Received: 4 June 2021 | Revised: 22 August 2021 | Accepted: 3 September 2021

DOI: 10.1002/int.22688

RESEARCH ARTICLE

WILEY

Multicriteria recommendation based on bacterial foraging optimization

Shuang Geng[1] | **Xiaofu He**[1] | **Yixin Wang**[1] | **Hong Wang**[1] | **Ben Niu**[1] | **Kris M. Law**[2]

[1]College of Management, Shenzhen University, Shenzhen, China

[2]School of Engineering, Faculty of Science, Engineering and Built Environment, Deakin University, Geelong, Australia

Correspondence
Hong Wang, College of Management, Shenzhen University, A518, Mingli Building, Xueyuan road 1066 Shenzhen 518071, China.
Email: ms.hongwang@gmail.com

Abstract

Recommender systems assist users to make decisions among a huge volume of options. Accuracy-oriented recommender systems focus on the prediction power of algorithms and neglect that users may appreciate diverse and novel recommendations in real-world scenarios. Thus, this paper proposed a multicriteria recommendation model that can optimize the recommendation accuracy, diversity, novelty, and individual tendency simultaneously. Additionally, a new multiobjective bacterial foraging optimization method is proposed to improve its searching capability and the performance of recommendation model. The proposed optimization-based multicriteria recommendation algorithm is compared with existing methods on both benchmark functions and real-world data sets. The results demonstrate that the proposed algorithm is superior to other recommendation algorithms in most cases. This study provides insights in recommendation system design and draws scholarly attention to the optimization-based recommendation strategy.

KEYWORDS

bacterial foraging optimization, hybrid recommendation algorithm, multicriteria recommendation system, multiobjective optimization

1 | INTRODUCTION

Recommendation systems utilize user historical information and side information to build connections between users and items. Recommender system (RS) contributes to the core competitive advantages of intelligent systems, such as e-commerce, social media, and short video platform, by identifying the most relevant products and reduces consumers' costs for searching.[1] In the past few decades, the huge demand of recommendation applications accelerates the development of recommendation approaches. Conventional recommendation techniques focus on maximizing the prediction accuracy and recommending highly rated items to users.[2] However, users tend to receive similar items due to the accuracy-oriented recommendation methods.[3] This phenomenon was coined as "portfolio effect" by Ali and van Stam.[4] For this reason, researchers suggest some other user perceived criteria, such as novelty representing what portion of the recommended items the user did not know about, and diversity of recommendation list.[5] Additionally, each user has his/her own preference for recommendation accuracy, novelty, and diversity, which creates challenges for recommendation methods effectiveness.

Various recommendation techniques have been proposed which consider not only the result accuracy but also other criteria, such as novelty and diversity. Most popular approaches include the two-stage approach which focuses on re-ranking items in the recommendation list,[6–8] graph-based approach,[9] clustering-based approach,[10–12] and matrix factorization approach.[13,14] Most of these approaches use a sequential process which optimizes one criteria at a time and then optimize the second criteria, and few of them consider more than three performance criteria simultaneously. Therefore, some studies utilize heuristic optimization methods by formulating the recommendation task as an optimization problem. The recommendation performance criteria are regarded as optimization goals. For example,[15–19] use particle swarm optimization (PSO) and evolutionary algorithm (EA) to balance the technique performance in accuracy and diversity, or accuracy and item unpopularity. However, these heuristic optimization algorithms are probably confined in the local optimum thus not able find global optimal solution. Inspired by the chemotactic (foraging) behavior of *Escherichia coli*, Passino proposed the bacterial foraging optimization (BFO) method which has superior global searching capability.[20] So far, most of these optimization-based recommendation techniques only leverage the optimization methods while have not considered enhancing the optimization algorithm fundamentally.

Therefore, in this paper, we propose a multicriteria recommendation methods building upon the collaborative filtering (CF) algorithm and enhanced BFO algorithm. We formulate the recommendation task as a multiobjective optimization problem that aims at simultaneously maximizing the recommendation accuracy, diversity, novelty, and fitness with individual tendency. To improve the search capability and convergence speed of BFO simultaneously, a hypercube fast searching strategy is proposed and used in BFO. The multiobjective BFO can effectively identify the Pareto frontier in the optimization problem, which contains more than one recommendation options with superior performance in at least one of the performance dimensions.

The main contributions of this study are as follows:

- An optimization-based CF recommendation method is proposed which considers four recommendation objectives simultaneously for the first time, that is, accuracy, diversity, novelty, and individual user tendency.

- An improved multiobjective bacteria foraging optimization algorithms based on hypercube fast searching strategy is proposed which demonstrate superior effectiveness by obtaining the Pareto frontier in solution space.
- The proposed recommendation method can obtain multiple optimal trade-off solutions for a target user, and help users to select the most appropriate solution according to personalized preference.

The reminder of this paper is organized as follows: Section 2 gives the related literature including CF and bacteria foraging optimization. Section 3 describes the proposed multicriteria recommendation method and the improved multiobjective BFO. Section 5 shows extensive experiments to evaluate the effectiveness of proposed method. Section 6 discusses the results and concludes this paper.

2 | RELATED WORK

2.1 | CF

Many approaches and techniques have been developed for RS applications in the past 20 years.[21] CF is one of the most commonly used and studied recommendation technique given its simplicity and effectiveness. As depicted in Figure 1, CF-based RS employs user features, item features, and user-item rating information to produce a ranked list of items as recommendation candidates. CF-based RSs are proven effective to "point" users to new and related items.

Content boosted collaborative filtering is a variation of CF method that combines user-based CF and item-based CF. The content boosted CF is realized by two steps: neighborhood selection and rating prediction. These two steps can be explained as follows.

2.1.1 | Neighborhood selection

In the setting of RS problem, a set of users is denoted by U and a set of items is denoted as M. A profile for user i consists of l_u user features. A profile for item j consists of l_j item features. A set of explicit ratings R is available in the RS database, where R_{ij} is the rating given by ith user to jth item. It is commonly supposed that one rating can be made by any user $u \in U$ for a

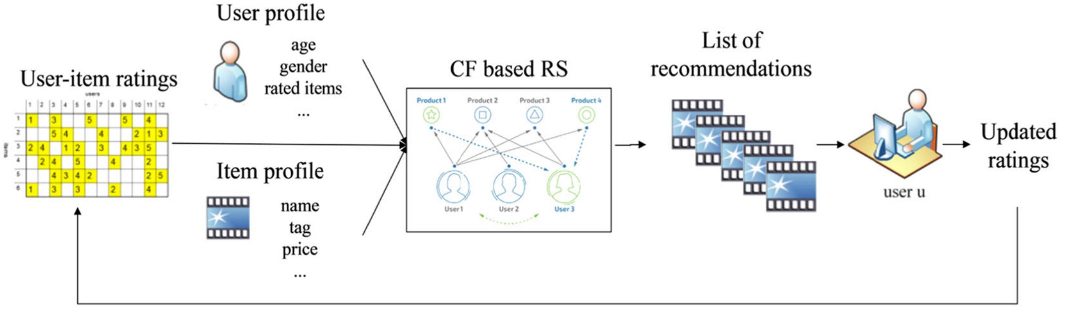

FIGURE 1 An example of CF-based RS. CF, collaborative filtering; RS, recommender system [Color figure can be viewed at wileyonlinelibrary.com]

particular item $m \in M$. The subset of users that have rated an item j is denoted by notion U_j. Likewise, M_i represents the subset of items that have been rated by a user i.

The first step to predict rating of user i on item j is to find the closest profile to the active user and active item. The distance between user profiles is calculated using Equation (1). User rating is also treated as one of the user features, the distance for rating feature is calculated according to the Pearson similarity function shown in Equation (2). The distance between item profiles is calculated using Equation (3). Among the item features, rating is also treated as a feature and the distance is calculated according to the Pearson similarity function (2) as well. The distance information between user profiles and item profiles help to identify the nearest neighbors of a target user.

$$Dist(u_a, u_b) = \sum w_i * Dist(f_{ai}, f_{bi}), \tag{1}$$

$$PearsonSim(ra, rb) = \frac{\sum_{i=1}^{m} (ra_i - \overline{ra_i}) * (rb_i - \overline{rb_i})}{\sqrt{\sum_{i=1}^{m} (ra_i - \overline{ra_i})^2 * \sum_{i=1}^{m} (rb_i - \overline{rb_i})^2}}, \tag{2}$$

$$Dist(I_a, I_b) = \sum w_j * Dist(f_{aj}, f_{bj}), \tag{3}$$

where w_i is the weight of ith feature in user profile; $Dist(f_{ai}, f_{bi})$ the measure of distance between ith feature in the user a profile and user b profile; ra_i the rating of user a to commonly rated item i; rb_i the rating of user b to commonly rated item i; $\overline{ra_i}$ the average of ratings of user a on all rated items; $\overline{rb_i}$ the average of ratings of user b on all rated items; M the total number of items rated commonly by user a and user b; w_j *the* weight of jth feature in item profile; and $Dist(f_{aj}, f_{bj})$ the measure of distance between jth feature in the item a and item b profile.

2.1.2 | Rating prediction

In the rating prediction process, k nearest neighborhoods of user i, which is denoted as $U = \{u_1, ..., u_k\}$. The ratings of k nearest neighborhoods on item j is calculated using Equation (4). The rating of user i on item j is then calculated by Equation (5). The candidate items are then ranked by predicted ratings. The items with highest ratings above the threshold are selected as recommendation results.

$$R_{k,j} = \begin{cases} R_{k,j}, & \text{if user } k \text{ has rated item } j \text{ already,} \\ \frac{\sum Dist(I_f, I_j) * R_{k,f}}{\sum Dist(I_f, I_j)}, & \text{otherwise,} \end{cases} \tag{4}$$

$$R_{i,j} = \overline{R_i} + \frac{\sum Dist(u_i, u_k) * (R_{k,j} - \overline{R_k})}{\sum Dist(u_i, u_k)}, \tag{5}$$

where $R_{k,j}$ is the rating of user k on item j; $Dist(I_f, I_j)$ the distance between item f and item j calculated according to Equation (3), here only n nearest neighborhoods of item j is used; $Dist(u_i, u_k)$ the Pearson distance between user i and user k calculated according to Equation (2); $R_{k,j}$ the rating of user k on item j; and $\overline{R_k}$ the average of ratings of user k on his/her rated items M_k.

Despite the simplicity of CF method, when the item set is very large but with only a very small portion is rated, the user-item matrix can be extremely sparse that cannot provide effective prediction results. New user or new item problems also make it difficult to identify similar users or items for predictions, which can result poor system accuracy performance.[22,23] As a result, a great number of studies leverage user and item side information to alleviate the data sparsity problem.[24] Considering that different user and item features may contribute differently in the recommendation generation process, the weighting method is adopted in this study.

2.2 | Swarm intelligence (SI) based RS

SI represents a set of nature inspired algorithms that employ a population of simple agents who interact with each other to explore the solution space for optimal solution.[25] SI-based recommendation method is an emerging trend in RS research with diverse application scenarios give the good performance of SI in solving the multiobjective optimization and feature selection problems[26,27] categorized the SI-based recommendation approaches into six categories, namely, optimizing feature weights, clustering users or items, assisting in graph-based recommendation scenarios, re-ranking recommendations, building latent factor models, and others. Among these studies, a lion's share of work focus on using SI to optimize weights of different features or parameters of RS.

Feature weighting and parameter setting are important process in content-based RS which uses multiple user and item features to identify neighborhood. Rad and Lucas[28] defined user similarity in different feature dimensions and used particle swarm optimization to determine the optimal weights of each similarity dimension (e.g., location, occupation, age). Following this model case, some researchers adopted different SI techniques, such as gray would optimization,[29] fuzzy PSO,[30] gravitational search algorithm30, and bat algorithm.[31] Sobecki incorporated SI into KNN algorithm for student course recommendation and compared the performance of particle swarm optimization, ant colony optimization, artificial bee colony optimization, invasive weed optimization and bat algorithm.[32] Results show that ant colony optimization, outperformed other approaches in prediction accuracy.

Moreover, SI techniques are used in multiobjective recommendation problems that consider more than one performance factors. Geng et al.[17] employed nondominated neighbor immune algorithm (NNIA) to select Top-N recommendation list from candidate recommendation list obtained by item-based CF. Multiobjective evolutionary algorithm and its variation have been employed to solve the two-objective recommendation problem.[16,18,19] Ribeiro et al.[33] used Pareto-efficient ranking method to find the optimal solutions in multiobjective recommendation task. Results in these studies validated the superior performance of SI techniques in comparison to traditional single-objective recommendation methods. However, few studies adequately evaluate the searching capability and converge speed of the adopted SI techniques, and none of them investigated more than three recommendation objectives.

2.3 | BFO

Inspired by the foraging behavior of *E. coli,* BFO was proposed by Passino in 2002.[20] It is composed of three steps in the optimization process: chemotaxis, reproduction, elimination, and dispersal.

- *Chemotaxis.* Chemotaxis is defined in BFO to simulate the swarming and tumbling behaviors of *E. coli*. Specifically, the flagellum of bacteria are used to control the directions of movement. The running operation is conducted when they are counter-clockwise, while the tumbling operation is performed when they are clockwise. In BFO method, the direction angle Δ is used to decide the counterclockwise or clockwise operation. The chemotaxis can be formulated as follows:

$$\theta(i+1, j, k, l) = \theta(i, j, k, l) + C(i)\frac{\Delta(i)}{\sqrt{\Delta(i)^T \Delta(i)}}, \tag{6}$$

where $\theta(i, j, k, l)$ is the position of the ith bacterium at jth reproduction, kth reproduction, and lth reproduction. $\Delta(i)$ is the direction angle of the ith bacterium randomly generated in $[-1, 1]$, and $C(i)$ is the chemotaxis step size. The larger values of $C(i)$ contribute to more randomness of the population for global search, while smaller values make it necessary to conduct more local search.

- *Reproduction*. In BFO, the health condition of bacteria over the past chemotactic process can be evaluated and used as a criterion for the reproduction. The parameter Jhealth_i is used to evaluate the searching capability of ith bacterium, which could be calculated as

$$\text{Jhealth}_i = \sum_{j=1}^{Nc} \text{J}(i, j, k, l). \tag{7}$$

The first half ranking bacteria with the better performance will be kept in population and the second half bacteria with poor search capability will be replaced by the first half bacteria as follows:

$$\theta(i+Sr, j, k, l) = \theta(i, j, k, l), \tag{8}$$

where $J(i, j, k, l)$ is the fitness value of ith bacterium at the jth chemotaxis, kth reproduction, lth dispersal, and $S_r = Pop/2$, Pop is the population size. If the optimization problem is to minimize the objective fitness function, then the smaller values of Jhealth_i means the better health condition of the bacterium.

- *Elimination and dispersal*. Following the predefined chemotactic steps and reproduction time, elimination-dispersal is taken to move the bacteria to the dynamic position.

$$\theta(i, j, k, l) = lb + (ub - lb) \times rand, \tag{9}$$

where ub and lb are the upper and lower boundary of the positions, and $rand$ is a randomly generated constant ranging from 0 to 1.

3 | THE BFO-BASED MULTICRITERIA RECOMMENDATION METHOD

In this section, we first provide an overview of the proposed recommendation method. Then the improved multiobjective BFO algorithm is described.

As depicted in Figure 2, besides historical user-item ratings, both user and item features are utilized in the recommendation framework which builds upon CF. The improved multiobjective

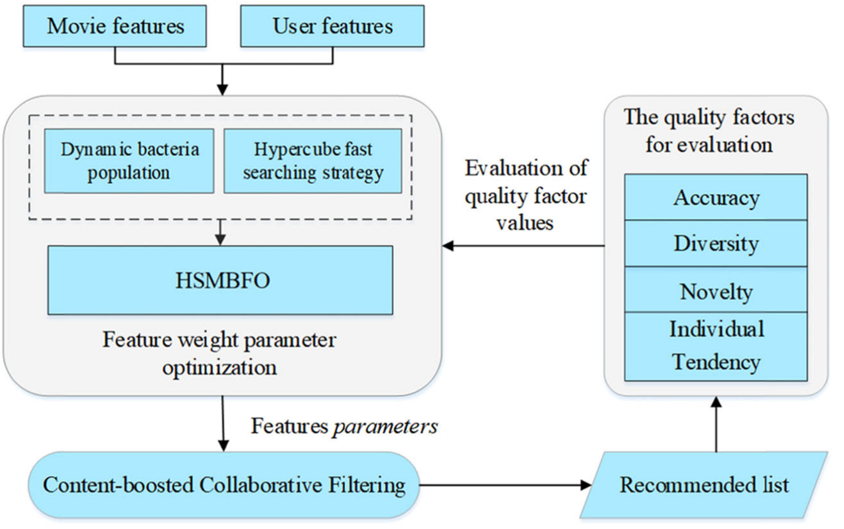

FIGURE 2 Overview of proposed multicriteria recommendation method [Color figure can be viewed at wileyonlinelibrary.com]

BFO was used to search for optimal weighting parameters of different feature similarities. The operation process of proposed recommendation method is illustrated by Pseudo code 1.

Pseudo code 1: HFSMOBFO for Multiple-Objective Quality Factors Optimization

Inputs: User-item ratings, user features, item features
Initialization: Bacteria positions θ, evaluation criteria (f_m^i, $m = 1, \ldots, M$)
For $l = 1{:}Ned$(**Elimination and dispersal loop**)
  For $k = 1{:}Nre$ (**Reproduction loop**)
    For $j = 1{:}Nc$(**Chemotaxis loop)**
      For $i = 1 : Pop$
        Location information oriented chemotaxis using Equations (16)-(19)
        Calculate the fitness values of multiple objectives (f_m^i, $m = 1, \ldots, M$)
      End for
      Create ranked set of bacteria *val* (see Pseudo code 1)
      Obtain virtual fitness value *fit* using Equations (20)-(21)
    End for (Chemotaxis loop)
    Partial reproduction using Pseudo code 2
End for (Reproduction loop)
Partial elimination using Pseudo code 3
End for (Elimination and dispersal loop)
Outputs: non-dominated frontier (user feature weightings and item feature weightings)

3.1 | Objective functions

In searching for the best combination of different feature similarities, four recommendation objectives are to be optimized, namely, accuracy, diversity, novelty, and individual user tendency. Those four criteria are considered in this study owing to their popularity in existing multicriteria recommendation system evaluation.[34,35]

The prediction accuracy measures how close the predicted rating is to the real rating. Therefore, the first objective of recommendation algorithm is to minimize the error in prediction as show in the following equation:

$$\text{Min } f_1 = \frac{1}{N * K} \sum_{i=1}^{N} \sum_{j=1}^{K} |Actual\ \ Rating_{ij} - Predict\ \ Rating_{ij}|, \tag{10}$$

where N is number of users selected from training data for the learning of optimal model parameters (criteria weightings), K is the number of recommended items for each user. $Actual\ \ Rating_{ij}$ represents the known rating of user i on item j. $Predict\ \ Rating_{ij}$ represents the predicted rating of user i on item j.

The recommendation diversity can be measured in multiple ways, such as the average pairwise dissimilarity,[36] Gini coefficient,[37] user perceived diversity based on questionnaire response,[38] and so forth. The pairwise dissimilarity measure is adopted in this study given its effectiveness in previous studies. The second objective for RS is to maximize the diversity function Equation (11).

$$\text{Max } f_2 = \frac{1}{N} \sum_{u \in users}^{N} \frac{\sum_{i \in S_u^1} \sum_{j \in S_u^1} (1 - Dist\,(I_i, I_j))}{L(L-1)}, \tag{11}$$

where S_u is the recommendation list of user u; L is the length of recommendation list; $Dist\,(I_i, I_j)$ is the distance between item i and j that are both in S_u. The distance calculation is explained in Section 3.1 as shown in Equation (1).

The novelty of recommendation list is measured by the following equation:[18]

$$\text{Max } f_3 = \frac{1}{N} \sum_{u \in \text{users}}^{N} \frac{\sum_{i \in S_u^1} \log_2 \left(\frac{N}{N_{rj}}\right)}{|S_u^1|}, \tag{12}$$

where N_{rj} denotes the number of ratings for item j in the training data. This equation measures the newness of recommendation list.

Individual user tendency measures the closeness between user history tendency (preference) and tendency of recommended list. The tendency measures the level of diversity that users prefers. Therefore, the objective of RS algorithm is calculated using Equation (15).[39] Terms in the equations have been defined in earlier sections. Besides the above quality criteria, explainability further impacts the persuasiveness of recommendations.[40] In this study, we mainly discuss four focal criteria—accuracy, diversity, novelty, and tendency.

$$\text{Tendency1} = \sum_{i \in S_u} \sum_{j \in S_u} \frac{\text{Dist}(I_i, I_j)}{|S_u|^2}, \tag{13}$$

$$\text{Tendency2} = \sum_{i \in S_u^1} \sum_{j \in S_u^1} \frac{\text{Dist}(I_i, I_j)}{|S_u^1|^2}, \tag{14}$$

$$\text{Min } f_4 = \frac{1}{N} \sum_{j=1}^{N} |\text{Tendency1} - \text{Tendency2}|. \tag{15}$$

3.2 | The improved multiobjective BFO

The improved multiobjective BFO is named as HFSMOBFO, which integrates four novel strategies to improve the searching capability and converge speed of traditional BFO. The four strategies are: (1) location information-oriented chemotaxis; (2) hypercube fast searching strategy for nondominated Pareto front; (3) virtual fitness of multiobjective optimization; (4) dynamic population generation operator. The overall framework of HFSMOBFO is illustrated in Figure 3. Explanations of each strategy are described below.

3.2.1 | Location information-oriented chemotaxis

In the basic BFO, there is no information exchange between the bacteria. Though it may bring higher capability for bacteria to search for the global best, the randomness exploitation without information exchange would consume more computational time if the problems are rather complexity with high dimensional space. Therefore, location information-oriented strategies are leveraged in the process of chemotaxis.

A new swimming direction generation method is designed with the intuition that the remained locations are best locations and will disseminate pheromone to attract other bacteria. Meanwhile, each bacterium will receive a stochastic pheromone from environment. The swimming direction index based on location information in chemotaxis process is defined as follows:

$$roll_i^d = C \cdot (norminv(rand, 1, P) - 1) \cdot (\theta^d(i, j, k, l) - \theta^d(A_i, j, k, l)), \tag{16}$$

$$C = \max\left\{CB \cdot \frac{\log_2 n}{n}, CB \cdot \frac{\log_2 \mu}{\mu}\right\}, \tag{17}$$

where $roll_i^d$ is swimming direction of the ith bacterium in dth dimension, $d \in \{1,2, ..., D\}$. The total dimension of position is D. A_i is a randomly selected bacterium, which the ith bacterium swims but the position should be different from index i. $norminv(\cdot)$ is the Gaussian distraction sampling function, whose parameters include: a random sampling probability between 0 and 1, mean value (equals to 1), variance P. The expression "$norminv(rand, 1, P) - 1$" controls the swimming step and improves the diversity in generated direction by proving possible random reduction or direction reverse. CB is the basic chemotaxis step size, C is step size after self-adaptation and it decreases as the iteration number increases. μ is a constant defined previously, which is larger than 1 and smaller than the largest iteration times. When n is larger than μ, C equals to step size calculated at iteration μ.

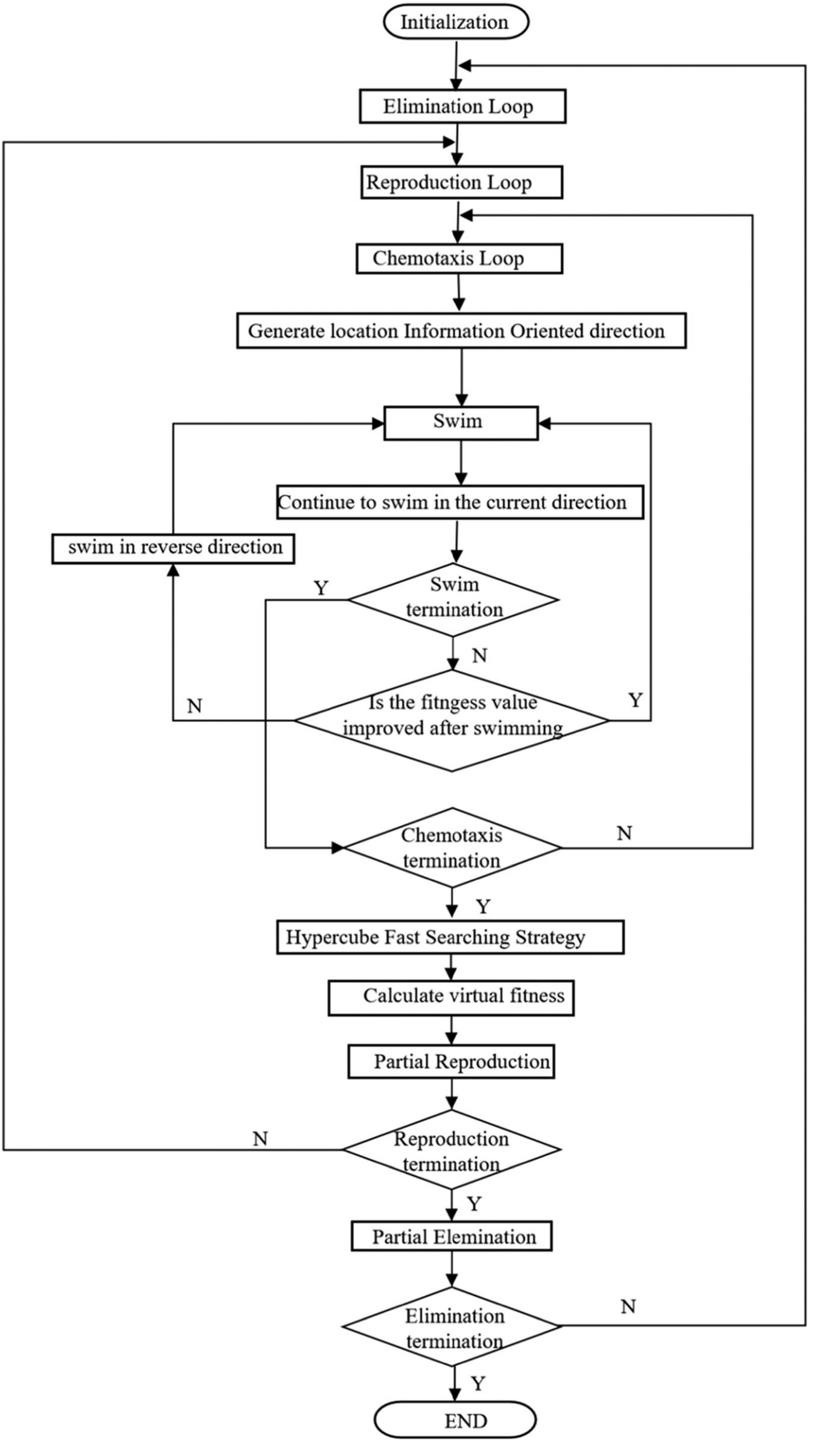

FIGURE 3 Overall framework of proposed HFSMOBFO [Color figure can be viewed at wileyonlinelibrary.com]

Based on the direction generator *roll*, the chemotaxis of bacteria is redesigned to accelerate the convergence and avoid the disturbance of searching for better positions (toward bacterium A_i), as shown in the following equations:

$$\theta^d(i,j,k,l) = \theta^d(i,j,k,l) + roll_i^d, \tag{18}$$

$$\theta^d(i,j,k,l) = \begin{cases} \theta^d(A_i,j,k,l) & \text{if } (rand_i^d \le 0.5) \\ \theta^d(i,j,k,l) & \text{otherwise} \end{cases} = 1, 2, ..., D, \tag{19}$$

where $rand_i^d$ is a randomly generated constant ranging from 0 to 1.

3.2.2 | Hypercube fast searching strategy (HFS)

HFS is proposed to identify the dominated, nondominated solutions according to the fitness values of bacteria. In this process, it is assumed that there are M number of objectives (i.e., $\mathrm{F(X)} = [f_1, f_2, ..., f_M]$) to be minimized, simultaneously. The definition of nondominated solutions or Pareto frontier can be provided.

Definition 1 (Dominance definition). Let X_1 and X_2 are two feasible solutions of the multiobjective problem. The solution X_1 dominates X_2 when two conditions are satisfied:

(1) Solution X_1 is no worse than X_2 in all objectives (i.e., $\forall\ i,\ f_i(X_1) \le f_i(X_2)$);
(2) Solution X_1 is strictly better than X_2 in at least one objective (i.e., $\exists\ i,\ f_i(X_1) < f_i(X_2)$);

Definition 2 (Nondominance solution set). Given a set of solutions, the nondominance solution set is a set of solutions which are not dominated by any other feasible solutions. The Pareto-optimal set consists of nondominated solution set of entire feasible decision space, and can be mapped as a boundary which is normally regarded as the Pareto front.

In HFS, the bacteria are divided into three groups in obtaining the nondominated solutions, that is, "selection," "removal," or "keeping" group. The HFS constructing hypercube recursively by drawing perpendicular lines toward the axis of the objective dimension at the positions where there exists maximum or minimum value. These lines form a hypercube that separate the solution space into two sub-space that helps to classify the bacteria.

To illustrate the HFS, we use a two-objective problem with two dimensions in solution space for example. As depicted in Figure 4, by drawing perpendicular lines from the leftmost point and lowermost point toward the f_1 axis and f_2 axis, a hypercube (a rectangle in 2-D space) is formed. The solution points outside the rectangle represent dominated solutions and the points inside represent nondominated solutions. In this way, all the solution points can be divided into three types in each iteration, namely, "select" (the selected points for hypercube creation), "keep" (the points within the hypercube), "remove" (the points outside the hypercube).

According to the pseudo code of HFS operator (Pseudo code 1), we use FIT^{iter} to represent the matrix containing the fitness values of all the positions in current iteration, and PF^{iter} as the current Pareto front. At the very beginning, the fitness values of all bacteria are ranked

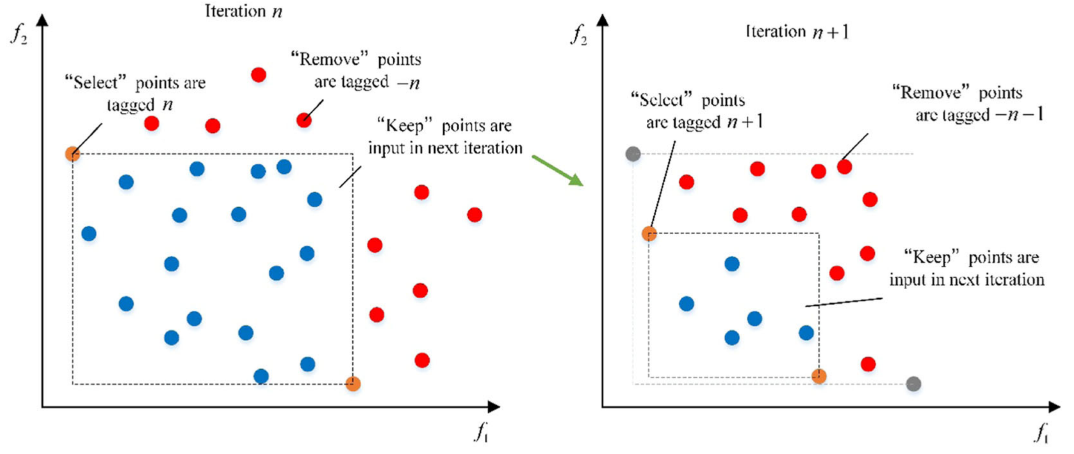

FIGURE 4 2-D illustration of HFS nondominated ranking operator. HFS, Hypercube fast searching strategy [Color figure can be viewed at wileyonlinelibrary.com]

according to FIT^{iter} and PF^{iter}. A parameter val is created to record the scores which related to the ranks of bacteria. It assumed that val_i^{iter} represents the score of the i^{th}ith ($i = 1, ..., Pop$) bacterium at the $iter^{th}$ HFS iteration, which was assigned with zero at the very beginning. In each iteration, the val_i^{iter} of bacteria classified as the "selection" type are increasing with the iteration (i.e., $val_i^{iter} = val_i^{iter-1} + 1$), while the val_i^{iter} of bacteria classified as the "removal" type are decreasing (i.e., $val_i^{iter} = val_i^{iter-1} - 1$). Thus, the earlier a point was classified as "selection" type, the larger val it will obtain. By contrast, the earlier a point was classified as "remove" type, the smaller val it will obtain.

The HFS for nondominated Pareto front is provided in Pseudo code 2.

Pseudo code 2: HFS for Non-dominated Pareto Front

Initialization:$val_i = 0$, $i = 1, ..., Pop$
While the "keep" group is not empty
  For$t = 1: M$ (//M is the number of objectives)
    Find minimum fitness value of the objectives FIT_t^{iter} among the bacteria in "keep" group, and put the bacterium with the minimum fitness value into the "selection" group (//only one bacterium with minimum fitness in t^{th} objective is obtained for "selection" group)
  End for(//thus, the number of bacteria obtained is M)
  Construct a hypercube based on those M bacteria
  Remove the bacteria outside the hypercube
  Update the "keep" group using the bacteria within the hypercube but not the 'selection'
  Update the score value of bacteria in "selection" group, i.e. $val_i = val_i + 1$, where i is the bacterium in "selection" group
  Update the score value of bacteria in "removal" group, i.e. $val_i = val_i - 1$, where i is the bacterium in "removal" group
End while
Outputs: bacteria ranking scores (recorded in val) and non-dominated solutions (consisting of bacteria from the "selection" group)

3.2.3 | Virtual fitness

In the above HFS strategy, the performance of bacteria can be ranked using the parameter *val*. However, simply using *val* for evaluation cannot comprehensively represent the good or poor contributions of solutions to the objectives. In multiobjective problems, all the nondominated solutions are important. For this reason, we defined "virtual fitness" as the criterion to evaluate the performance of bacteria in each iteration. To differentiate the performance of bacteria and enhance the calculation process efficiency, the virtual fitness is defined as follows:

$$val_i = \begin{cases} val_i - min_{i=1,\,\dots,\,Pop}\{val_i\} & \text{if } val_i < 0 \\ val_i - max_{i=1,\,\dots,\,Pop}\{val_i\} - 1 & \text{if } val_i > 0 \end{cases} \quad i = 1, ..,Pop, \tag{20}$$

$$fit_i = val_i \cdot \sum_{m=1}^{M} f_m^i / (min_{m=1,\,\dots,\,M}\{f_m^i\}) \quad m = 1, 2, \dots, M, \tag{21}$$

where *Pop* is total number of bacteria in the given set, fit_i represents the virtual fitness value of the ith bacterium. MM is the total number of objectives and m is the objective index, f_m^i is the fitness value of the ith bacterium at the mth object.

It needs to note that, for solutions with positive values in *val* (i.e., $val_i > 0, i = 1, ..,Pop$), the larger value of *val* indicates that the solution is more closer to the Pareto front. And for solutions with negative values of *val* (i.e., $val < 0, i = 1, ..,Pop$), the smaller value of *val* indicates that solution is far from the Pareto front.

Pseudo code 3: Partial reproduction

Sort the bacteria according to the virtual fitness value *fit* (obtained using Pseudo code 1 and Equations (20)–(21)), the bacteria located at the frontier indicate the better performance
for $i = 1 : Sr$//where $Sr = Pop/2Pop$ is the population size
  for $d = 1 : D$
    if $rand < P_{re}$
      $\theta^d(Sr + i, j, k, l) = \theta^d(i, j, k, l)$//the d^{th} dimension of the position
  else
      $\theta^d(Sr + i, j, k, l) = \theta^d(Sr + i, j, k, l)$
    end *if*
  end *for*
end *for*

3.2.4 | Partial reproduction and partial elimination

The duplication process in the original BFO simply replaces half of the bacteria (less healthy bacteria) with the healthier bacteria. This dynamic population generation strategy makes it harder to distinguish between good bacteria and bad bacteria on the Pareto frontier. Therefore, we use the "virtual fitness" to identify less healthy bacteria and replace part of their dimension values with dimension values of healthier bacteria according to a probability-based threshold P_{re}. The Pseudo code 3 describes the process of partial reproduction in details.

The primary goal of elimination is to increase the diversity of bacteria positions. However, multiobjective optimization problems usually have relatively flat gradient and initializing the bacteria positions may disturb the approaching to optimal positions. Therefore, only some dimensions of bacteria are initialized with the probability of P_{ed} to maintain the approaching to optimal solutions as well as increase diversity. The process of partial elimination is realized as follow (Pseudo code 4):

Pseudo code 4: Partial elimination

if $rand < P_{ed}$

  $\theta^d(Sr+i,j,k,l) = lb^d + (ub^d - lb^d) \times rand$//the d^{th} dimension of the position, $d = 1, \ldots, D$

else

  $\theta^d(Sr+i,j,k,l) = \theta^d(Sr+i,j,k,l)$

end *if*

where ub^d and lb^d are the upper and lower boundary of the positions in the d^{th} dimension, and *rand* is a randomly generated constant ranging from 0 to 1.

4 | EXPERIMENT AND RESULTS

This section firstly evaluates the performance of proposed HFSMOBFO by making comparison with six well-known multiobjective optimization algorithms: (NSGAII,[41] MOPSO,[42] MRBCO,[43] MOCLBFO,[44] MCMBFO[45]) on 14 well-known benchmark functions with two or three objectives. Subsequently, Section 4.2 further demonstrates the effectiveness of HFSMOBFO-based CF by comparing it with eight CF-based recommendation algorithms on a real-world data set.

4.1 | Comparison on benchmark problems

In this paper, experiments were performed using the MATLAB R2019a (Math Works Inc.) environment. Table 1 listed the parameter settings for the five algorithms. The parameters settings for the six candidate algorithms are consistent with their corresponding references. All the algorithms use equal population size, number of objectives, dimension, and FEs.

The six candidate algorithms are firstly tested on ten representative bi-objective benchmark problems, including Schaffer1 and Schaffer2,[46] Fonseca and Fleming,[47] Kursawe,[48] Poloni,[49] ZDT1, ZDT2, ZDT3, ZDT4, and ZDT6[50] and multiobjective benchmark problems DTLZ test suite.[51] The popular performance measure, hyper-volume (HV)[52] and inverted generational distance (IGD),[53,54] are used as indicator of algorithm performance.

The HV metric measures the size of objective space covered by an approximation set and is usually used to evaluate the convergence and diversity of the obtained approximate optimal solution set. The larger the HV value, the better the comprehensive performance of the algorithm. The IGD is an inverse variation of generational distance metric. IGD calculates the minimum Euclidean distance between an approximation set A and the Pareto front PF^*. IGD can effectively evaluate the convergence performance and distribution performance of the algorithm. The smaller the value, the better the comprehensive performance of the algorithm.

TABLE 1 Parameter settings considered in this study

Algorithms	Parameters	Meaning	Value
NSGAII[41]	proC	The probability of doing crossover	1
	disC	The distribution index of simulated binary crossover	20
	proM	The expectation of number of bits doing mutation	1
	disM	The distribution index of polynomial mutation	20
MOPSO[42]	div	The number of divisions in each objective	10
	w	The inertial weight in PSO	0.4
MORBCO[43]	freRe	The frequency of reproduction	10
	freEl	The frequency of elimination and dispersal	20
	NS	The maximum length of swimming	4
	Pel	Probability of elimination	0.25
	C1, C2	The learning factor	3
	C	Chemotaxis step	0.1
	pRep	Proportion of bacteria for reproducing	1/3
	npbest	the maximum number of historical best positions	2
MOCLBFO[44]	Nc	Number of chemotaxis steps	200
	Ns	Limits the length of a swim when it is on a gradient	4
	Nre	The number of reproduction steps	5
	Sr	The number of bacteria reproductions (splits) per generation	$0.5 \times S$
	Ned	The number of elimination-dispersal events	2
	ped	Probability that each bacteria will be eliminated/ dispersed	0.2
MCMBFO[45]	Nc	Number of chemotaxis steps	200
	Ns	Limit the length of a swim when it is on a gradient	4
	Nre	The number of reproduction steps	5
	Ned	The number of elimination-dispersal events	2
	C_{max}	The maximum of the run length unit	1
	C_{min}	The minimum of the run length unit	0.1
	C1	The left learning factor of the ring topology structure	7.5
	C2	The right learning factor of the ring topology structure	7.5
	C3	The learning factor of the star topology structure	10
	$Infor_{max}$	The maximum of the neighborhood information	0.002
	T	The number of slave-swarms	4

TABLE 1 (Continued)

Algorithms	Parameters	Meaning	Value
HFSMOBFO	S	The number of bacteria;	100
	Nrep	The number of external archive	100
	Nc	Number of chemotaxis steps	200
	Ns	Limits the length of a swim when it is on a gradient	4
	Nre	The number of reproduction steps	5
	Ned	The number of elimination-dispersal events	2
	Pre	Probability of reproduction	0.3
	Ped	Probability of elimination	0.25
	C	The run-length unit during each run or tumble	0.4

Therefore, the HV and IGD measure can adequately measure diversity and convergence of A if sufficient members of PF* are known.[55] Each trial is conducted 30 times independently. The experiment results are shown in Tables 2 and 3 with the average results highlighted in bold and the standard deviation results are shown in brackets. The obtained Pareto solutions of six algorithms are illustrated in Figure 5.

Results in Table 2 reveal that the proposed HFSMOBFO approach outperforms other candidate algorithms on seven (Schaffer1, Schaffer1, Kursawe, ZDT2, ZDT4, DTLZ2, DTLZ4) test problems. NSGAII performs better on ZDT1, ZDT3, DTLZ5, and DTLZ6 test problem. It is noteworthy that HFSMOBFO obtains almost equally better result as NSGAII on DTLZ6. According to results in Table 3, the IGD values of HFSMOBFO is better on seven (Schaffer1, Schaffer2, Kursawe, ZDT2, ZDT4, DTLZ2, DTLZ4) test problems. The NSGAII still demonstrate good solution quality on ZDT1, ZDT3, DTLZ5 and DTLZ6. Additionally, Figure 5 also demonstrates that the proposed HFSMOBFO outperforms most of candidate algorithms and equally well as NSGAII, which proves that it could effectively find the optimal solutions with good diversity and convergence in most cases.

4.2 | Comparison on movie recommendation task

Given the outstanding optimization capability of HFSMOBFO, the HFSMOBFO-based CF is further evaluated in real recommendation data sets using Grouplens—movielens 100 K and 1 M data sets. These two data sets are widely used test sets for evaluating the performance of recommendation algorithms in RS research. The data set details are reported in Table 4. In the experiment, the default five subsets of 100 K data set are used for fivefold training and testing, the 1 M data set is divided into training set (80% of data) and test set (20% of data) randomly and used for fivefold training and testing.

Four conventional recommendation methods and four optimization-based recommendation methods are implemented to quantify the performance of HFSMOBFO-based CF (HSMBFO-CF). The list of algorithms is shown in Table 5. Table 6 lists the parameter setting in the CF calculation process.

TABLE 2 HV results of algorithms

Algorithms	MOPSO	NSGAII	MCLBFO	MORBCO	MCMBFO	HFSMOBFO
Schaffer1	8.5700e−1 (9.86e−5)	8.5871e−1 (1.72e−4)	8.4820e−1 (2.08e−5)	8.4654e−1 (6.56e−4)	8.5261e−1 (6.24e−5)	**8.6351e−1** (7.65e−4)
Schaffer2	6.7694e−1 (5.69e−3)	6.7075e−1 (1.05e−4)	6.6896e−1 (6.71e−4)	6.5081e−01 (1.03e−4)	6.6080e−01 (1.45e−4)	**6.7838e−1** (1.99e−4)
Fonseca	4.2521e−1 (4.94e−3)	5.2850e−1 (1.87e−4)	4.4221e−1 (7.94e−4)	**6.3553e−1** (8.40e−4)	4.1433e−1 (1.27e−3)	4.1226e−1 (5.14e−3)
Kursawe	5.0184e−1 (5.78e−3)	5.0401e−1 (3.23e−4)	4.1312e−1 (1.72e−3)	6.1774e−01 (3.80e−03)	5.9826e−1 (8.08e−3)	**7.2526e−1** (2.19e−3)
Poloni	8.7664e−1 (1.27e−5)	8.7816e−1 (3.66e−5)	8.0836e−1 (2.05e−5)	8.1783e−1 (4.01e−5)	**9.0116e−1** (2.57e−5)	8.8794e−1 (9.81e−4)
ZDT1	6.2164e−2 (7.39e−2)	**7.1927e−1** (1.78e−4)	9.7422e−2 (8.61e−3)	6.1888e−1 (3.48e−3)	7.0946e−1 (1.24e−3)	5.9480e−1 (6.73e−2)
ZDT2	0.0000e+0 (0.00e+0)	4.4409e−1 (5.57e−5)	1.7222e−1 (4.57e−3)	4.5471e−1 (3.31e−4)	2.5406e−1 (3.25e−3)	**4.5961e−1** (2.06e−4)
ZDT3	9.1992e−2 (8.72e−2)	**5.9938e−1** (1.19e−4)	1.0009e−1 (7.43e−3)	5.6436e−1 (3.07e−4)	5.7826e−1 (3.07e−4)	4.9141e−1 (5.57e−4)
ZDT4	5.6554e−1 (1.37e−4)	4.1829e−1 (3.54e−4)	1.9732e−2 (1.33e−3)	7.6753e−1 (1.67e−4)	5.6428e−1 (6.77e−3)	**7.6996e−1** (5.47e−4)
ZDT6	3.6888e−1 (3.22e−2)	1.8830e−1 (5.98e−2)	4.5539e−1 (1.68e−2)	5.5351e−1 (8.84e−03)	**5.7951e−1** (1.73e−1)	5.6051e−1 (4.05e−2)
DTLZ2	4.6665e−1 (1.08e−2)	5.4637e−1 (2.46e−3)	3.4485e−1 (1.52e−2)	4.0098e−01 (5.03e−2)	6.6047e−01 (3.47e−3)	**7.0704e−1** (8.41e−3)
DTLZ4	4.1845e−1 (5.74e−2)	5.4884e−1 (1.97e−3)	2.0830e−01 (4.60e−2)	3.3580e−01 (1.81e−1)	6.2129e−01 (2.71e−2)	**7.0432e−1** (9.60e−3)
DTLZ5	1.9477e−1 (2.28e−3)	**2.0133e−1** (1.06e−3)	9.4738e−2 (1.59e−2)	9.0026e−2 (5.20e−2)	1.0601e−1 (7.65e−2)	1.1974e−1 (5.02e−3)
DTLZ6	3.9556e−2 (8.85e−2)	**2.0053e−1** (7.40e−3)	2.0167e−2 (2.19e−2)	1.5591e−1 (1.59e−3)	9.2341e−2 (2.53e−3)	2.0008e−1 (2.38e−3)

TABLE 3 IGD results of algorithms

Algorithms	MOPSO	NSGAII	MCLBFO	MORBCO	MCMBFO	HFSMOBFO
Schaffer1	3.0587e−2 (3.25e−3)	3.1476e−2 (8.16e−4)	3.5310e−2 (2.38e−3)	3.8986e−2 (6.56e−3)	3.5941e−2 (6.39e−3)	**3.0342e−2** (7.65e−4)
Schaffer2	5.1760e−2 (5.61e−3)	5.3286e−2 (1.03e−3)	6.7106e−2 (3.38e−3)	5.3010e−02 (8.05e−03)	6.7091e−2 (3.40e−3)	**5.0241e−2** (3.27e−3)
Fonseca	9.5233e−3 (5.05e−3)	5.4131e−3 (1.56e−4)	7.0518e−3 (6.24e−4)	**5.3551e−3** (3.08e−4)	6.8732e−3 (1.25e−3)	8.8688e−3 (9.33e−4)
Kursawe	1.1198e−1 (1.13e−1)	4.2876e−2 (3.42e−3)	8.9431e−1 (2.78e−1)	1.1455e−1 (8.07e−3)	6.9832e−2 (2.07e−1)	**4.3246e−2** (3.30e−3)
Poloni	6.5248e−2 (4.39e−2)	7.0252e−2 (5.11e−3)	5.6552e−2 (3.98e−3)	8.8044e−2 (2.44e−2)	**4.0162e−2** (5.55e−3)	5.0632e−2 (5.34e−3)
ZDT1	8.7193e−1 (3.27e−1)	**4.7885e−3** (1.14e−4)	4.8625e−1 (6.88e−3)	1.1737e−2 (2.47e−3)	6.3461e−3 (8.25e−3)	1.0343e−2 (7.76e−4)
ZDT2	2.1324e+0 (2.49e−1)	4.8233e−3 (7.93e−3)	5.7622e−1 (3.70e−2)	5.4393e−3 (2.87e−3)	6.5211e−2 (2.71e−2)	**3.9972e−3** (1.90e−3)
ZDT3	7.7166e−1 (1.82e−1)	**5.3560e−3** (2.17e−4)	5.1836e−1 (1.38e−2)	9.0061e−2 (1.03e−2)	8.5667e−3 (5.76e−2)	9.0554e−3 (5.35e−4)
ZDT4	6.7960e−2 (9.93e−3)	9.8747e−2 (1.65e−3)	1.8548e+1 (1.52e−2)	4.3631e−2 (9.80e−2)	6.5642e−2 (2.04e−2)	**3.5131e−2** (3.31e−1)
ZDT6	7.8445e−3 (2.67e−2)	3.6928e−2 (7.07e−3)	7.6537e−3 (8.57e−3)	4.1556e−3 (4.13e−3)	**2.1004e−3** (5.44e−3)	2.4611e−3 (7.05e−2)
DTLZ2	9.7457e−2 (5.97e−3)	9.1343e−2 (1.68e−3)	8.9942e−1 (1.08e−2)	2.9430e−1 (2.67e−2)	6.7021e−1 (7.47e−1)	**8.9925e−2** (3.80e−3)
DTLZ4	1.4183e−1 (8.49e−2)	9.8557e−2 (1.39e−3)	1.3522e+0 (3.60e−2)	5.3638e−1 (1.34e−1)	1.3653e+0 (1.05e+0)	**8.7259e−2** (3.90e−3)
DTLZ5	8.9738e−2 (8.19e−3)	**7.8168e−2** (1.96e−3)	8.7265e−1 (2.19e−2)	1.5817e−1 (5.63e−2)	6.1122e−1 (7.34e−1)	2.6852e−1 (3.23e−3)
DTLZ6	1.7508e+0 (1.24e+0)	**8.9984e−2** (2.02e−3)	1.0022e+0 (3.08e−2)	8.0314e−1 (6.82e−1)	8.0024e−1 (8.31e−1)	3.1179e−1 (1.17e−1)

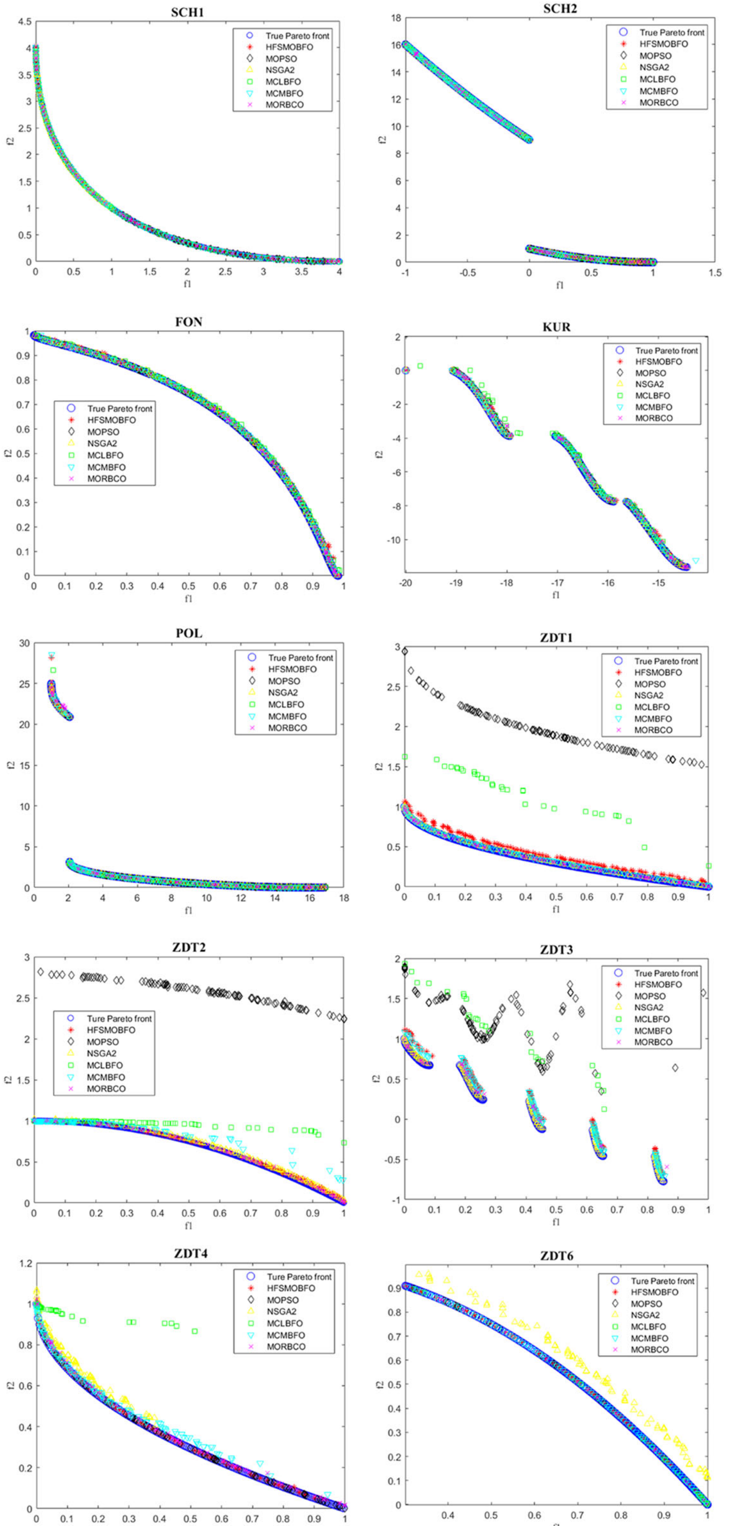

FIGURE 5 Pareto solution sets obtained by each algorithm [Color figure can be viewed at wileyonlinelibrary.com]

TABLE 4 Description of movielens data set

Data sets	Number of users	Number of movies	Number of ratings	Sparsity
Movielens 100 K	943	1682	10,000	93.7%
Movielens 1 M	6040	3883	1,002,209	95.5%

TABLE 5 Algorithms in movie recommendation experiment

Algorithms	Meaning
CBCF	Content boosted collaborative filtering[56]
Item-CF	Item-based collaborative filtering[57]
User-CF	User-based collaborative filtering[58]
KNN-CF	K nearest neighbors collaborative filtering[59]
NSGAII-CF	Collaborative filtering integrated with NSGAII
MOPSO-CF	Collaborative filtering integrated with MOPSO
MOBFO-CF	Collaborative filtering integrated with MOBFO
MOCLBFO-CF	Collaborative filtering integrated with MOCLBFO
HSMBFO-CF	Collaborative filtering integrated with HFSMOBFO

TABLE 6 Parameter setting for CF

Parameter	Meaning	Value
threshold	Threshold for movie scores to determine if recommend or not	3.5
N	Number of users selected from training data set for model learning	50
k	Use k nearest user neighbors for CF	30
r	Select the top r movies with highest predicted scores as candidates	10
scLb	Lower bound of the feature weighting	0
scUb	Upper bound of the feature weighting	1
dim	Dimensions for the weightings	4
λNSM	Weighting for movie name similarity	0.01
λTSM	Weighting for movie time similarity	0.1
λGCSM	Weighting for movie genre similarity	0.1
λRISM	Weighting for movie ratings similarity	0.8

The accuracy performance of recommended list is measured by three commonly adopted measures: precision, recall, and F1 measure.[60] Given that precision usually drops with an increase in the length of recommendation list while recall improves, F1 measure combines precision and recall and provide an overall accuracy measure. Moreover, the proposed method is also evaluated by how well they balance between the multiple RS quality factors.

TABLE 7 Results on 100 K movielens data set

Algorithms		Precision	Recall	F1-measure	Diversity	Novelty	Tendency
CBCF		57.937	46.001	50.969	82.049	208.846	81.536
Item-CF		71.463	53.266	60.832	81.939	219.644	81.811
KNN-CF		65.528	48.601	55.527	81.911	227.310	81.063
User-CF		69.266	50.941	58.365	81.918	225.260	81.036
NSGAII-CF	Ave	71.605	54.251	61.371	81.896	227.169	82.495
	Pre	71.436	53.529	60.835	81.904	227.278	82.253
	Div	71.281	54.275	61.275	81.903	227.280	82.717
	Ten	71.484	54.089	61.248	81.899	227.069	82.695
MOPSO-CF	Ave	71.374	54.050	61.175	81.894	227.426	82.318
	Pre	71.372	53.716	60.946	81.856	227.785	82.222
	Div	71.436	54.072	61.215	81.848	226.294	82.527
	Ten	71.237	54.300	61.269	81.852	226.856	82.501
MOBFO-CF	Ave	69.400	**60.359**	**64.199**	82.064	226.745	82.710
	Pre	70.338	58.768	63.637	82.029	226.799	82.564
	Div	69.768	59.916	64.098	82.059	227.391	83.110
	Ten	69.800	59.926	64.117	82.061	227.310	82.878
MCLBFO-CF	Ave	71.353	58.309	63.615	82.019	231.117	83.119
	Pre	71.415	58.166	63.580	82.007	231.189	82.968
	Div	71.326	57.900	63.383	82.006	231.048	83.004
	Ten	71.326	57.900	63.383	82.006	231.048	83.004
HSMBFO-CF	Ave	**71.619**	54.445	61.502	82.062	**233.952**	**84.686**
	Pre	71.492	54.093	61.226	**82.078**	230.520	83.566
	Div	70.987	55.159	61.613	**82.786**	**233.293**	84.552
	Ten	71.222	54.847	61.576	82.074	233.108	**84.574**

The diversity, novelty, and user tendency evaluation criteria are measured by Equations (11), (12), (15), respectively.

For optimization-based CF (NSGAII-CF, MOPSO-CF, MOBFO-CF, MCLBFO-CF, HSMBFO-CF), the obtained Pareto frontier represents a set of optimal solutions. Since the optimization-based method can obtain a collection of nondominated solutions on the Pareto frontier, there exist optimal solution for each of the four optimization objective or evaluation criteria. Therefore, it is necessary to investigate the algorithm performance in each dimension. In this paper, we developed four different measures to guarantee the reliability and generalizability of results. Specifically, the "Ave" measure is calculated by averaging the performance of all nondominated solutions on Pareto frontier. The "Pre" measure is to select the solution with best precision measure on the Pareto frontier. The "Div" measure is to select the point with best diversity measure on the Pareto frontier, and the "Ten" measure is to select the point

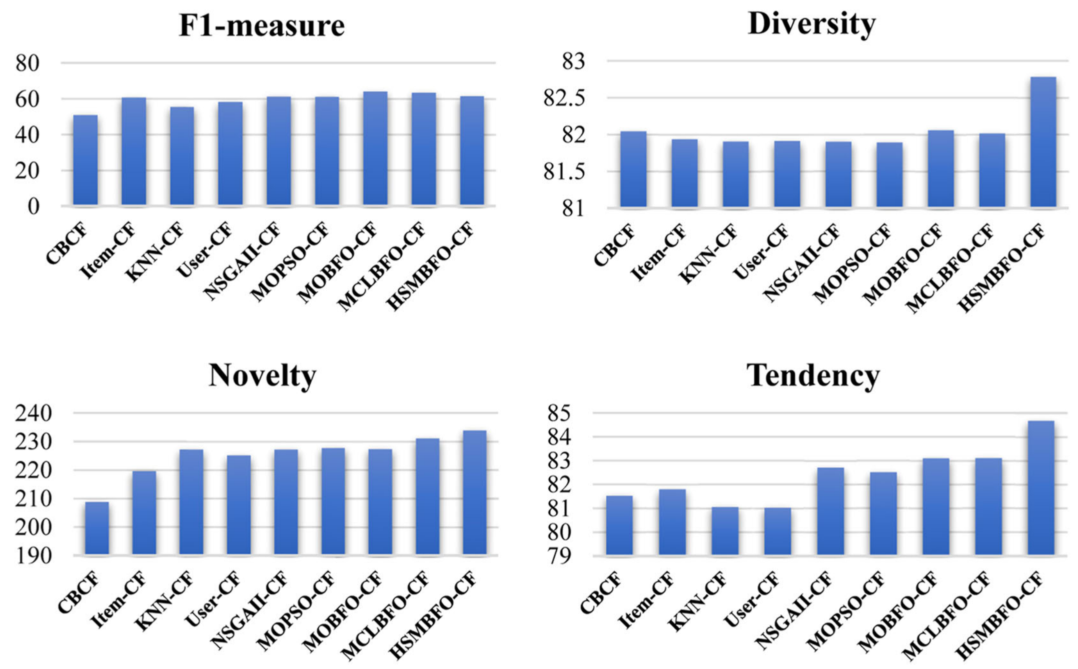

FIGURE 6 Results for F1-measure, diversity, novelty, and tendency on 100 K movielens data set [Color figure can be viewed at wileyonlinelibrary.com]

with best tendency measure on the Pareto frontier. The reported value for "Ave," "Pre," "Div," and "Ten" are averaged over all tested users.

Results on the 100 K movielens data set are reported in Table 7 with the best two values highlighted in bold. Figure 6 visualizes the performance of algorithms for F1-measure, diversity, novelty, and tendency. Since optimization-based CF methods obtain four sets of results, the best values are used for visualization. Table 8 and Figure 7 presents the experiment results on the 1 M movielens data set.

From Table 7, the proposed HSMBFO-CF outperforms all other algorithms in providing more precise, diverse and novel items according to user tendency, while MOBFO-CF shows the superiority in terms of recall and F1-measure. From the comparison results on 1 M movienlends data set in Table 8, HSMBFO-CF achieves the best performance in F1-measure, diversity, novelty and tendency, while Item-CF and MOBFO-CF obtain the superiority in terms of precision and recall, respectively.

4.3 | Discussions

- *The optimization performance of HFSMOBFO*. The main drawback associated with the multiobjective BFO and its variants is their less conductive convergence speed in searching for the Pareto front. The improvement of chemotaxis process with information exchange and hypercube fast searching strategy prove to be effective in enhancing the searching capability of BFO. The partial reproduction and partial elimination strategy also effectively increase the diversity of bacteria positions in BFO by creating an external archive that preserves selected particles for the next generation and copies the nondominant solution in the current

TABLE 8 Results on 1 M movielens data set

Algorithms		Precision	Recall	F1-measure	Diversity	Novelty	Tendency
CBCF		50.595	66.990	57.649	86.861	563.540	75.282
Item-CF		**70.695**	77.587	73.980	86.838	529.260	69.472
KNN-CF		50.648	66.191	57.385	86.865	563.074	74.805
User-CF		50.209	66.196	57.104	86.865	563.706	75.385
NSGAII-CF	Ave	64.776	87.592	74.475	86.853	568.460	75.253
	Pre	64.313	87.655	74.190	86.815	569.651	75.311
	Div	64.837	87.794	74.589	86.854	568.266	75.225
	Ten	**64.922**	87.435	74.515	86.875	568.173	75.030
MOPSO-CF	Ave	64.226	87.618	74.117	86.822	568.282	**75.406**
	Pre	64.252	87.374	74.048	86.815	569.766	75.375
	Div	64.526	87.627	74.319	86.856	568.205	75.214
	Ten	64.473	87.948	74.401	86.841	568.118	75.311
MOBFO-CF	Ave	62.396	87.444	72.824	86.877	567.529	74.845
	Pre	62.477	**88.534**	73.255	86.868	568.430	75.164
	Div	62.405	**88.177**	73.085	86.878	567.904	75.193
	Ten	62.397	88.069	73.042	86.878	567.940	75.156
MCLBFO-CF	Ave	60.932	83.648	70.502	86.874	564.902	73.774
	Pre	60.283	82.714	69.736	86.887	564.776	73.739
	Div	60.291	82.728	69.747	86.887	564.779	73.749
	Ten	60.291	82.728	69.747	86.887	564.779	73.749
HSMBFO-CF	Ave	64.668	87.866	74.503	87.107	**570.176**	75.209
	Pre	64.259	88.028	74.287	**87.117**	**570.318**	**75.413**
	Div	64.801	87.867	**74.591**	**87.403**	569.122	75.208
	Ten	64.787	87.899	**74.593**	87.116	568.946	75.182

population into the archive. Thus, the obtained solutions of HFSMOBFO outperforms the other three recently proposed BFO-based methods (MRBCO, MOCLBFO, MCMBFO). Though MOPSO and NSGAII are two well-known multiobjective methods which demonstrate to be superior to most multiobjective methods, the comparison results on benchmark problems demonstrate that the proposed HFSMOBFO still maintain the superiority in most cases.

- *The recommendation performance of HSMBFO-CF*. From the comparison results on 100k and 1 M movienlens data sets, we can find that the proposed HSMBFO-CF obtain the best performance in most cases for six performance measures. MOBFO-CF obtains the best performance in terms of recall and F1-measure on 100k data set and best recall on 1 M data set. Item-CF gets the best in terms of precision on 1 M data set. The proposed HSMBFO-CF has obvious advantages in providing more diverse and novel items according to user

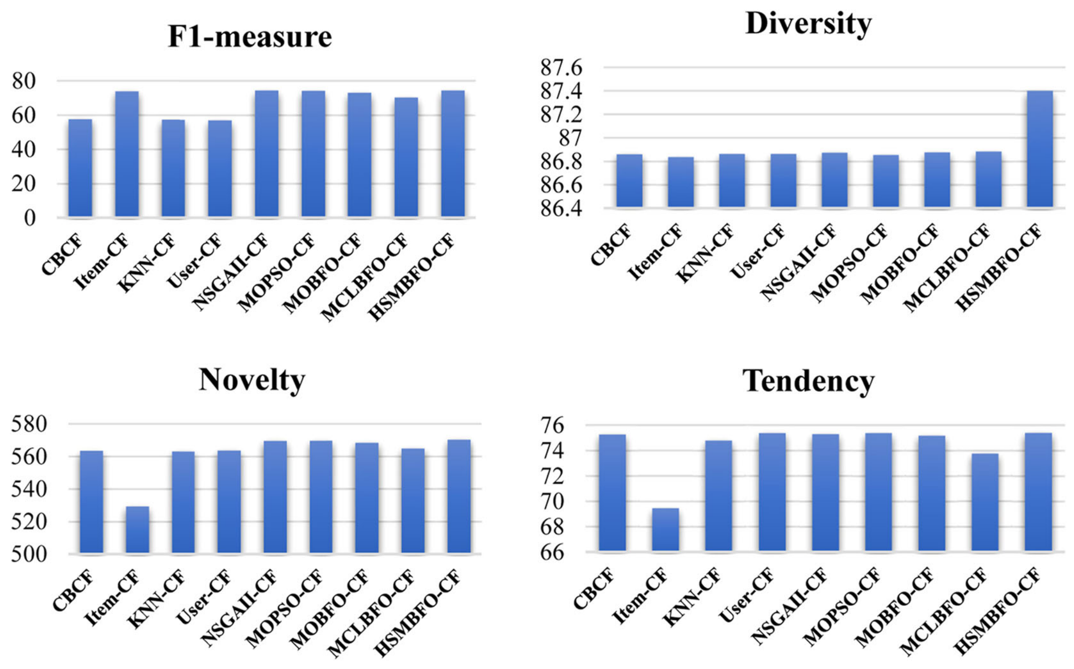

FIGURE 7 Results for F1-measure, diversity, novelty, and tendency on 1 M movielens data set [Color figure can be viewed at wileyonlinelibrary.com]

individual tendency. It implies that the proposed strategy can effectively optimize other recommendation quality factors with only slight decline in accuracy. Overall, the optimization-based algorithms show superior performance in recommendation applications. Among the intelligent optimization-based recommendation methods, HSMBFO-CF and MOBFO-CF demonstrate the best optimization capability, and HSMBFO-CF yields the most diverse and novel solutions according to user individual tendency. Evidently, HSMBFO-CF demonstrates stable capability to obtain the optimal solutions effectively on large movielens data set. Thus, it implies that incorporating heuristic optimization methods in intelligent system design contributes to greater number of feasible solutions with adequate diversity and novelty.

5 | CONCLUSIONS AND FUTURE WORK

This paper proposes a new multiobjective BFO methods by incorporating four strategies in the operation process. First, the adoption of information-oriented chemotaxis enables the information exchange between bacteria, thus accelerates the convergence speed and avoids the disturbance of searching for better positions. Then, the hypercube fast searching strategy for nondominated Pareto front classifies solutions into dominated and nondominated groups, and provides a simple method to save computational cost. The use of "virtual fitness" effectively represents the contributions of solutions to each optimization objective. Lastly, the partial reproduction and partial elimination strategy maintains the solution diversity when removing less healthy bacteria.

Current results on real world data sets demonstrate that the proposed recommendation method achieves good balance across different evaluation measures. From application perspective, this study provides an importance reference for intelligent RS design.

There are a few limitations in this study that point to our future research. First, we will focus on incorporating user contextual information and constructing effective user feature representation. Second, we also plan to investigate other recommendation system performance criteria, that is, explainability, based on current study. Third, using larger real-world data sets can enhance the generalization of optimization-based recommendation methods.

ACKNOWLEDGMENTS

This study is supported by the National Natural Science Foundation of China (Grants Nos. 71901150, 71901152, 71971143), Major Research plan for National Natural Science Foundation of China (91846301), Major Project for National Natural Science Foundation of China (71790615), Natural Science Foundation of Guangdong Province (Grants No. 2020A1515010752, 2020A1515010749), Shenzhen Higher Education Institute Support Plan (20200826144104001).

CONFLICT OF INTERESTS

The authors declare that there are no conflict of interests.

AUTHOR CONTRIBUTIONS

Conceptualization: Shuang Geng. *Methodology*: Shuang Geng and Hong Wang. *Validation*: Xiaofu He, Shuang Geng, and Yixin Wang. *Writing—original draft: Shuang Geng. Writing—review and editing*: Ben Niu and Kris M. Law. *Project administration*: Shuang Geng, Ben Niu, and Hong Wang. *Funding acquisition*: Shuang Geng, Ben Niu, and Hong Wang. All authors have read and agreed to the published version of the manuscript.

DATA AVAILABILITY STATEMENT

All relevant data are within the paper.

ORCID

Shuang Geng http://orcid.org/0000-0001-8146-0786
Xiaofu He http://orcid.org/0000-0003-2862-0925
Yixin Wang https://orcid.org/0000-0002-2789-7909
Hong Wang http://orcid.org/0000-0002-4671-6122
Ben Niu http://orcid.org/0000-0001-5822-8743
Kris M. Law http://orcid.org/0000-0003-3659-0033

REFERENCES

1. Köhler S, Wöhner T, Peters R. The impact of consumer preferences on the accuracy of collaborative filtering recommender systems. *Electron Mark*. 2016;26(4):369-379.
2. Liu H, Hu Z, Mian A, Tian H, Zhu X. A new user similarity model to improve the accuracy of collaborative filtering. *Knowl-Based Syst*. 2014;56(Jan):156-166.
3. McNee SM, Riedl J, Konstan JA. Being accurate is not enough: how accuracy metrics have hurt recommender systems. CHI'06 Extended Abstracts On Human Factors in Computing Systems. April 2006:1097-1101.
4. Ali K, Van Stam W. TiVo: making show recommendations using a distributed collaborative filtering architecture. Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2004:394-401.

5. Shani G, Gunawardana A. Evaluating recommendation systems. In: Francesco R, Lior R, Bracha S, Paul BK. eds. *Recommender Systems Handbook*. Springer; 2011:257-297.
6. Adomavicius G, Kwon Y. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Trans Knowl Data Eng*. 2011;24(5):896-911.
7. Di Noia T, Rosati J, Tomeo P, Di Sciascio E. Adaptive multi-attribute diversity for recommender systems. *Inf Sci*. 2017;382:234-253.
8. Lathia N, Hailes S, Capra L, Amatriain X. Temporal diversity in recommender systems. Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval. July 2010:210-217.
9. Lee K, Lee K. Escaping your comfort zone: a graph-based recommender system for finding novel recommendations among relevant items. *Expert Systems with Applications*. 2015;42(10):4851-4858.
10. Aytekin T, Karakaya MÖ. Clustering-based diversity improvement in top-N recommendation. *J Intell Inf Syst*. 2014;42(1):1-18.
11. Boim R, Milo T, Novgorodov S. Diversification and refinement in collaborative filtering recommender. Proceedings of the 20th ACM International Conference on Information and Knowledge Management. 2011:739-744.
12. Park YJ, Tuzhilin A. The long tail of recommender systems and how to leverage it. Proceedings of the 2008 ACM conference on Recommender systems. 2008:11-18.
13. Gogna A, Majumdar A. DiABlO: Optimization based design for improving diversity in recommender system. *Inf Sci*. 2017;378:59-74.
14. Gogna A, Majumdar A. Balancing accuracy and diversity in recommendations using matrix completion framework. *Knowl-Based Syst*. 2017;125:83-95.
15. Choudhary P, Kant V, Dwivedi P. Handling natural noise in multi criteria recommender system utilizing effective similarity measure and particle swarm optimization. *Proc Comput Sci*. 2017;115:853-862.
16. Cui L, Ou P, Fu X, Wen Z, Lu N. A novel multi-objective evolutionary algorithm for recommendation systems. *J Parallel Distrib Comput*. 2017;103:53-63.
17. Geng B, Li L, Jiao L, Gong M, Cai Q, Wu Y. NNIA-RS: A multi-objective optimization based recommender system. *Phys A Stat Mech Appl*. 2015;424:383-397.
18. Jain A, Singh PK, Dhar J. Multi-objective item evaluation for diverse as well as novel item recommendations. *Expert Syst Appl*. 2020;139:112857.
19. Wang S, Gong M, Li H, Yang J. Multi-objective optimization for long tail recommendation. *Knowl-Based Syst*. 2016;104:145-155.
20. Passino KM. Biomimicry of bacterial foraging for distributed optimization and control. *IEEE Control Syst Mag*. 2002;22(3):52-67.
21. Lu J, Wu D, Mao M, Wang W, Zhang G. Recommender system application developments: a survey. *Decis Support Syst*. 2015;74:12-32.
22. Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Trans Knowl Data Eng*. 2005;17(6):734-749.
23. Yu K, Schwaighofer A, Tresp V, Xu X, Kriegel HP. Probabilistic memory-based collaborative filtering. *IEEE Trans Knowl Data Eng*. 2004;16(1):56-69.
24. Sun Z, Guo Q, Yang J, et al. Research commentary on recommendations with side information: a survey and research directions. *Electron Commer Res Appl*. 2019;37:100879-102499.
25. Garnier S, Gautrais J, Theraulaz G. The biological principles of swarm intelligence. *Swarm Intell*. 2007;1(1):3-31.
26. Chen YP, Li Y, Wang G, et al. A novel bacterial foraging optimization algorithm for feature selection. *Expert Syst Appl*. 2017;83:1-17.
27. Peška L, Tashu TM, Horváth T. Swarm intelligence techniques in recommender systems—a review of recent research. *Swarm Evol Comput*. 2019;48:201-219.
28. Rad HS, Lucas C. On the effectiveness of prioritized user-profile and detecting active users in collaborative filtering recommender systems. 2007 IEEE 23rd International Conference on Data Engineering Workshop. IEEE; 2007:863-870.
29. Katarya R. Movie recommender system with metaheuristic artificial bee. *Neural Comput Appl*. 2018;30(6):1983-1990.

30. Wasid M, Kant V. A particle swarm approach to collaborative filtering based recommender systems through fuzzy features. *Proc Comput Sci*. 2015;54:440-448.
31. Choudhary V, Mullick D, Nagpal S. Gravitational search algorithm in recommendation systems. International Conference on Swarm Intelligence. Springer, Cham; 2017:597-607.
32. Sobecki J. Comparison of selected swarm intelligence algorithms in student courses recommendation application. *Int J Softw Eng Knowl Eng*. 2014;24(01):91-109.
33. Ribeiro MT, Lacerda A, Veloso A, Ziviani N. Pareto-efficient hybridization for multi-objective recommender systems. Proceedings of the Sixth ACM Conference on Recommender Systems. 2012: 19-26.
34. Kunaver M, Požrl T. Diversity in recommender systems—a survey. *Knowl-Based Syst*. 2017;123: 154-162.
35. Kaminskas M, Bridge D. Diversity, serendipity, novelty, and coverage: a survey and empirical analysis of beyond-accuracy objectives in recommender systems. ACM Transactions on Interactive Intelligent Systems (TiiS). 2016;7(1):1-42.
36. Bradley K, Smyth B. Improving recommendation diversity. Proceedings of the Twelfth Irish Conference on Artificial Intelligence and Cognitive Science. Maynooth. 2001;85:141-152.
37. Fleder DM, Hosanagar K. Recommender systems and their impact on sales diversity. Proceedings of the 8th ACM Conference on Electronic Commerce. 2007:192-199.
38. Vargas S. *Novelty and Diversity Enhancement and Evaluation in Recommender Systems*. Master's thesis. Autonomous University of Madrid; 2012.
39. Jugovac M, Jannach D, Lerche L. Efficient optimization of multiple recommendation quality factors according to individual user tendencies. *Expert Syst Appl*. 2017;81:321-331.
40. Sardianos C, Varlamis I, Chronis C, et al. The emergence of explainability of intelligent systems: delivering explainable and personalized recommendations for energy efficiency. *Int J Intell Syst*. 2021; 36(2):656-680.
41. Deb K, Pratap A, Agarwal S, Meyarivan TAMT. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans Evol Comput*. 2002;6(2):182-197.
42. Coello CAC, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization. *IEEE Trans Evol Comput*. 2004;8(3):256-279.
43. Niu B, Liu Q, Wang Z, Tan L, Li L. Multi-objective bacterial colony optimization algorithm for integrated container terminal scheduling problem. *Nat Comput*. 2021;20(1):89-104.
44. Niu B, Wang H, Wang J, Tan L. Multi-objective bacterial foraging optimization. *Neurocomputing*. 2013;116: 336-345.
45. Niu B, Liu J, Tan L. Multi-swarm cooperative multi-objective bacterial foraging optimisation. *Int J Bio-Inspir Comput*. 2019;13(1):21-31.
46. Schaffer JD. Multiple objective optimization with vector evaluated genetic algorithms. Proceedings of International Conference on Genetic Algorithms (ICGA'85). 1985:93–100.
47. Fonseca CM, Fleming PJ. Multiobjective optimization and multiple constraint handling with evolutionary algorithms. I. A unified formulation. *IEEE Trans Syst Man Cybern Part A Syst Humans*. 1998; 28(1):26-37.
48. Kursawe F. A variant of evolution strategies for vector optimization. International Conference on Parallel Problem Solving from Nature. Springer, Berlin, Heidelberg. 1990:193-197.
49. Poloni C, Giurgevich A, Onesti L, Pediroda V. Hybridization of a multi-objective genetic algorithm, a neural network and a classical optimizer for a complex design problem in fluid dynamics. *Comput Methods Appl Mech Eng*. 2000;186(2-4):403-420.
50. Zitzler E, Thiele L. Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach. *IEEE Trans Evol Comput*. 1999;3(4):257-271.
51. Deb K, Thiele L, Laumanns M, Zitzler E. Scalable multi-objective optimization test problems. Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02. Vol. 1. IEEE. 2002:825-830.
52. Wang Z, Zhang Q, Li H, Ishibuchi H, Jiao L. On the use of two reference points in decomposition based multiobjective evolutionary algorithms. *Swarm and Evol Comput*. 2017;34:89-102.
53. Bosman PA, Thierens D. The balance between proximity and diversity in multiobjective evolutionary algorithms. *IEEE Trans Evol Comput*. 2003;7(2):174-188.

54. Li M, Zheng J. Spread assessment for evolutionary multi-objective optimization. International conference on evolutionary multi-criterion optimization. Springer, Berlin, Heidelberg. 2009:216-230.
55. Riquelme N, Von Lücken C, Baran B. Performance metrics in multi-objective optimization. In 2015 Latin American Computing Conference(CLEI). IEEE. 2015:1-11.
56. Melville P, Mooney RJ, Nagarajan R. Content-boosted collaborative filtering for improved recommendations. *AAAI/IAAI*. 2002;23:187-192.
57. Linden G, Smith B, York J. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Comput*. 2003;7(1):76-80.
58. Zhao ZD, Shang MS. User-based collaborative-filtering recommendation algorithms on hadoop. In 2010 Third International Conference on Knowledge Discovery and Data Mining. IEEE. 2010: 478-481.
59. Park Y, Park S, Lee SG, Jung W. Fast collaborative filtering with a k-nearest neighbor graph. 2014 International Conference on Big Data and Smart Computing (BIGCOMP). IEEE. 2014: 92-95.
60. Gunawardana A, Shani G. Evaluating recommender systems. *Recommender Systems Handbook*. Springer; 2015:265-308.

How to cite this article: Geng S, He X, Wang H, Ben N, Law KM. Multicriteria recommendation based on bacterial foraging optimization. *Int J Intell Syst*. 2022;37: 1618-1645. https://doi.org/10.1002/int.22688