly  $X_i$  absorbs towards firefly  $X_j$ . In the first step, difference between these two fireflies is calculated. As it is clear, the differences are in the indexes 4, 6, 7, and 9. Then, in the second step, the amount of absorption should be



**Fig. 6** An example of the absorption operation on two fireflies.

determined based on the *alfa* value. According to differences list, the amount of absorption is 2; it means that two instances should be selected from the *m* list which are equal to indexes 4 and 6. Then, in step 4, the contents of indexes 4 and 6 are received from  $X_j$  solution and applied to  $X_i$  solution. In this way, two solutions of the firefly algorithm are absorbed in a discrete space.

### 3.3.7 The imperialist competition operator

In this step, the weakest colony of the weakest empire is selected, and then an empire is selected using rank selection operator. Rank selection is in the way that all empires are ranked based on the empires power, according to equation 8.

$$(Mux - Cost_i) + Cr_{1 \leq i \leq n} \quad (8)$$

where  $Cr_i$  is the rank of empire *i*, *Mux* is maximum cost or minimum power and  $Cost_i$  is the cost or power of empire *i*. The best empire receives the rank of *Mux-Cost+1* and the worst empire receives the rank of 1. So, in this selection method all of the empires have a chance for selection. Then the weak colony will be allocated to the empire and the power of the weak empire that is selected will be re-computed. If the weak empire has no colony, the weak empire will be eliminated.

### 3.4 The online phase

After completion of the offline phase, and the implementation of the clustering algorithm as well as determining the best cluster using the imperialist competitive and firefly algorithms, in this section of online phase, the recommender system is able to recommend the best items to each user as soon as the user log into the system. This phase of the recommender system is composed of two steps: The first step is to choose the best cluster for an active user; the second step is to predict items. We will explain these two steps of the online phase in the following.

### 3.4.1 Choosing the best cluster

In the first step, as soon as the active user logs into the system, the list of items that the user has already rated will be fetched from the database system. Then the best cluster among the performed clustering will be selected for the active user. The best cluster for the active user is the cluster with a cluster head that has the most similarity with the items that the active user rated, compared to other cluster heads. This similarity is obtained based on the equation 3-1. It should be noted that the items are in a cluster as well as the users rated each item are clear. After the end of this step the best items of this cluster should be recommended to the active user.

### 3.4.2 The prediction of the active users' items

Predicting the rank of the active users' items is as follows:

1. First, the closest cluster to the active user item that should be predicted is selected based on the similarity equation 1-3.
2. Then K nearest neighbors to the item whose rank should be predicted, is selected. The appropriate K value will have a direct impact on the accuracy of the prediction.
3. The average of ranks that each item has earned will be calculated.
4. Finally, total average of total rates which are obtained from K items in the previous step will be calculated and divided by k; therefore the final average is obtained that its value is equal to the predicted value for the active user item.

## 4 Experimental results

For a more detailed evaluation and comparison of the proposed method, we utilized the real data sets Movie Lens 1M and Film Trust which are available on the Lens group and Trust web sites. Movie lens database is a reference in recommender systems research over the past few years. Every week thousands of users visit the Movie Lens website to rate movies and receive movie recommendations. This site has over 6040 users who have expressed their opinions on the 3706 film. This site contains 1000209 rates. Also the Film trust database contains 1508 users who gave rate for 1000 items. The number of users' votes is 35497. The value of each vote is in the range of 1 to 5. The higher the value is, the greater the user's interest for the movie is. This data set was provided by Guo, G. And Zhang for recommender systems for Trust site in 2011. After providing the data test the proposed ICAFA algorithm has been compared to GA genetic and ICA imperialist competitive algorithms based on various criteria. To compare each of these algorithms the C# programming language is used. The algorithm is executed on the Intel 3.00GHz processor and 4 GB of RAM. To predict, firstly 80 percent of Movie lens and Film Trust test data

**Table 3** the proposed algorithm parameters

values	The proposed method	imperialist competitive	Genetic	parameters
100	✓	✓	✓	The initial population size
5	✓	✓		The number of initial empires
6000-7000	✓	✓	✓	maximum number of iterations
70	✓	✓		Absorption rate
40	✓	✓		Rate of revolution
100	✓			The number of iteration of firefly algorithm
30	✓			Absorption Rate of Firefly
5	✓	✓		Imperialist competitive rates
60			✓	Mutation rate
70			✓	intersection rate

were selected as training test data and then clustering phase have been implemented for them. The remaining 20 percent of the data has been selected as test. The parameters of compared algorithms for clustering are shown in table 3. In table 3 for Movie Lens and Film trust test data repeat up to 7000 and 6000 is considered respectively. In the following, we'll show simulation results. At first the results will be evaluated in the offline phase, and then the prediction and the accuracy of the algorithms will be evaluated for active users in the prediction online phase.

#### 4.1 Clustering runtime results

After running three compared algorithms for each data set, the time taken to obtain the best answer which shows the best clustering for each algorithm, is achieved in a time in seconds. In other words, we have evaluated algorithms based on the best time to achieve the optimal solution. Figure 7 shows the run-time chart for each of the data set. Obviously the proposed clustering algorithm reduced the clustering time in the online phase. It is always a low complexity algorithm. In this section, the imperialist competitive algorithm has required more time than the genetic algorithm; since the imperialist competitive algorithm alone, doesn't have a proper convergence. The acceptable answer of the imperialist competitive algorithm is provided in more time than the genetic algorithm. The proposed method combines imperialist competitive algorithm with the firefly algorithm, this combination always results in maintaining population diversity and achieving an acceptable answer in a low time.

##### 4.1.1 Convergence results in the clustering algorithm

Figure 8 shows the convergence graph of the compared algorithms for Film Trust and Movie Lens data set. In the Film Trust plot, the proposed algorithm has the best answer in the beginning and in the early iterations in terms

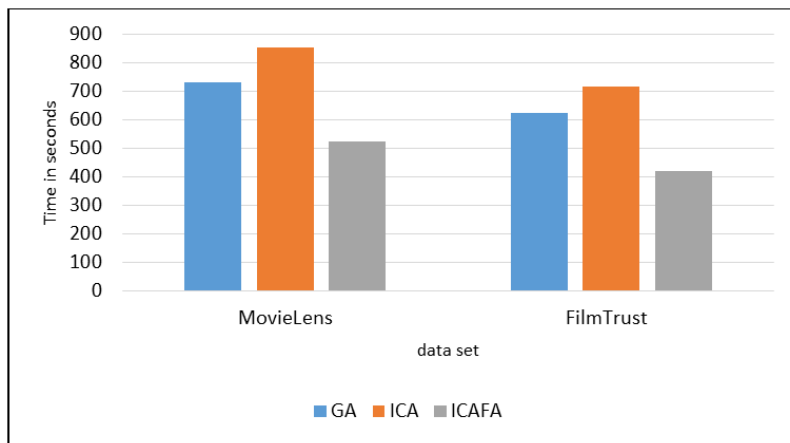


Fig. 7 runtime to compared algorithms

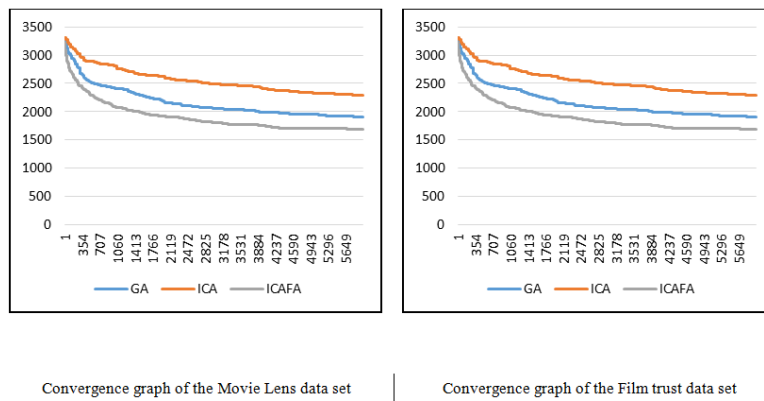


Fig. 8 Convergence graph of the compared algorithms

of objective function in comparison with two compared algorithms. Until in the final iteration the proposed method achieved much better answer than the two compared algorithms. In the Movie Lens plot, the genetic algorithm outperformed the proposed method in early iterations and finally in the final iteration the proposed algorithm achieved a better answer; because there are various update operators and detailed search problem space using the proposed algorithm. Also in the proposed method, the firefly algorithm always improved the population in the imperialist competitive algorithm and it achieved a good convergence.

**Table 4** the results from each algorithm based on the clusters' internal distance criterion.

Data set	Evaluation of parameters	GA	ICA	ICAFA
film Trust	best	1830.302	2310.244	1680.690
	worst	2260.353	2889.543	1620.643
Movie lens	best	2784.354	2426.178	2410.352
	worst	3265.435	3892.981	2765.932

## 4.2 Accuracy Results of Clustering

This section compares the algorithms based on the clustering accuracy. A clustering algorithm should be done so that the members of the cluster have the highest similarity. This process makes the similar data or items to locate within a cluster. Also, it makes the prediction of every item for each active user to be better. Table 4 shows the results. These results, which are based on the best and the worst possible answers, are shown to be able to measure the stability of algorithms relative to each other. It is obvious that the proposed algorithm do the clustering better than other algorithms. Also the differences obtained from the best and the worst answers show that the proposed algorithm has better stability and reliability and the confidence of the algorithm is always better than the other two algorithms.

## 4.3 Prediction accuracy measure

This criterion is based on the accuracy. It calculates the distance between the predicted and the actual ratings based on the equation 9 where  $r_i$  and  $\hat{r}_i$  are the actual and the predicted rate, respectively. The lower MAE value results better performance of the algorithm, because of less distance between the actual and the predicted ratings.

$$MAE = \frac{1}{n} \sum_{i=1}^n (r_i - \hat{r}_i) \quad (9)$$

We define the following scenarios to obtain the prediction accuracy of the compared algorithms:

- The first scenario

80% of data set sets Movie lens and Film Trust are selected to learn, and are clustered using the algorithm. Then, we choose 20% of the remaining data for testing. In this 20% of the data, some rates for items should be predicted. After the execution of the compared algorithms and the predicted rates, it becomes clear which algorithm gained the least prediction average error. The



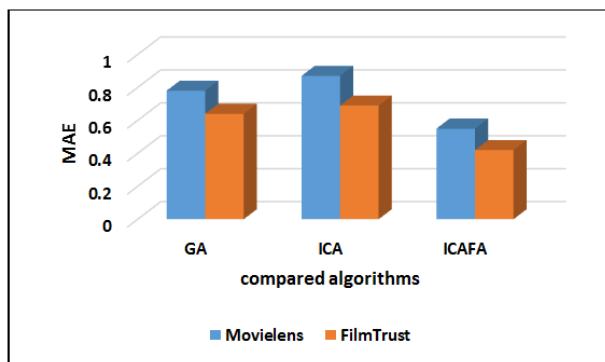


Fig. 9 the mean error obtained for each data set

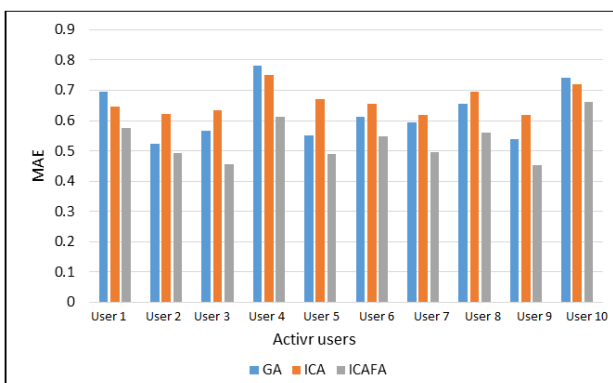


Fig. 10 the prediction error related to Movie lens data set for 10 active users

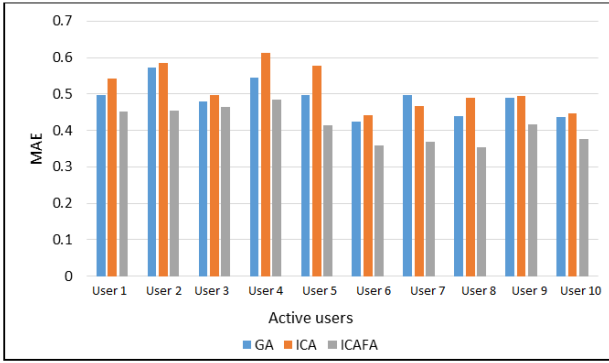
calculation of the prediction error is based on equation (1). Figure 9 shows the mean error obtained for testing data.

As is clear from the above, the proposed algorithm did better prediction than the compared genetic algorithm and gained lower error rate.

– The second scenario

In this scenario, 80% of the data has been chosen again as learning, Then 10 users are considered to be active users. After predicting and calculating equation (1), it has been found that how prediction accuracy for each active user was. Figure 10 illustrates the prediction error related to Movie lens data set for 10 active users.

As is clear from figure 10, the proposed algorithm was able to recommend items which are more likely to be interested by users, to active users. The proposed algorithm has a less percentage of error than genetic algorithm. In this chart the imperialist competitive algorithm alone is not able to show a good performance. It is only able to predict better than genetic for users 1 and 4. Figure 11 shows the absolute error obtained of 10 active users related



**Fig. 11** the prediction error on Film Trust data set for 10 active users

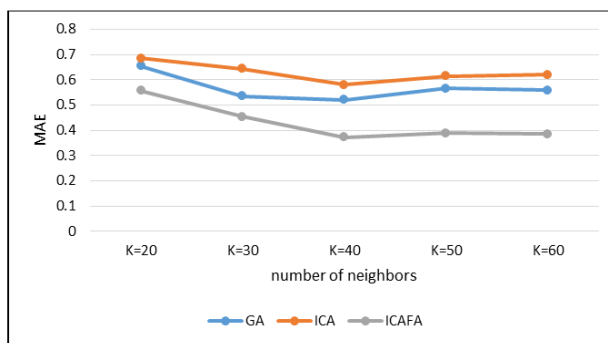
to Film Trust data set. In this chart, the imperialist competitive algorithm outperformed the genetic algorithm for active user 7. The proposed algorithm always is able to do a better prediction for active users, because of the clustering performed in the proposed method. The proposed hybrid algorithm always maintains population diversity; therefore, it can achieve an acceptable answer.

– The third scenario

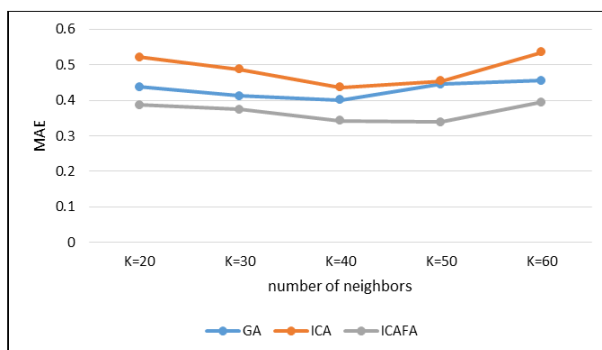
In this scenario we measure the algorithms based on the changes in the neighborhood. For instance, as stated in the third chapter, when an active user log into the site, first, the algorithm selects the best cluster according to the most similar cluster to the active user. Then  $K$  neighbors that are the most similar to the active user are selected. The items that will be recommended to the user are derived based on the prediction value of these neighbor items. For a better understanding, assume that you are supposed to obtain the predicted value item  $i$  for the active user that The predicted value is obtained based on the selected neighbors' items. The predicted value is obtained based on the average of predictions. For a detailed evaluation, we have implemented the data set with the compared algorithms 5 times; at any time the  $k$  value was equal to 20, 30, 40, 50, and 60 respectively. So we can conclude that what the effect of changing neighbors is on the prediction. Figure 12 shows the absolute accuracy of the prediction error to the data set Movie lens.

It can be seen from figure 12 that prediction accuracy will be changed by changing the number of neighbors. According to the results when the  $K$  value is equal to 40, better results can be achieved. In this graph, the proposed algorithm always is able to do a better prediction than two other algorithms. Figure 13 shows the accuracy of the absolute prediction error for the Film Trust data set.

In this graph, the proposed algorithm is more accurate than others on any change in the number of neighbors. The genetic algorithm is better than the imperialist competitive algorithm on clustering to ensure the high prediction accuracy. The results showed that if the algorithm can do a better clustering



**Fig. 12** the absolute accuracy of the prediction error to the data set Movie lens



**Fig. 13** the accuracy of the absolute prediction error for the Film Trust data set

and locate the similar data in one group, then the prediction accuracy will increase. Because the item whose prediction rating is supposed to be calculated, the closer to the neighbor is, the better it is.

## 5 Conclusions

In this paper we have presented a new meta-heuristic algorithm based on the combination of imperialist competitive and firefly algorithms, in recommender systems according to the clustering technique. The proposed method emphasizes the improvement of prediction accuracy, the scalability as well as achieving an appropriate answer in a reasonable time. The simulation results for the actual data set have showed that utilizing hybrid approaches and various operators to update the solutions will increase the search speed. Also utilizing the clustering technique in the proposed method, the best data in terms of the most similarity are in one cluster; as well as the favorite items of each active user are predicted. In other words, if we clustered data with a

high quality so that the data within each cluster have the most similarity, we could have a good recommender system.

## References

1. N. L. M. Shuib, N. Baiti, A. B. Normadhi, L. F. H. B. M. Alias, and N. S. Binti, "Collaborative recommender system: A review," in *International Conference On Advances In Computing, Electronics, And Electrical Technology*, (2015).
2. T.-P. Liang, H.-J. Lai, and Y.-C. Ku, "Personalized content recommendation and user satisfaction: Theoretical synthesis and empirical findings," *Journal of Management Information Systems*, vol. 23, no. 3, pp. 45–70, (2006).
3. J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: a survey," *Decision Support Systems*, vol. 74, pp. 12–32, (2015).
4. O. Khalid, M. U. S. Khan, S. U. Khan, and A. Y. Zomaya, "Omnisuggest: A ubiquitous cloud-based context-aware recommendation system for mobile social networks," *IEEE Transactions on Services Computing*, vol. 7, no. 3, pp. 401–414, (2014).
5. M. Sharma and S. Mann, "A survey of recommender systems: approaches and limitations," *International Journal of Innovations in Engineering and Technology*, vol. 2, no. 2, pp. 8–14, (2013).
6. M. J. Pazzani and D. Billsus, "Content-based recommendation systems," in *The adaptive web*. Springer, (2007), pp. 325–341.
7. X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in artificial intelligence*, vol. 2009, p. 4, (2009).
8. S. Trewin, "Knowledge-based recommender systems," *Encyclopedia of library and information science*, vol. 69, no. Supplement 32, p. 180, (2000).
9. S. Ujjin and P. J. Bentley, "Particle swarm optimization recommender system," in *Swarm Intelligence Symposium, 2003. SIS'03. Proceedings of the 2003 IEEE*. IEEE, (2003), pp. 124–131.
10. F. Lorenzi<sup>12</sup>, D. S. dos Santos, and A. L. Bazzan, "Case-based recommender system inspired by social insects," (2005).
11. J. Sobecki, "Web-based system user interface hybrid recommendation using ant colony metaphor," in *International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*. Springer, (2007), pp. 1033–1040.
12. M. Y. H. Al-Shamri and K. K. Bharadwaj, "Fuzzy-genetic approach to recommender systems based on a novel hybrid user model," *Expert systems with applications*, vol. 35, no. 3, pp. 1386–1399, (2008).
13. J. Handl and B. Meyer, "Ant-based and swarm-based clustering," *Swarm Intelligence*, vol. 1, no. 2, pp. 95–113, (2007).
14. J. Bobadilla, F. Ortega, A. Hernando, and J. Alcalá, "Improving collaborative filtering recommender system results and performance using genetic algorithms," *Knowledge-based systems*, vol. 24, no. 8, pp. 1310–1316, (2011).
15. M. Salehi, M. Pourzaferani, and S. A. Razavi, "Hybrid attribute-based recommender system for learning material using genetic algorithm and a multidimensional information model," *Egyptian Informatics Journal*, vol. 14, no. 1, pp. 67–78, (2013).
16. N. Badhe, D. Mishra, C. Joshi, and N. Shukla, "Recommender system for music data using genetic algorithm," *Int J Innov Adv Comput Sci*, vol. 3, no. 2, pp. 66–69, (2014).
17. A. Shrivastava and S. Rajawat, "An implementation of hybrid genetic algorithm for clustering based data for web recommendation system," *Int J Comput Sci Eng*, vol. 2, no. 4, pp. 6–11, (2014).
18. Z. Zhang, H. Lin, K. Liu, D. Wu, G. Zhang, and J. Lu, "A hybrid fuzzy-based personalized recommender system for telecom products/services," *Information Sciences*, vol. 235, pp. 117–129, (2013).
19. J. Lu, Q. Shambour, Y. Xu, Q. Lin, and G. Zhang, "A web-based personalized business partner recommendation system using fuzzy semantic techniques," *Computational Intelligence*, vol. 29, no. 1, pp. 37–69, (2013).

- 
20. L.-C. Cheng and H.-A. Wang, "A fuzzy recommender system based on the integration of subjective preferences and objective information," *Applied Soft Computing*, vol. 18, pp. 290–301, (2014).
  21. C. Porcel, A. G. López-Herrera, and E. Herrera-Viedma, "A recommender system for research resources based on fuzzy linguistic modeling," *Expert Systems with Applications*, vol. 36, no. 3, pp. 5173–5183, (2009).
  22. E. Atashpaz-Gargari and C. Lucas, "Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition," in *2007 IEEE Congress on Evolutionary Computation*. IEEE, (2007), pp. 4661–4667.
  23. X.-S. Yang, "Firefly algorithm," *Engineering Optimization*, pp. 221–230, (2010).