



A many objective commercial recommendation algorithm via Game-Based core node extraction

Yifei Sun^{a,*}, Yifei Cao^a, Shi Cheng^b, Jie Yang^a, Wenya Shi^a, Ao Zhang^a, Jiale Ju^a

^a School of Physics and Information Technology, Shaanxi Normal University, Xi'an, China

^b School of Computer Science, Shaanxi Normal University, Xi'an, China

ARTICLE INFO

Keywords:

Many objective optimization problem
Game theory
Core node
Recommendation system

ABSTRACT

The development of the recommendation system (RS) has been focused on improving the accuracy of algorithms. As the number of new users grows, RS suffers from the cold start problem, data sparsity and long-tail problem. In this paper, we proposed a many objective commercial recommendation algorithm that extracts the core node through the pay-off function of game theory and makes it guide the evolutionary computation. Firstly, the core aim of commercial recommendations is to maximize total profit. Profits can vary wildly with the same accuracy. Thus, the profit value as one of the goals is considered in attention to accuracy, coverage, and novelty. Besides, the current user state value is updated according to user features and rating scores matrix with the attention mechanism at different time stamps. The nodes are divided into different communities, forming a dynamic community, and recommendations are made within each community. Meanwhile, to reduce the computational complexity and runtime, the pay-off function of game theory is used to extract the core user in each community. Each contribution value and user interaction as criteria to guide the current community of users. Experiments on real datasets show that selecting core nodes can improve the performance of business RS. Compared with other existing algorithms in many experiments, our algorithm performs better.

1. Introduction

Information overload has become a pervasive problem with the development of Internet technology. In order to help people accurately find the information they need and are interested in, theoretical research and practical applications of RS have continued [1]. Back in October 2006, Netflix, the online DVD rental company, announced the Netflix Prize, which challenged researchers to develop RS that could beat the company's RS [2]. Recently, RS has played a significant role in e-commerce, e.g., Taobao, Tmall, Amazon, JD, and IKEA. Except in e-commerce, the recommender system is becoming increasingly popular in many areas. For instance, Steam recommends based on historical favorite games or a user's game styles. Social websites push some potential friends and link inferences [3,4], and e-learning educational institutions assist students in choosing courses and learning activities [5].

RS has three main factors: users, items and feedback. A unique identifier and some corresponding characteristics represent each user and item. Most user characteristics are generic, such as age, gender, education, profession, and nationality. In the case of movies, item characteristics are mostly domain-specific, such as title, genre, year,

director, actors, budget, or nominations. The main task for RS is getting feedback and some knowledge to learn a model. Types of feedback consist of explicit and implicit feedback. Since explicit feedback is often used as a number to evaluate a project, it needs to be more accurate to express their views on the project and also easy to impose a certain burden on users [6]. Implicit feedback recorded only the presence of the user's past interactions with the recommender system [7]. Implicit feedback is easier to collect than explicit feedback at a cost that is less reliable than explicit feedback. Like regression and classification machine learning tasks, two main recommendation tasks can be distinguished: rating prediction and item recommendation [8]. The rating prediction model aims to predict users' ratings on items, deals with explicit feedback, and recommends items with the top-n highest predicted rankings. Item recommendation aims to provide a personalized ranking of items to the user. A higher ranking score means a higher likelihood of a user's preference for an item. Item recommendation is usually related to implicit feedback.

Research on RS can be regarded as an optimization task, which selects some models and optimizes their parameters according to some quality standards. The recommendation model reflects user preferences

* Corresponding author.

E-mail address: yifeis@snnu.edu.cn (Y. Sun).

for different features or dimensions of the project, which can be explicit or potential variables [9]. Based on the model type and the information used in the learning, we can broadly divide the recommender system into content-based, collaborative filtering, and hybrid recommendation algorithms. Content-based recommendation algorithms [10] get the interest of users from the attributes of the project that the user has provided feedback on. Since the content-based recommendation algorithm must extract meaningful features, it avoids cold start and sparse data problems. In collaborative filtering [11], input from other users besides user and item attributes is considered. Singh et al. have shown that the degree of clustering can affect the quality of prediction of collaborative filtering systems [12]. Therefore, RS based on evolutionary clustering has received much attention recently. A hybrid algorithm is produced by combining collaborative and content-based methods. Note that the hybrid algorithm is created using a primary weighted linear aggregation method. As a result, the weight parameter should be appropriately tuned to overcome the shortcomings of a single model.

As real-world problems are often accompanied by more complex structured goals and constraints than before, classical optimization techniques need to be improved in dealing with the recommendation requirements. Evolutionary computing is widely used because it is powerful, efficient, and flexible. It can divide users or items into different communities and then generate recommendation lists in the same community, reducing the time complexity of algorithms and improving prediction accuracy. Evolutionary computing can also establish the RS as a many objective problem while optimizing more than three conflicting objectives. The performance of recommendations in the real world is unable to be expressed by accuracy alone but also depends on many subjective, user-perceived criteria. Zuo et al. proposed a multi objective recommendation algorithm combining a multi objective evolutionary algorithm (MOEA) and ProBS to co-optimize three objectives [13]. The experimental results proved the effectiveness of their method in improving diversity with high accuracy. Wei et al. presented a hybrid probabilistic MOEA (PMOEA) based on NSGA-II and SMS-EMOA [14]. This method pays attention to the properties of items observed from a real-world dataset. Chen et al. focused on using evolution clustering on the problem of algorithm clusters for RS. Specifically, the recommended time is significantly reduced by applying dynamic evolution clustering and a double-layer network to divide users and items into several categories. Then, they carry out collaborative filtering recommendations within the clustering [15]. Despite the great success of evolutionary computing in RS, the existing algorithms have never considered personalized recommendation as a dynamic social network. The interests of users do change over time and the diverse needs still need to be met. Once we consider the time-related information, the most challenging problem in RS is identifying the core nodes in the dynamic social networks. Trying to optimize more than three objectives to recommend is also an important topic in the field of evolutionary computing in recommender systems. To address the above challenges, a Many Objective Recommendation System with the dynamic evolutionary Game-Based Core Node (*MaORS-GBCN*) is proposed for a commercial recommendation system. The main contributions are as follows:

- **A Comprehensive Commercial Recommendation Model:** This many objective commercial recommendation algorithm can simultaneously optimize the accuracy, coverage, novelty, and total profit. Not only can provide personalized recommendation for users, but also consider the profit of the owner of the recommender system.
- **Updating User Interest and Considering historical information:** The forgetting function is used to describe the change in users' interest and update the state value of users. The current status values of users are determined jointly by historical status values, thus dynamic the user community.

- **The Concept of Game-Based Core Node:** Since the influence degree of each user is different in the network, the payoff function of game theory is used to identify the core nodes. The selected core nodes are used to guide the evolution of other nodes within the community to reduce recommendation time.

The rest of this paper is organized as follows: Section 2 reviews the related work. In Section 3, the proposed algorithm is introduced. Then, Section 4 provides the experimental results. Eventually, this paper and future work is summarized in Section 5.

2. Background

This section briefly introduces some related work and summarizes three main aspects: many objective optimization, forgetting score matrix and core user extraction.

2.1. Many objective optimization

Optimization problems are one of the primary forms of problems in engineering practice and scientific research. Optimization problems with more than three conflicting objective functions that require simultaneous processing are called many objective optimization problems (MaOPs). MaOPs can be defined as follows:

$$\min F(x) = (f_1(x), f_2(x), \dots, f_m(x)) \quad (m > 3) \quad (1)$$

$$\text{s.t.} \begin{cases} x \in \Omega \\ g_i(x) \leq 0, i = 1, \dots, k \\ h_j(x) = 0, i = 1, \dots, p \end{cases} \quad (2)$$

where $x = [x_1, x_2, \dots, x_d] \in \Omega$ is the decision vector, and Ω is the D -dimensional decision space. The decision vector x can be continuous or discrete. In our work, the discrete decision variable x stand for the unique identifier of an item. All items recommended to the user will be encoded on the chromosome using the item number as a matrix. As Fig. 1 shows, rows represent users, and columns represent items. $g_i(x)$ represents k inequality constraints, and $h_j(x)$ represents p equality constraints. The constraints in many objective RS are introduced in Section 3.3. At the same time, because the objectives conflict, increasing one goal means a decrease in the other. For two feasible solutions x_1 and x_2 , if $f_i(x_2) \geq f_i(x_1)$ for every $i \in \{1, \dots, m\}$ and $f_j(x_2) > f_j(x_1)$ for least one $j \in \{1, \dots, m\}$, x_1 is said to dominate x_2 . The solution, which is not dominated by any solution, is called Pareto optimal solution. Therefore, a MaOP aims to find a set of Pareto optimal solutions, called a Pareto-optimal set (PS). The Pareto front (PF) is the image of the Pareto set under the objective function space.

The MOEA based decomposition (MOEA/D), which decomposes many objective optimization problems into a certain number of single-objective optimization problems, is the basic idea for solving MaOP. Zhang et al. constructed MOEA/D, which combines the decomposition

	1	2	3	...	L
User 1	3	6	29	...	15
User 2	1	38	51	...	47
...
User N	9	22	35	...	88

Fig. 1. Illustration of chromosome encoding.

approach with evolutionary computation [16]. The algorithm successfully introduces the decomposition method commonly used in mathematical programming into the evolution of many objective fields and can directly use the evolutionary algorithm to solve the fitness allocation and diversity maintenance strategy of single objective optimization problems. In recent years, many researchers have drawn great attention to perfecting MOEA/D. The basic weight vectors may lead to many solutions assembling in boundaries of the discontinuous Pareto front while solving problems with discontinuous PF. Zhang and Gao et al. propose a mechanism for resetting weight vectors for MOEA/D to overcome this limitation [17]. Lei et al. focused on the hypervolume reference point specification and proposed a new *Hv*-based algorithm with two fixed reference points [18]. Li et al. introduced differential evolution operations into MOEA/D [19]. The experimental results indicate that combining mathematical programming methods with evolutionary algorithms is an effective method for solving many objective optimization problems. Our algorithm is also based on MOEA/D to optimize four objectives simultaneously in the recommendation system.

2.2. Forgetting score matrix

It is known that human attention is finite. The forgetting rule in RS can impact the next decision of users. The German Psychologist Ebbinghaus proposed the curve of forgetting control [20]. The significance of this function is that the process of forgetting is not uniform. There are four typical forgetting functions shown in Table 1. The forgetting speed is breakneck at first, then gradually slow down. So far, many efforts have been put into designing dynamic networks using a forgetting score matrix to obtain better recommendation performance.

For instance, Wang et al. proposed a personal recommender system. This model uses collaborative filtering to combine the interests of users into the newfound score matrix [21]. Besides, Chmiel et al. presented a forgetting function based on the Ebbinghaus forgetting curve of memory retention and Berlynes *inverted-U* model. Since the intermediate level of variables generated relatively high preference, the model handled well to present significant untapped data for music RS [22]. Wang et al. presented a novel method for providing recommendations by using an attention mechanism to learn the weight of each neighbor in the propagation process to distinguish the importance of neighbors called KGAT [23].

2.3. Core user extraction

In recent years, some methods of extracting core users have been introduced in designing recommendation algorithms. Information core refers to the user group carrying reliable and objective information about the recommender system. The information core extraction problem aims to select a group of users from all users who best represent the system. At present, core node extraction can be divided into the following categories: degree-based, resource-based, similarity-based, and multi-property. For example, Cao et al. used trust and similarity degrees between two users to extract information cores [24]. The recommended time is significantly reduced as the algorithm extracts core users. And then only these users will receive the resources from items.

Table 1
Forgetting curve and description.

Forgetting curve	Description
$f(t, i) = \frac{e^b}{(t_i + t_{min})^c}$	$b = 0.42, c = 0.0225$
$f(t, i) = \frac{10k}{\log(t_i + t_{min})^c + k}$	$k = 1.84, c = 1.25$
$f(t, i) = e^{\frac{t_i - t_{min}}{t_{max} - t_{min}} - 1}$	forget index function
$f(t, i) = (1 - \omega) + \omega \frac{ t_i - t_{min} ^2}{t_{max} - t_{min}}$	ω is forgetting factor, $0 \leq \omega \leq 1$

The performance of their method showed the core-users-based recommendation had less computation compared with other models. After that, Ren et al. proposed an extracting information core algorithm using a multi objective evolutionary algorithm [25]. The information core extracted by their model can not only be helpful in providing accurate recommendations but also provide recommendations of good performance in terms of coverage and diversity criteria. Furthermore, Sun et al. studied how the core nodes in dynamic optimization ensure the smoothness of the community structure [26].

3. Proposed algorithm

The original movie dataset is regarded as a model of a complex network structure that changes over time. The current node state value is determined by the state value that precedes the node. According to the score rating and node status information, the complex network is divided into different sub-communities. Divide similar nodes of interest into the same sub-community. Many users of RS provide little information and need direct search targets. The payoff function of game theory is utilized to find core users in the dynamic community to boot these inactive users. Finally, the many objective evolutionary algorithm is used to get recommendation lists. The framework of our *MaORS-GBCN* is shown in Fig. 2.

In this section, we describe in detail the proposed algorithm. The user-item feature item and forgetting rules are explained in Section 3.1. Section 3.2 accounts for how the core node is defined. Four conflicting objectives and constraints are shown in Section 3.3. The details of genetic operators are introduced in Section 3.4.

3.1. Dynamic user community

In fact, during the Data pre-processing process, we found that different user characteristics can affect their movie preferences. Some user characteristics are strongly associated with the type of movie they like [27]. Children like cartoons, young people like action movies, and older adults like history movies. Men like action movies, and women like romance movies. Different occupations can also influence the preference for film. Multiple user characteristics together affect user preferences. Thus, we divided the dataset into communities based on the user characteristics provided by the dataset according to the following rules: The genders are divided into male and female. There are four age categories: younger than 18, 18 to 35, 35 to 56, and over 56. Occupations are divided into culture, leisure, and management. For each user, we assign a 9-dimensional 0–1 vector U_i . If the user has this property, set it to 1; otherwise, set it to 0. From this, the attribute similarity between users can be calculated as follows:

$$S_U^{attribute}(i, j) = \frac{1}{1 + \|U_i - U_j\|} \quad (3)$$

However, user attribute similarity is unable to be used exclusively to express a similarity between users. The final preferences are not only determined by user attributes but also influenced by rating scores. The rating similarity between users can be calculated as follows:

$$S_U^{rating}(i, j) = \frac{1}{1 + \|U_{ri} - U_{rj}\|} \quad (4)$$

where U_{ri} and U_{rj} are the item score vector that user i and user j score at the same time.

Finally, the parameter $\mu \in [0, 1]$ balances the effect of two similarity degrees by the normalization methos.

$$S_U = \mu \bullet S_U^{attribute} + (1 - \mu) \bullet S_U^{rating} \quad (5)$$

In addition, the MovieLens dataset also provides the items attribute. A 19-dimensional vector is constructed by filtering items for these attributes. The final similarly of item is balanced item attribute similarity

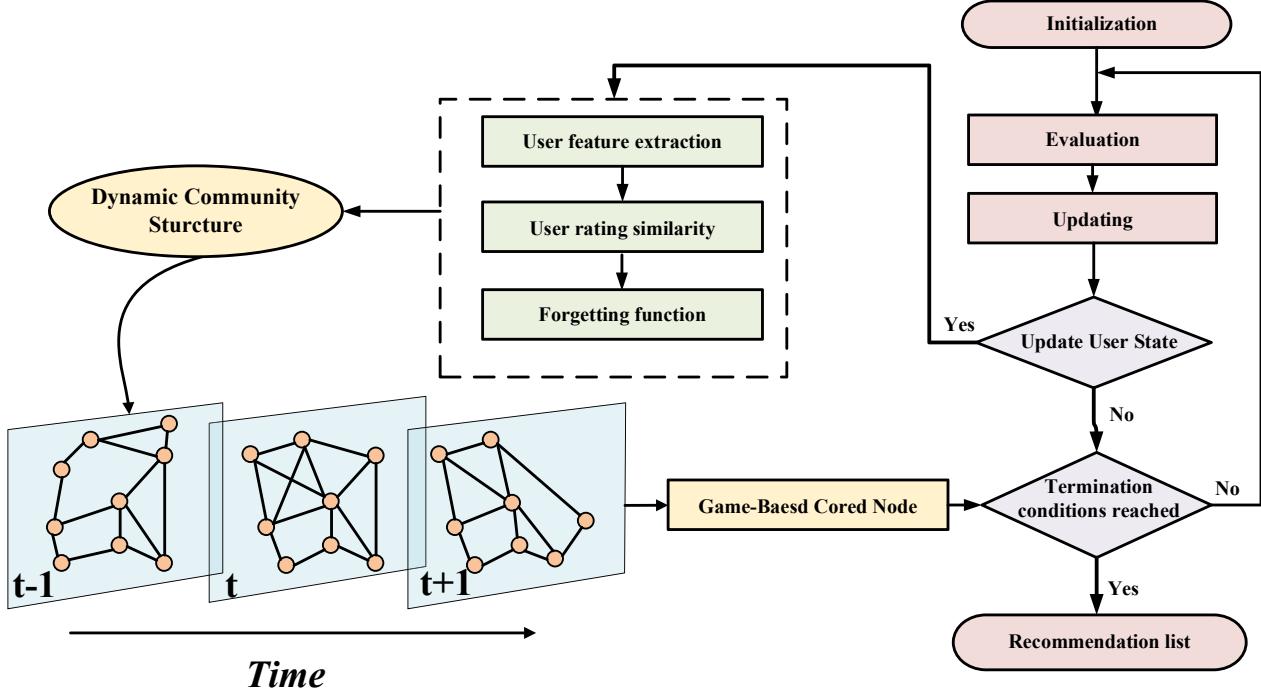


Fig. 2. The flowchart of proposed MaORS-GBCN.

and score similarity by the parameter $\lambda \in [0, 1]$. I_i denotes the attribute vector and I_{r_i} denotes the score vectors of items i evaluated by the users.

$$S_I = \lambda \frac{1}{1 + \|I_i - I_j\|} + (1 - \lambda) \frac{1}{1 + \|I_{r_i} - I_{r_j}\|} \quad (6)$$

The interest of each user will change over time. The closer the rating time is to the current time, the better the expression of user interest. In this paper, by fitting the Ebbinghaus forgetting curve, the forgetting function is put forward as follows:

$$f(t_{ui}) = a \cdot \exp\left(b \cdot \frac{t_{ui} - t_u^{\min}}{t_u^{\max} - t_u^{\min}}\right) + c \cdot \exp\left(d \cdot \frac{t_{ui} - t_u^{\min}}{t_u^{\max} - t_u^{\min}}\right) \quad (7)$$

where, the fitted coefficients are $a = 0.6368$, $b = -1.947$, $c = 0.3623$, and $d = 0.0025$, which have the same value as when they were presented. Additionally, $\exp(\cdot)$ is the exponential function. $f(t_{ui})$ represents the forgetting degree of u th user to the i th item. t_{ui} represents the time of the u th user rating for the i th item. t_u^{\min} represents the earliest rating time of user u . t_u^{\max} represents the latest rating time of user u .

From this, the current preference of users for the item is defined as follows:

$$A_U(u, i) = \frac{\exp(U_{r_u} \times f(t_{ui}))}{\sum_{n \in C_u} \exp(U_{r_n} \times f(t_{ni}))} \quad (8)$$

where, C_u represents the neighbor user set of user u .

Then, the user node state vectors $x_i(t)$ at time t is decided by the user node state vectors $x_i(t-1)$. $y_j(t)$ represents the state vector of item node j at time t . Finally, users are classified according to their final state using a dynamic clustering method. The experiment shows that the community partition can effectively reduce the running speed and improve the accuracy of RS.

$$\begin{aligned} x_i(t+1) = & x_i(t) + K_1 \cdot \sum_{j \in DU_i} S_{U_{ij}} \cdot \sin(x_j(t) - x_i(t)) + K_2 \cdot \sum_{j \notin DU_i} S_{U_{ij}} \\ & \cdot \sin(x_j(t) - x_i(t)) + K_3 \cdot \sum_{j \notin AU_i} A_{U_{ij}} \cdot \sin(y_j(t) - y_i(t)) + K_4 \\ & \cdot \sum_{j \notin AU_i} A_{U_{ij}} \cdot \sin(y_j(t) - y_i(t)) \end{aligned} \quad (9)$$

where the parameters K_1, K_2, K_3 and K_4 are set to 0.003, -0.002, 0.005 and -0.004.

3.2. Core node

Twitter statistics show that less than 1 % of active users contribute more than half of Twitter's content, with the rest performing far less well [28]. In addition, a smaller percentage of users write their thoughts on social networks, while more ordinary users pay attention to this information. As Fig. 3 shows, the steps to extract the core node are realized as follows: First, all users are divided into different communities based on the user node state. Second, we integrate the dynamic equations of game theory proposed by [29] to find the core nodes according to the user contribution value and the interaction between users. The final core

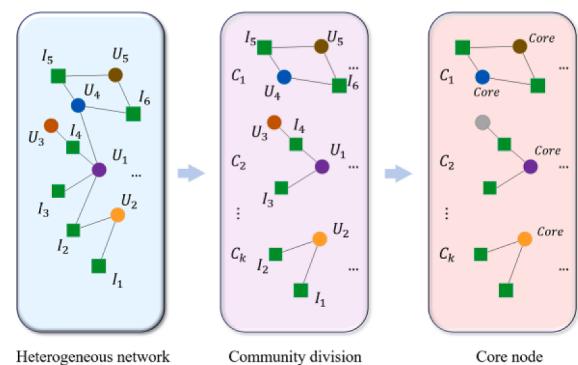


Fig. 3. The process of finding the core node.

node should be in the community center, interacting more with other users. The communities where the core nodes reside rarely change, which can better guide recommendations in a dynamic community. In this article, L is the Laplacian matrix in the community. The loss of contribution of user i to user j is expressed as follows:

$$r_{ij} = \alpha_i (L^{-1})_{ii} + \beta_j (L^{-1})_{jj} - K (L^{-1})_{ij} \quad (10)$$

where L^{-1} represents the generalized inverse matrix of L . α_i is the intrinsic marginal utility of each node. The interaction between i and j is computed by the $(L^{-1})_{ij}$. The rationale behind the approaches is that the core users in the community should have a relatively low contribution loss between core users and other users.

3.3. Objectives and constraints

During the training phase, the recommended items are ranked by predicting the item score due to the inability to get accurate user preferences. Without considering order, the top- n items form a recommendation list. The predicted rating for a recommendation list is defined as follows:

$$PR = \frac{\sum_{i \in S} \sum_{a=1}^L pr_{ia}}{|S| \times L} \quad (11)$$

where S is the user number in the cluster, L is the number of items to be recommended, pr_{ia} is the predicted rating for item i given by user a .

When the system recommends projects to users, just because the accuracy is high does not mean that the recommendation results are meaningful to the user [30]. Taking coverages as an indicator can represent the diversity of the recommendation list to explore the interest points of users. The ratio of items in the recommendation list with all available items is used as performance measures to measure the coverage of RS.

$$CV = \frac{N_{dif}}{N} \quad (12)$$

where N_{dif} is the number of different items in the recommendations lists for the users in the same cluster, and N is the total number of item.

Novelty measures the ability to recommend non-popular products to users, so high novelty can often lead to lower accuracy. A standard measure of novelty is to use the average of recommended items [31]. The lower average of items in a list means that the list is more novel to the user. Due to its rationality, self-information is used in this paper. The mean value of the self-information of items given by the RS is the novelty of the recommendation result, which is computed as follows:

$$Nov(N) = \frac{1}{NL} \sum_{u=1}^S \sum_{i \in O_L^u} \log_2 \left(\frac{N}{k_i} \right) \quad (13)$$

where k_i is the degree of the item i .

The total value of profits is proportional to sales and influenced by cost and other factors. A recommender list with a total profit of 20 may have the same accuracy as a recommendation list with a total profit of 50. In reality, films that cost too much are more likely to be in the red. In the training process, p_i represents the profit that item i can generate. The ratio of the predicted score $\widetilde{r}_{u,i}$ to the maximum predicted score \widetilde{r}_{max} to represent the probability that the item i will be received. The expected profit $p_{u,i}$ of item i to user u can be defined as:

$$p_{u,i} = \frac{\widetilde{r}_{u,i}}{\widetilde{r}_{max}} \times p_i \quad (14)$$

In addition, there are two primary constraints in commercial RS. A commercial RS should not have the same item appearing twice in a recommendation list, and items that users have already rated should not appear in the recommended list again. This means that duplicate

numbers will not appear on the same line in the chromosome.

3.4. Genetic operators

The extreme points in the prediction matrix play a positive effect on extending the PF. In this paper, all selectable items under each objective for each user are ranked. The top- n points are selected into an initialized individual, and the other initialized individuals are chosen randomly. Due to the constraint in RS, the uniform crossover operation is used to produce new solutions. The principle is that the identical items from two parents are propagated to the child directly. The remaining alleles are then crossed. We generate a random number from 0 to 1. The child 1 will get the allele from parent 1 if the random number is greater than 0.5. Otherwise, it receives the allele from parent 2. The detailed flow of the crossover and mutation operator is shown in Fig. 4.

The function of mutation in evolutionary algorithms is to keep population diversity and prevent the population from falling into local optimum. The Sigmoid function is improved to apply to many-objective genetic applications. The dominance hierarchy and crowding degree of the individual population are used to sort the individuals. Individuals with lower dominance hierarchies have lower mutation rates. γ_1 and γ_2 are used to determine the probability of mutation and the parameter D distinguish the fine and poor individuals. The mutation probability is shown as follows:

$$p_m^i = \begin{cases} \frac{0.5}{1 + e^{-\gamma_1(i-N_s)}}, & i < D \\ \frac{0.5}{1 + e^{-\gamma_2(i-N_s)}}, & i \geq D \end{cases} \quad (15)$$

where the parameters γ_1 , γ_2 and D are set to 0.2, 0.1, 0.4, respectively.

4. Experiments

In this paper, all the experiments are conducted in 2021a Matlab on an Intel(R) Corei5 computer with CPU(TM) i5-12500H and 2.50 GHz memory. In Section 4.2, we confirm the effectiveness of game-based core node guided on our evolutionary model. The parameter analysis is presented in Section 4.3. Section 4.4 explores the performance of our model with other compared algorithms.

4.1. Experiment Settings

4.1.1. Datasets

In our experiments, three benchmark datasets are used to test the effectiveness of the proposed recommendation algorithm. Table 2 summarizes the general statistics for these three datasets. The scores value in all three datasets ranged from 1 to 5. Like [32], we consider a binary likes and dislikes scoring system. An item is viewed as a user favorite if the user scores at least 3. Then, we randomly select 80 % of the data as the training set and the rest as the probe set. Due to the

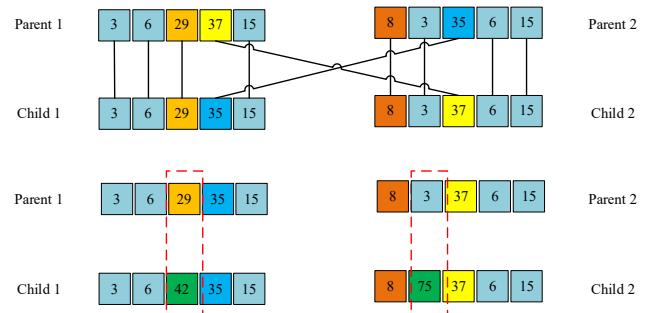


Fig. 4. Illustration of the crossover and mutation operator.

Table 2
Forgetting curve and description.

Parameter	user	item	Sparseness	Ratings
MovieLens-100 K	943	1682	93.71 %	100,000
MovieLens-1 M	6040	3952	95.81 %	1,000,209
CiaoDVD	7375	99,746	99.97 %	278,483

influence of the sparsity of the dataset, there are a large number of projects in probe sets that still can not know whether it is liked or not by users. To ensure the experiment's credibility, each user has rated at least 20 items, and each item is rated by at least 20 users selected to construct a new dataset.

4.1.2. Performance metrics

Here, four performance metrics are used in this paper. Precision is the most common metric for evaluating RS [33]. For a given user u , the precision is defined as the ratio of correctly recommended related items as follows:

$$P_u(L) = \frac{d_u(L)}{L} \quad (16)$$

where $d_u(L)$ is the number of related items in the recommended list and the test dataset preferred by user u . The obtained mean precision of all users can reveal the recommendation accuracy of RS. The higher $P_u(L)$ means the more accurate the recommendation model.

In this paper, coverage is one of the four conflicting objectives. The novelty measure for RS describes the ability of RS to recommend unknown items to users. The formula (11) and (12) are used as performance measures to measure the coverage and novelty.

Actual total profit P_{real} represent the total profit of the recommended items accepted by the user and measures the revenue from RS.

$$P_{real} = \sum_{u \in U} \sum_{i \in d_u} p_i \quad (17)$$

Besides, the hypervolume(Hv), also known as an *S metric* or *Lebesgue measure*, is adopted as the performance metric [34]. The Hv is the only known monadic metric that can evaluate the quality of a solution set. Hv is a comprehensive index to assess the convergence and diversity of approximate solution sets.

$$Hv(S) = \mathcal{L}(\bigcup_{x \in S} [f_1(X), z_1^r] \times \cdots \times [f_m(X), z_m^r]) \quad (18)$$

where \mathcal{L} is the Lebesgue measure, which is a standard method for assigning a length, area, or a volume to a subset of Euclidean space. $z^r = (z_1^r, \dots, z_m^r)$ represents a reference point in the objective space, which is dominated by all solutions in the true PF. A recent survey points out that the effect of the reference point on multi-objective problems with triangular Pareto fronts and two-objective minimization problems is small [18]. Thus, unless the reference point is too close to the nadir point, the experimental results are not sensitive to the reference point specification.

4.1.3. Compared algorithms

To verify the recommendation performance of *MaORS-GBCN*, the algorithm is compared with MOEA-EPG, MOEA-Probs, PMOEA, User-

based CF and Item-based-CF. Their parameter settings are set as suggested and are listed in Table 3, where L is the length of the recommendation list, N is the population size, P_m is the mutation probability, and P_c is the crossover probability. The parameter P_n is only used in PMOEA, representing the number of parents in a crossover operation. Parameter k used in MOEA-EPG and MOEA-Probs represents the ratio of the most similar users to the target users within a cluster and represent the ratio of the core node in *MaORS-GBCN*.

4.2. The effectiveness of Gamed-Based core node

In our experiments, the user state calculated from Formula 9 is the major determining factor. Users in MoiveLens-100 K and MoiveLens-1 M are divided into four clusters and three clusters in CiaoDVD. Fig. 6 shows the clustering results based on our dynamic network evolutionary clustering method on MoiveLens-100 K. Some users whose rating matrices are too sparse just cluster around the edge of the cluster. There are few connections between user communities, and users are clustered around the core nodes of their communities.

We designed a control experiment to verify the effectiveness of core node guidance. We set two comparison algorithms, which are the neighbor node guide with the highest similarity and the resistance distance-based core node guide. Because of the different community sizes, the number of core nodes set to a fixed value is unable to describe all cases. This article sets the number of core nodes as a ratio of the total number of users, which is set to 0.1. From [35], the computational complexities of all comparison recommendation algorithms can be easily deduced. The computational complexity is mainly determined by the evolutionary search. Calculating the objective values of the individuals in the population and selecting parent solutions by using the binary tournament take a computational complexity with $O(N \cdot m^2 \cdot n)$, where N is the population size, m is the user number of the community and n is the number of items. Moreover, the complexity of uniform crossover operation and mutation is $O(N \cdot m^2 \cdot n)$. Therefore, the complexity of overall computational complexity of *MaORS-GBCN* is $O(T \cdot N \cdot m^2 \cdot n)$. The complexity of obtaining core users is $O(kn^2)$, where n is the number of users and k is the number of core nodes as a ratio of the total number of users. Despite the fact that the core user needs to spend more time, once the core users are found, the computation time can be shortened by about 10 times. In practice, more than 90 % of core users do not need high-frequency updates. As time goes on, the reduced computational complexity will continue to accumulate.

Hv represents the range of many objective solutions in space. Hv comparison results of three core node algorithms are collected in Table 4, where the best results are highlighted in boldface. We can

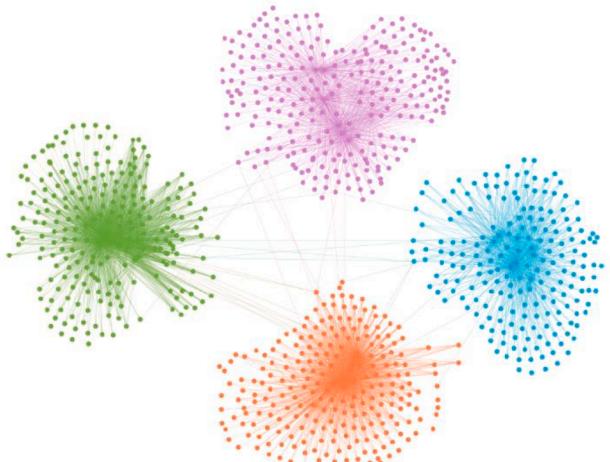


Fig. 6. User Clustering results based on MovieLens-100 k.

Table 3
Parameter settings of the algorithm.

Algorithm	Parameters Settings
User-based-CF	$L = 10$
Item-based-CF	$L = 10$
MOEA-Probs	$P_c = 0.8, P_m = 1/L, L = 10, N = 100, Gen = 3000, k = 0.9$
PMOEA	$P_n = 5, P_m = 1/L, L = 10, N = 100, Gen = 3000, n = 20$
MOEA-EPG	$P_c = 0.8, P_m = 1/L, L = 10, N = 100, Gen = 3000, k = 0.5$
MaORS-GBCN	$P_c = 0.8, L = 10, N = 100, Gen = 3000, k = 0.1$

Table 4

Hv Results of different core node algorithm.

Dataset		<i>neighbor node</i>	<i>distance-based core node</i>	<i>Game-based core node</i>
MoiveLens-100 K	Movielens1	6.7383e + 04	2.7040e + 05	3.6541e + 06
	Movielens2	9.8786e + 04	3.7485e + 05	3.7091e + 06
	Movielens3	7.3078e + 04	2.7186e + 05	4.0687e + 06
	Movielens4	1.8108e + 05	4.0413e + 05	6.2833e + 06
MoiveLens-1 M	Movielens5	1.8217e + 06	9.9962e + 06	1.7553e + 08
	Movielens6	3.4940e + 05	4.7967e + 05	2.4221e + 07
	Movielens7	2.9501e + 05	3.3356e + 05	1.2091e + 07
	Movielens8	1.3049e + 05	3.1376e + 05	1.2007e + 07
CiaoDVD	CiaoDVD1	6.2849e + 05	1.4642e + 06	1.4864e + 06
	CiaoDVD2	1.1685e + 06	2.7379e + 06	2.7650e + 06
	CiaoDVD3	2.1134e + 06	5.5246e + 06	5.4323e + 06

conclude from the experimental results the recommendation algorithms with the core node are better than those with the neighbor node guided. The game-based core node performs better than the distance-based core node in the MoiveLens. However, there is little difference in performance between the game-based core node and the distance-based core node in the CiaoDVD.

4.3. Parameter analysis

In our proposed algorithm, the selection of the core node directly affects the final experimental results. In formula (8), the core node is determined by three parameters, α , β and K . We experimented with the results in each dataset. We tested three parameters with a step size of 0.1, ranging from 0.1 to 0.9, while the other parameters remained the same as listed in Table 3. There are 9 subgraphs in Fig. 7, concerning the results for each dataset when the three parameters take different values without clustering. From the experimental results, the three parameters

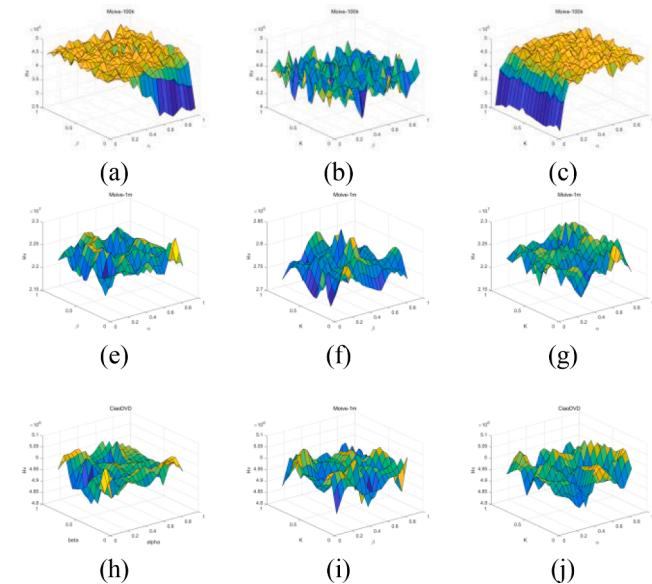


Fig. 7. Exploring the influence of different parameters K , α and β on prediction performance.

take different values for the experimental results to have a significant impact. In MoiveLens-1 m and CiaoDVD datasets, the influence of the three parameters fluctuates wildly. Still, in MovieLens-100 k, the value of Hv fluctuates significantly with the change of K and β when α is determined. Finally, according to the practical results chart, the parameters of each experimental dataset are selected for statistics in Table 5.

4.4. Compare with other algorithm

As displayed in Figs. 8 and 9, User-based-CF and Item-based-CF are dominated by those of MaORS-GBCN as they only consider one objective. In the experiment of this section, we carry on the community to all algorithms according to the dynamic clustering method considering the attention mechanism. According to Fig. 8, although the novelty value obtained by the model proposed in this paper is not the best, MaORS-GBCN can get a recommendation list with better accuracy than the other three many objective algorithms. MaORS-GBCN, compared with different algorithms, can be the best in terms of accuracy and comprehensive performance.

By comparing each solution in Fig. 8 and Fig. 9, profit is not necessarily proportional to accuracy and conflicts much with novelty. This effectively validates the value of our proposed model. The parallel coordinate plot is used to illustrate this conflicting relationship better [36]. In Fig. 10, the left coordinate represents the target of the expected total profit, and the right coordinate represents novelty. For each solution, we connect two target values with a straight line, so the number of intersections between these lines indicates the degree of conflict between the targets in MovieLens. Thus, Fig. 10 shows the degree of disagreement between the two goals in MovieLens-100 k, in which both goals are normalized. According to Fig. 9, MaORS-GBCN can always obtain a recommendation list with better total profit than the other three many objective algorithms under the same accuracy. Meanwhile, the existence of the core nodes ensures that MaORS-GBCN gets a wider PF than others. In general, the performance of MaORS-GBCN in all four evaluation indicators is good except for novelty.

All average Hv values in each clustering are collected in Table 6 for performance comparisons, where the best results are highlighted in boldface. Based on the experimental results of all the recommended datasets, MaORS-GBCN performs better than MOEA-Probs and PMOEA. Compared to MOEA-EPG, there is no apparent advantage in MovieLens-100 K, but there is no obvious defect. This further proves that under the guidance of the game-based core points, our algorithm can get a larger range in the search space even though it lacks novelty.

5. Conclusion

In this paper, a commercial RS model is established, and a new objective function is introduced to improve the total profit of the RS. Then, a dynamic community over time is built based on the changing interests of the users over time and the changing status values of the users. Then, the payoff function of game theory is used to extract the core nodes of each community and select the core nodes by considering the individual contributions of users and the roles of users. The core node can guide recommendations from other users within the community. Finally, the many objective evolutionary algorithm with adaptive strategy is used to get the recommendation lists. The MaORS-GBCN algorithm is compared with the existing many objective recommendation

Table 5
Hv Comparison Results of Parameter Analysis.

Dataset	K	α	β
MoiveLens-100 K	0.2	0.8	0.9
MoiveLens-1 M	0.6	0.6	0.7
CiaoDVD	0.3	0.7	0.6

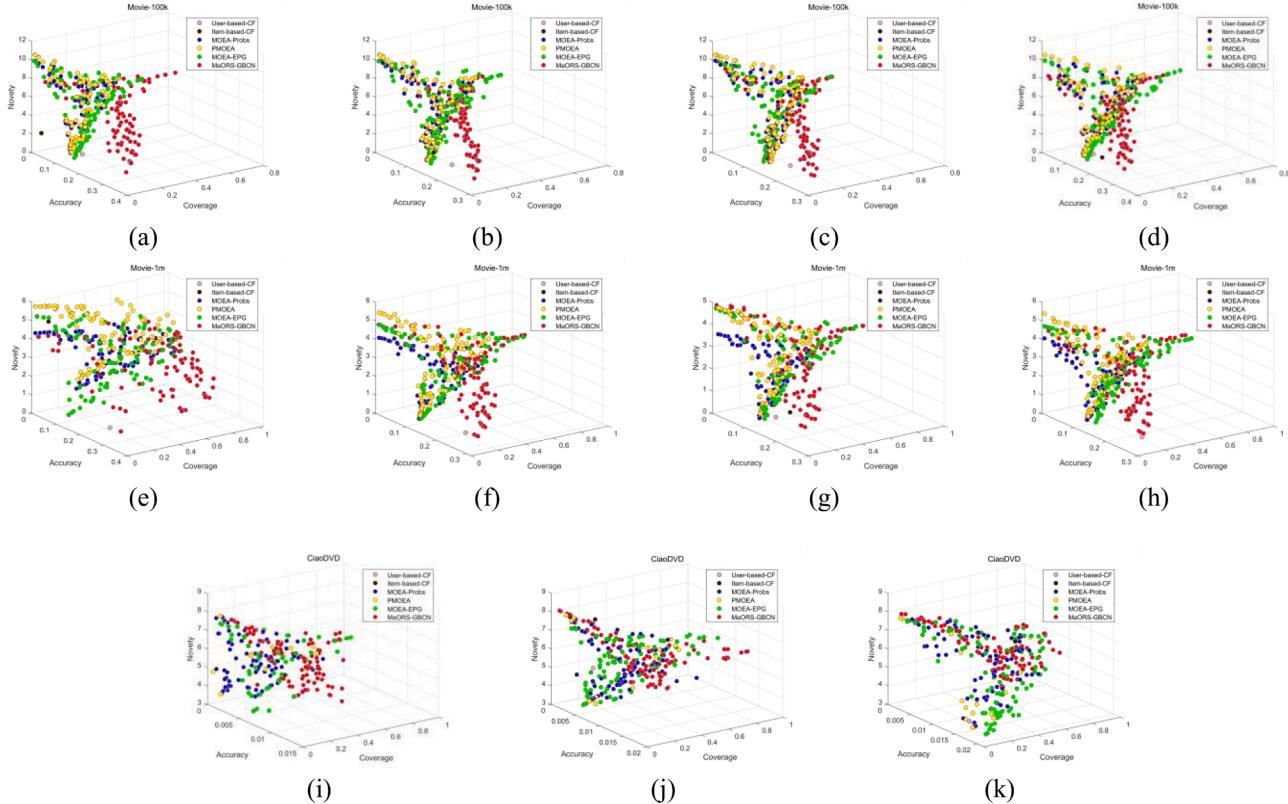


Fig. 8. Final recommendation results of comparison algorithms under the accuracy-coverage-novelty space.

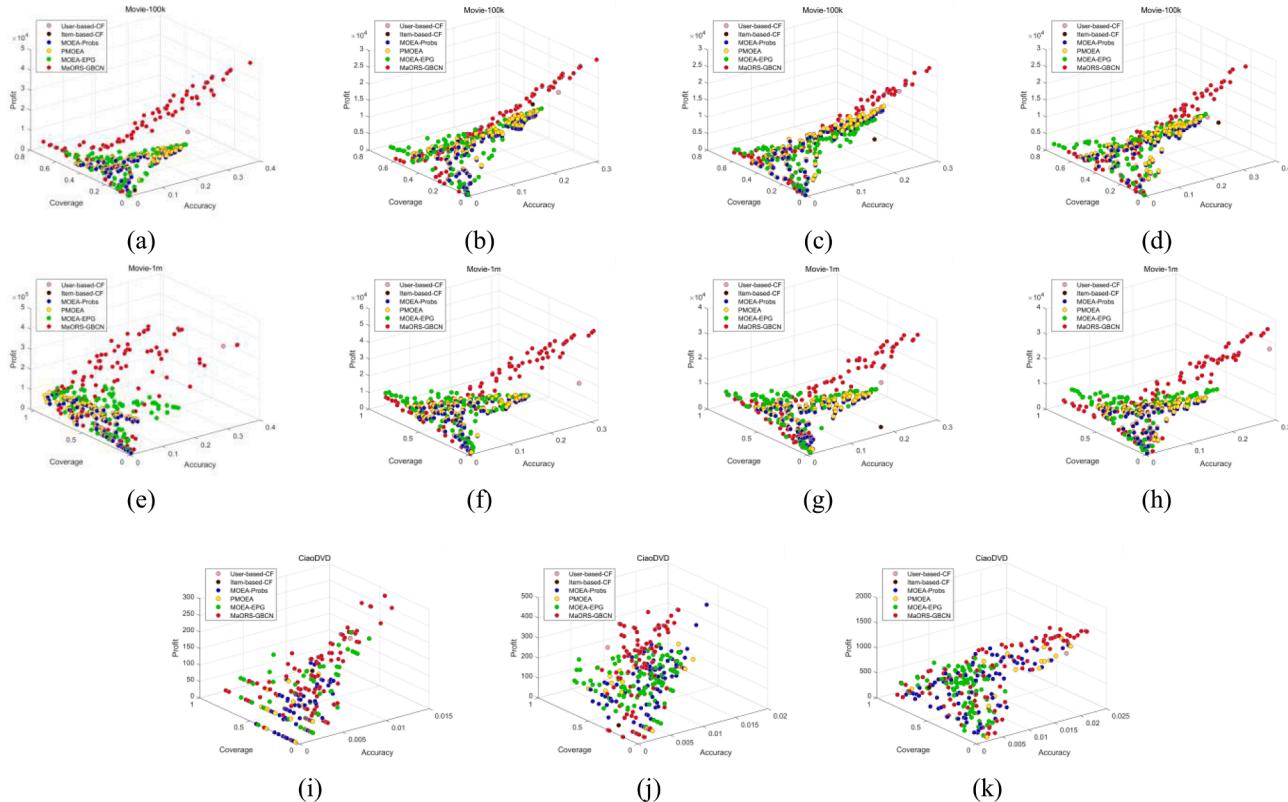


Fig. 9. Final recommendation results of comparison algorithms under the accuracy-coverage-profit space.

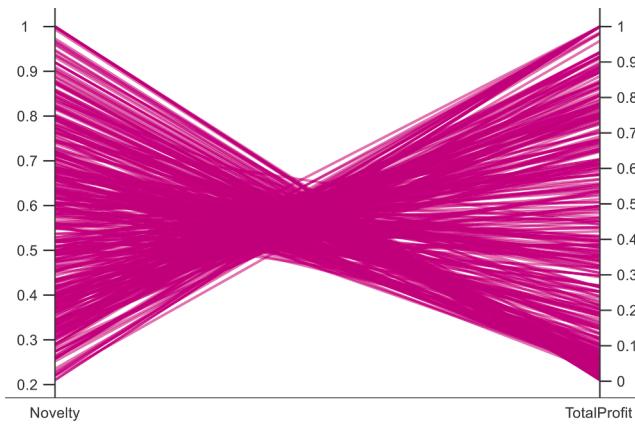


Fig. 10. Parallel coordinates for solutions of Movielens-100 k.

Table 6
H_v Comparison Results of Compared Algorithm.

Dataset		MOEA- Probs	PMOEA	MOEA- EPG	MAORS- GBCN
MoiveLens- 100 K	Movielens1	2.3562e + 06	2.0374e + 06	4.0417e + 06	3.6541e + 06
	Movielens2	3.3810e + 06	3.3513e + 06	4.3902e + 06	3.7091e + 06
	Movielens3	3.2751e + 06	3.4366e + 06	4.3521e + 06	4.0687e + 06
	Movielens4	4.4356e + 06	4.6152e + 06	5.6698e + 06	6.2833e + 06
MoiveLens- 1 M	Movielens5	6.7820e + 07	6.7412e + 07	1.5005e + 08	1.7553e + 08
	Movielens6	1.0362e + 07	1.0126e + 07	1.3484e + 07	2.4221e + 07
	Movielens7	7.6935e + 06	7.7412e + 06	1.0562e + 07	1.2091e + 07
	Movielens8	7.3269e + 06	7.6555e + 06	9.3540e + 06	1.2007e + 07
CiaoDVD	CiaoDVD1	2.5228e + 05	5.9962e + 05	6.2849e + 05	1.4864e + 06
	CiaoDVD2	4.8306e + 05	8.2998e + 05	1.1685e + 06	2.7650e + 06
	CiaoDVD3	1.3317e + 06	1.8882e + 06	2.1134e + 06	5.4323e + 06

system algorithms. The game-based core node is verified by comparing the neighbor node with the highest similarity and resistance distance-based core node guide. Experimental results show that our algorithm can always get a larger search range in the target space. However, the general forgetting rule is not suitable for all users. Repeated viewing and repeated rating of items by users are not included in the dataset. Our future work needs to be focused on ensuring that the recommendations are made in real time while considering the time-related information for re-viewing. In addition, machine learning can be used to train the parameters of our algorithm to find core users. The lower the update frequency of the core node, the greater the performance of the recommendation system.

CRediT authorship contribution statement

Yifei Sun: Supervision, Validation. **Yifei Cao:** Writing – original draft, Methodology, Conceptualization. **Shi Cheng:** Writing – review & editing. **Jie Yang:** Visualization. **Wenya Shi:** Data curation. **Ao Zhang:** Formal analysis. **Jiale Ju:** Data curation.

Declaration of Competing Interest

The authors declare that they have no known competing financial

interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment:

This work was supported by the Natural Science Basic Research Plan in Shaanxi Province of China (Program No. 2022JM-381, 2017JQ6070) National Natural Science Foundation of China (Grant No. 61703256), Foundation of State Key Laboratory of Public Big Data (No. PBD2022-08) and the Fundamental Research Funds for the Central Universities (Program No.GK202201014, GK202202003, GK201803020).

References

- [1] Lü L, Medo M, Yeung CH, Zhang Y-C, Zhang Z-K, Zhou T. Recommender systems. *Phys Rep Oct.* 2012;519(1):1–49. <https://doi.org/10.1016/j.physrep.2012.02.006>.
- [2] Hallinan B, Striphakis T. Recommended for you: the Netflix Prize and the production of algorithmic culture. *New Media Soc Jan.* 2016;18(1):117–37. <https://doi.org/10.1177/1461444814538646>.
- [3] Ma L, Li J, Lin Q, Gong M, Coello Coello CA, Ming Z. Reliable link inference for network data with community structures. *IEEE Trans Cybern* 2019;49(9):3347–61.
- [4] Schafer JB, Konstan JA, Riedl J. E-commerce recommendation applications. *Data Min Knowl Discov* 2001;5(1/2):115–53. <https://doi.org/10.1023/A:1009804230409>.
- [5] C. Li and L. Ma, "Item-based Collaborative Filtering Algorithm Based on Group Weighted Rating," in *2020 13th International Symposium on Computational Intelligence and Design (ISCID)*, Hangzhou, China: IEEE, Dec. 2020, pp. 114–117. doi: 10.1109/ISCID51228.2020.00032.
- [6] S. Pero and T. Horváth, "Opinion-Driven Matrix Factorization for Rating Prediction," in *User Modeling, Adaptation, and Personalization*, vol. 7899, S. Carberry, S. Weibelzahl, A. Micarelli, and G. Semeraro, Eds., in *Lecture Notes in Computer Science*, vol. 7899, , Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 1–13. doi: 10.1007/978-3-642-38844-6_1.
- [7] Kelly D, Teevan J. Implicit feedback for inferring user preference: a bibliography. *ACM SIGIR Forum Sep.* 2003;37(2):18–28. <https://doi.org/10.1145/959258.959260>.
- [8] Y. Baştanlar and M. Özysal, "Introduction to Machine Learning," in *miRNomics: MicroRNA Biology and Computational Analysis*, vol. 1107, M. Yousef and J. Almer, Eds., in *Methods in Molecular Biology*, vol. 1107. , Totowa, NJ: Humana Press, 2014, pp. 105–128. doi: 10.1007/978-1-62703-748-8_7.
- [9] Horváth T, De Carvalho ACPLF. Evolutionary computing in recommender systems: a review of recent research. *Nat Comput* 2017;16(3):441–62. <https://doi.org/10.1007/s11047-016-9540-y>.
- [10] Pazzani MJ, Billsus D. In: *Lecture Notes in Computer Science the Adaptive Web*. Berlin, Heidelberg: Springer Berlin Heidelberg; 2007. p. 325–41.
- [11] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. *Computer* 2009;42(8):30–7. <https://doi.org/10.1109/MC.2009.263>.
- [12] Singh M, Mehrotra M. Impact of clustering on quality of recommendation in cluster-based collaborative filtering: an empirical study. *Int J Bus Intell Data Min* 2020;17(2):206. <https://doi.org/10.1504/IJBIDM.2020.108774>.
- [13] Zuo Y, Gong M, Zeng J, Ma L, Jiao L. Personalized recommendation based on evolutionary multi-objective optimization [research frontier]. *IEEE Comput Intell Mag Feb.* 2015;10(1):52–62. <https://doi.org/10.1109/MCI.2014.2369894>.
- [14] Wei G, Wu Q, Zhou M. A hybrid probabilistic multiobjective evolutionary algorithm for commercial recommendation systems. *IEEE Trans Comput Soc Syst Jun.* 2021;8(3):589–98. <https://doi.org/10.1109/TCSS.2021.3055823>.
- [15] Chen J, Wang Z, Zhu T, Rosas FE. Recommendation algorithm in double-layer network based on vector dynamic evolution clustering and attention mechanism. *Complexity* 2020;2020:1–19. <https://doi.org/10.1155/2020/5206087>.
- [16] Zhang Q, Li H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans Evol Comput* 2007;11(6):712–31. <https://doi.org/10.1109/TEVC.2007.892759>.
- [17] Zhang C, Gao L, Li X, Shen W, Zhou J, Tan KC. Resetting weight vectors in MOEA/D for multiobjective optimization problems with discontinuous pareto front. *IEEE Trans Cybern* 2022;52(9):9770–83. <https://doi.org/10.1109/TCYB.2021.3062949>.
- [18] Pang LM, Ishibuchi H, He L, Shang K, Chen L. Hypervolume-Based Cooperative Coevolution with Two Reference Points for Multi-Objective Optimization. *IEEE Trans Evol Comput* 2023;1. <https://doi.org/10.1109/TEVC.2023.3287399>.
- [19] Gong M, Jiao L, Du H, Bi L. Multiobjective immune algorithm with nondominated neighbor-based selection. *Evol Comput* 2008;16(2):225–55. <https://doi.org/10.1162/evco.2008.16.2.225>.
- [20] Ebbinghaus H. Memory: a Contribution to Experimental Psychology. *ANS* 2013;20 (4).
- [21] Wang Z, Chen J, Rosas FE, Zhu T. A hypergraph-based framework for personalized recommendations via user preference and dynamics clustering. *Expert Syst Appl* 2022;204:117552. <https://doi.org/10.1016/j.eswa.2022.117552>.
- [22] Alayri F, Jafari Navimipour N. Recommender systems: A systematic review of the state of the art literature and suggestions for future research. *K* 2018;47(5):985–1017.
- [23] X. Wang, X. He, Y. Cao, M. Liu, and T.-S. Chua, "KGAT: Knowledge Graph Attention Network for Recommendation," in *Proceedings of the 25th ACM SIGKDD*

- International Conference on Knowledge Discovery & Data Mining*, Anchorage AK USA: ACM, Jul. 2019, pp. 950–958. doi: 10.1145/3292500.3330989.
- [24] G. Cao and L. Kuang, “Identifying Core Users Based on Trust Relationships and Interest Similarity in Recommender System,” in *2016 IEEE International Conference on Web Services (ICWS)*, San Francisco, CA, USA: IEEE, Jun. 2016, pp. 284–291. doi: 10.1109/ICWS.2016.44.
- [25] Z. Ren and J. Liu, “Extracting Information Cores with Multi-property Using a Multiobjective Evolutionary Algorithm,” in *2019 IEEE Congress on Evolutionary Computation (CEC)*, Wellington, New Zealand: IEEE, Jun. 2019, pp. 1014–1021. doi: 10.1109/CEC.2019.8790235.
- [26] Sun Y, Sun X, Liu Z, Cao Y, Yang J. Core node knowledge based multi-objective particle swarm optimization for dynamic community detection. *Comput Ind Eng* 2023;175:108843. <https://doi.org/10.1016/j.cie.2022.108843>.
- [27] Chen J, Lu Y, Shang F, Wang Y. A fuzzy matrix factor recommendation method with forgetting function and user features. *Appl Soft Comput* 2021;100:106910. <https://doi.org/10.1016/j.asoc.2020.106910>.
- [28] Li H-J, Bu Z, Wang Z, Cao J. Dynamical clustering in electronic commerce systems via optimization and leadership expansion. *IEEE Trans Ind Inform* 2020;16(8): 5327–34. <https://doi.org/10.1109/TII.2019.2960835>.
- [29] Kim M-S, Han J. A particle-and-density based evolutionary clustering method for dynamic networks. *Proc VLDB Endow* 2009;2(1):622–33. <https://doi.org/10.14778/1687627.1687698>.
- [30] Cui Z, Xue Z, Fan T, Cai X, Zhang W. A many-objective evolutionary algorithm based on constraints for collaborative computation offloading. *Swarm Evol Comput* 2023;77:101244. <https://doi.org/10.1016/j.swevo.2023.101244>.
- [31] Zhang Z-K, Liu C, Zhang Y-C, Zhou T. Solving the cold-start problem in recommender systems with social tags. *EPL Europhys Lett* 2010;92(2):28002. <https://doi.org/10.1209/0295-5075/92/28002>.
- [32] Zhou T, Ren J, Medo M, Zhang Y-C. Bipartite network projection and personal recommendation. *Phys Rev E* 2007;76(4):046115. <https://doi.org/10.1103/PhysRevE.76.046115>.
- [33] Cui Z, Wen J, Lan Y, Zhang Z, Cai J. Communication-efficient federated recommendation model based on many-objective evolutionary algorithm. *Expert Syst Appl* 2022;201:116963. <https://doi.org/10.1016/j.eswa.2022.116963>.
- [34] Sun Y, Bian K, Liu Z, Sun X, Yao R, Wang R. Adaptive strategies based on differential evolutionary algorithm for many-objective optimization. *Discrete Dyn Nat Soc* 2021;2021:1–17.
- [35] Lin Q, Wang X, Hu B, Ma L, Chen F, Li J, et al. Multiobjective personalized recommendation algorithm using extreme point guided evolutionary computation. *Complexity* 2018;2018:1–18.
- [36] Gao S, Song S, Cheng J, Todo Y, Zhou M. Incorporation of solvent effect into multi-objective evolutionary algorithm for improved protein structure prediction. *IEEE/ACM Trans Comput Biol Bioinform* 2018;15(4):1365–78. <https://doi.org/10.1109/TCBB.2017.2705094>.