Complex & Intelligent Systems (2021) 7:2765–2781
https://doi.org/10.1007/s40747-021-00450-6

ORIGINAL ARTICLE

A two-stage information retrieval system based on interactive multimodal genetic algorithm for query weight optimization

Hao Cong[1] · **Wei-Neng Chen**[1,2] · **Wei-Jie Yu**[3]

Received: 5 July 2020 / Accepted: 8 June 2021 / Published online: 14 July 2021

Abstract
Query weight optimization, which aims to find an optimal combination of the weights of query terms for sorting relevant documents, is an important topic in the information retrieval system. Due to the huge search space, the query optimization problem is intractable, and evolutionary algorithms have become one popular approach. But as the size of the database grows, traditional retrieval approaches may return a lot of results, which leads to low efficiency and poor practicality. To solve this problem, this paper proposes a two-stage information retrieval system based on an interactive multimodal genetic algorithm (IMGA) for a query weight optimization system. The proposed IMGA has two stages: quantity control and quality optimization. In the quantity control stage, a multimodal genetic algorithm with the aid of the niching method selects multiple promising combinations of query terms simultaneously by which the numbers of retrieved documents are controlled in an appropriate range. In the quality optimization stage, an interactive genetic algorithm is designed to find the optimal query weights so that the most user-friendly document retrieval sequence can be yielded. Users' feedback information will accelerate the optimization process, and a genetic algorithm (GA) performs interactively with the action of relevance feedback mechanism. Replacing user evaluation, a mathematical model is built to evaluate the fitness values of individuals. In the proposed two-stage method, not only the number of returned results can be controlled, but also the quality and accuracy of retrieval can be improved. The proposed method is run on the database which with more than 2000 documents. The experimental results show that our proposed method outperforms several state-of-the-art query weight optimization approaches in terms of the precision rate and the recall rate.

Keywords Query weight · Interactive · Multimodal · Genetic algorithm

Introduction

The process of an information retrieval system (IRS) is to find the information which stays consistent mostly with the user's need from a huge database [1–5]. With the development of information technology, the amount of information grows explosively. As a result, how to find the most satisfactory information from databases becomes very challenging [6–10]. Information retrieval has attracted increasing attention in recent years [11–14].

Vector space model (VSM) [15] as a famous IRS model is characterized by describing the queries. By sorting similarities between documents vectors and the query vector, an ordered sequence of retrieved documents can be obtained. The documents that meet the user's needs best will be placed in the front of the document sequence. In the VSM, the retrieval process can be seen as the process of finding the best combination of weights of query terms, which is called query optimization. With numerous possible combinations of query weights, the problem of query optimization becomes a challenging issue in information retrieval.

Evolutionary computation (EC) has become a popular technique for solving query weight optimization problems. EC is a population-based stochastic optimization method with high robustness and wide applicability [16–18]. For example, Cordon et al. [19] have verified that the application

✉ Wei-Neng Chen
cschenwn@scut.edu.cn

[1] School of Computer Science and Engineering, South China University of Technology, 510006 Guangzhou, China

[2] Pazhou Lab, 510330 Guangzhou, China

[3] School of Information Management, Sun Yat-sen University, 510006 Guangzhou, China

of evolutionary algorithms (EAs) is promising in solving the query weight optimization problem. They also stated that genetic algorithm (GA) is the most commonly-used EA in this field. Horng and Yeh et al. [20] proposed a new objective function of GA to express the score of the rank of retrieved documents. If a combination of the query weights can place relevant documents in a higher rank in the retrieved list, the solution will be assigned a better fitness value. Lopez et al. [21, 22] also investigated different fitness functions of GA and further compared them with the classical Ide algorithm [1]. The final conclusion that GA has great potential in solving the query weight optimization is drawn.

As a kind of emerging EC technology, interactive evolutionary computation (IEC) which applies user's evaluation to the process of EC, has been commonly applied in website design, data analysis, IRS, product design, and so on [23–27] in recent years. Different from traditional EC, the fitness evaluation of IEC is subjective and relative, and IEC needs users to assign fitness values for individuals. Users tend to feel tired when performing a large number of repetitive operations, which can affect the quality of user evaluation. Furthermore, the optimization results are also influenced. More and more studies focus on improving IEC with different strategies. The first approach is to improve the way users evaluate. For example, Takagi et al. [28] proposed an input method using discrete fitness values for an interactive genetic algorithm (IGA). They changed the traditional percentage system to a smaller scale of five or seven during the user evaluation, which helps reduce user fatigue. Lee et al. [29] proposed a sparse fitness evaluation in IGA for reducing the user burden. They adopt clustering to divide the population into several sub-populations. One representative individual is selected from each sub-population. Then, the fitness value of other individuals in this subpopulation is determined according to the fitness value of the representative individual and the distance to the representative one. Watanabe et al. [30] proposed an interactive genetic algorithm based on a paired comparison (PC-IGA). Instead of assigning a specific score to the user, PC-IGA allowed that the user only needed to compare two individuals and select the better one. The selected individual can enter the next round of competition until the final winner is obtained. Sun et al. [31] proposed an interactive genetic algorithm with the individual's fuzzy and stochastic fitness to replace the user's evaluation for each individual. By using a fuzzy number to assign fitness, this approach can reduce the user's burden dramatically. Another strategy for improving IEC is to build a surrogate model to evaluate the fitness value of the individual. The second method is to build the surrogate model to replace user evaluation. Wang et al. [32] proposed an interactive genetic algorithm combined with a support vector machine. Making full use of the positive and negative examples selected by the user at the initial stage, a support vector machine is utilized to construct a classifier. Experimental results show that this method can better reduce user fatigue. Li et al. [33] proposed an adaptive learning evaluation model to judge beauty instead of users in the evolutionary art system. The model extracts specific aesthetic features from internal evolutionary images and external real-world painting. By training these features, a more accurate learning method is selected and a model is established. Although these methods can alleviate the user's fatigue greatly, effective evolutionary operators are not involved. The third strategy is to modify evolutionary operators in IEC. Tinkle et al. [34] utilized the weighted hypervolume to assign the fitness of solutions for solving multiobjective optimization problems, which can accelerate the convergence of IEC. Gong et al. [35] proposed a hierarchical interactive evolution algorithm. The algorithm initially performed a global search in the entire search space. When it reached a certain level, it switched to the area search of key gene segments. This continuously reduced the search area until a satisfactory solution is found.

However, research on IEC in the field of the query weight optimization has lots of issues that need to be solved. For instance, with the increase of documents in the database, the search space increases exponentially, and the query optimization problem becomes intractable. Therefore, traditional IEC faces the disadvantage of poor efficiency in solving the query weight optimization problem. Moreover, it is necessary to find as many solutions as possible to meet the needs of the user in a single search. Traditional query weight optimization methods still need to improve the accuracy of the search to make the retrieval system more user-friendly. To solve the above issues, this paper proposes a two-stage information retrieval system based on interactive multimodal GA (IMGA) for query weight optimization. Due to the sizes of databases grow rapidly, the query optimization problem becomes a large-scale optimization problem. To increase the search accuracy of the algorithm, we adopt a multimodal GA to reduce the search space firstly, which optimizes the number of returned relevant documents in case that the number of returned documents is too many or too few. An IRS can be more user-friendly by controlling the number of returned documents. Then an interactive GA is implemented for query weight optimization to improve the accuracy of results. After the search space of the query weight optimization in the second stage is reduced, the search accuracy of the proposed method IMGA is easier to be improved. Thus, a two-stage method which contains a quantity control stage and a quality optimization stage at the meantime. The proposed method has the following contributions:

1. A two-stage method for query weight optimization is introduced, including a quantity control stage and a quality optimization stage. In this way, both the quantity and

quality of the returned results can be taken into account in the proposed approach. In the quantity control stage, a multimodal GA is used to optimize the number of returned relevant documents in case that the number of returned documents is too many or too few. In the quality control stage, an interactive GA is implemented for query weight optimization to improve the accuracy of results.

2. To tackle the sub-problems in these two stages, we adopt two different GAs respectively. In the first stage, the aim is to search for several combinations of the query terms, so that the number of retrieved documents found by each combination can be controlled in a suitable range. To obtain multiple combinations simultaneously instead of a single combination, a special multimodal GA with niching is adopted to yield multiple solutions at the same time that minimize the difference between the actual number of documents retrieved and the expected range of the number of retrieved documents. In this way, it is able to control the number of retrieved documents and decrease the scale of the optimization problem. In the second stage, an interactive GA is applied to search for the optimal weights of the query terms, so that the sequence of documents retrieved is the most suitable for the users. Replacing user evaluation, returned results that are multiple sequences of documents are evaluated by a mathematical model. What's more, for each subgroups of the query terms found during the first stage, the corresponding weights are optimized separately by an interactive GA process. In this way, the search performance of the proposed method can be improved.

To verify the effectiveness of the proposed two-stage method IMGA, we conduct the experiment on the database which contains more than 2000 documents from different categories. The proposed approach is run on the database compared with some state-of-the-art query weight optimization algorithms in terms of the retrieval performance.

The remainder of this paper is organized as follows. Section "Related concepts" gives a brief introduction to the background information. In Section "A two-stage information retrieval system with relevance feedback based on IMGA for query weight optimization", we present a novel two-stage information retrieval system based on an interactive multimodal genetic algorithm for query weight optimization. The experimental results of the proposed method are shown in Section "Experiment results". Finally, Section "Conclusion" concludes this paper.

Related concepts

Research of EAs for information retrieval mainly involves three parts, i.e., the expression of queries and documents, the relevance feedback mechanism, and the design of EAs. To facilitate understanding, this section will present the background of this work, including VSM, IEC, and GA as well. Besides, the niching strategy, designed for improving search diversity of EAs, will also be presented.

Vector space model

Vector space model (VSM) is firstly introduced by Salton et al. [15] to transform the intricate document retrieval process into intuitive vector operations in the vector space. Query terms and documents are described as vectors. By operating on the vectors, the relevance between documents and the query can be accessed by calculating the relevant degrees among vectors.

Given a query, each query term in the query is expressed as t_i. A document k which is consisted of a set of terms t_i $(1 = i = n)$ is represented as a vector $\vec{d}_k$

$$\vec{d}_k = \left(w_{1k}, w_{2k}, \ldots, w_{nk}\right), \tag{1}$$

where w_{ik} denotes the weight of the query term t_i in the document vector. The query q is also represented as a vector $\vec{Q}_k$

$$\vec{Q}_k = \left(w_{1k}, w_{2k}, \ldots, w_{nk}\right), \tag{2}$$

where w_{ik} denotes the weight of the query term t_i in the query vector.

For the ith query term t_i, its weight in the query vector is related to the frequency of the query term t_i in the document d_k. The frequency of the ith term of the query vector in the kth document is denoted by tf_{ik}. In fact, some terms appear frequently in the documents and contribute little to increase the query accuracy, so it is necessary to give lower weights to the terms that appear frequently in the document database. Assume that the number of all documents is N, the number of keywords t_i in the document is n_i, the inverse document frequency (IDF) of the term t_i is defined as follows [36]

$$\mathrm{IDF} = \log \frac{N}{n_i}. \tag{3}$$

In this way, the more frequently a term appears in the document database, the smaller value of IDF it has. When the number of keywords t_i in the document is 0, the value of IDF will be set infinity. For the sake of distinguishing the importance of each term, the weight w_{ik} for the term t_i in the document d_k is given as follows [36]

$$w_{ik} = \frac{tf_{ik}}{\max tf_{jk}} \times \mathrm{IDF}_i, \tag{4}$$

where $maxtf_{ik}$ denotes the frequency of the term that appears the most frequently in the document. In other words, the larger the weight of the term has, the more informative the term is.

Interactive evolutionary computation

In IEC, the user's need is attached to the evaluation of solutions of EC. IEC organically combines the intelligent evaluation of human beings with EC, breaking through the limitation of establishing the numerical performance index of the optimized system. The process of IEC is represented in Fig. 1.

Relevance feedback mechanism

The mechanism of relevance feedback employed in the IRS is a kind of IEC [37–39], in which part of the returned results would be marked as relevant documents or irrelevant ones by the user. This technology separates relevant documents from irrelevant documents according to the user's opinion. As a kind of relevance feedback algorithms, Ricchio's relevance feedback algorithm is widely used in Salton's SMART system in 1970 [40]. This algorithm constantly modifies the new query vector by the documents which are part of the database, marked as relevant or irrelevant with respect to the uses' need. It expects to find an optimal vector which is nearest to the centroid vector of relevant documents and farthest to the centroid vector of irrelevant documents. Relevance feedback techniques based on GA have shown great potential. Zhu et al. [41] combined different relevance feedback techniques with GA in the form of designing fitness functions and introduced three different genetic operators to develop a new GA-based query optimization method. They compared the proposed method with three well-known query optimization methods with relevance feedback: the traditional Ide Dec-hi method [1], the Horng and Yeh's method [20], and the Lopez-Pujalte et al.'s method [19] which are all based on GA. The results have shown that the information retrieval methods with the relevance feedback mechanism based on GA have better performance. In the process of the IRS, the relevant feedback mechanism mainly aims to improve the quality of final search results through user interaction [1]. The fundamental of the relevance feedback mechanism is described as follows. The user submits a query and the system returns an initial search result. Then, partial results are marked as relevant or not by the user, and the query will be modified to keep the query vector moving closer towards the centroid of vectors of relevant documents and farther from the centroid of vectors of irrelevant documents. After several relevance feedbacks, the optimal query vector is obtained.

Fitness function

The setting of the fitness function in IEC is related to users' preferences. To improve IEC, many approaches adopt a surrogate model to evaluate individuals instead of manual marking. In the relevance feedback mechanisms, fitness functions are designed in a variety of ways. One of the most famous methods is proposed by Chang et al. [36]. By comparing the degrees of similarity between the query vector and the document vectors, an ordered sequence of the retrieved documents can be returned. The degree of similarity between the query vector $\overrightarrow{Q}$ and the document vector $\overrightarrow{d_k}$ is calculated as follows

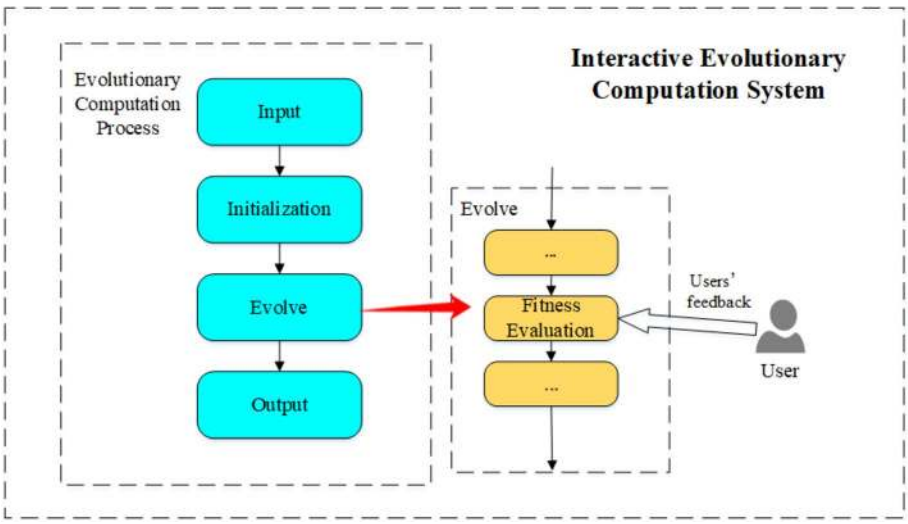

Fig. 1 The IEC system

$$S\left(\vec{Q},\vec{d}_k\right)=\frac{\sum_{i=1}^{s}\left(1-\left|w_{iq}-w_{ik}\right|\right)r_i}{\sum_{i=1}^{s}r_i}, \tag{5}$$

where $S(\vec{Q},\vec{d}_k)$ is between 0 and 1, and r_i is the relevant degree of the query term t_i. They used 16 fuzzy rules and the membership functions to infer the value of r_i for each query term t_i [20]. The larger the value of $S(\vec{Q},\vec{d}_k)$ is, the more relevant $\overrightarrow{\mathrm{Q}}$ and $\overrightarrow{\mathrm{d_k}}$ are.

The fitness value of the query can be reflected by the sequence of the retrieved documents, which is sorted by $S(\vec{Q},\vec{d}_k)$. Instead of users' evaluation of each individual, each individual is evaluated by the following formula [36]

$$F=\frac{1}{|D|}\sum_{i=1}^{|D|}r(di)\sum_{j=i}^{|D|}\frac{1}{j}, \tag{6}$$

where D is the set of documents in the system based on the user's feedback;|D| is the number of retrieved documents; r (d_i) represents the relevance of document di. If d_i is relevant to the query vector q_i, r (d_i) is equal to 1; otherwise, it is 0. The fitness value is between 0 and 1. The higher a relative document is ranked, the larger the fitness value of this sort is. Obviously, the fitness value of the top-ranked documents is equal to 1.

Genetic algorithm

Genetic algorithm (GA) is a stochastic search method inspired by the evolutionary laws of biology, which is one of the most popular evolutionary algorithms. It was first introduced in 1975 by J. Holland [42]. GA introduces the concept of population composed by chromosomes which are also regarded as individuals. Each chromosome corresponds to a solution to the problem. The population consisted of chromosomes in GA evolves iteratively to approximate the global optimal solution of the problem by performing three major evolutionary operators: selection, crossover, and mutation. By evaluating the fitness value of chromosomes, the optimal individual will be picked out.

Niching method

Niching method is a popular strategy to aid classical EAs to improve search diversity [43–45]. The principle of niching is to divide the whole population into several sub-population named niches based on various split rules. Each niche can search for one local optimum. In this way, multiple promising solutions are returned simultaneously and the diversity of the solutions can be improved. Crowding [43], as a common niching method, decomposes the whole population by distances of solutions. The framework of crowding is shown as Algorithm 1.

Algorithm 1: Niching strategy – Crowding [43]	
Input:	population P, population size N, niches number M
Step 1:	**Initialize** $P(I)$;
Step 2:	Choose an individual randomly in the population as a reference point X, and compute the its distance to other points;
Step 3:	**for** i = 1 to $\left\lceil \frac{N}{M} - 1 \right\rceil$ do
Step 4:	Select the nearest point to X in P;
Step 5:	Form a niche by X and N/M-1 points which are nearest to X;
Step 6:	Remove X and $\left\lceil \frac{N}{M} - 1 \right\rceil$ points from P;
Output:	A set of niches

A two-stage information retrieval system with relevance feedback based on IMGA for query weight optimization

An overview of the two-stage information retrieval system

The user submits a query containing multiple terms firstly. Then the optimization process is implemented with two different GAs corresponding to the two stages: the quantity control stage and the quality optimization stage. Accordingly, the quantity control stage aims to control the number of retrieved documents to improve the performance of the retrieval in the next stage and the quality optimization stage is for the purpose of optimizing the weights of query terms so that the retrieved documents are suitable for users. After obtaining the original search results, the user marks part of the search results as relevant or irrelevant. The system will continue to optimize the query weights until it meets the terminal condition. The procedure of the system is presented in Fig. 2.

At the first stage, the search space of GA is composed of all possible combinations of the original query terms. The objective of the multimodal GA in this stage is to find several combinations of the query terms so that the number of retrieved documents found by each of these combinations is in a suitable range. In order to find multiple feasible combinations simultaneously, a multimodal GA with a niching strategy is applied. The niching strategy makes it possible to obtain multiple promising groups of solutions at the same time. At the second stage, each sub-group of query terms serves as an independent query to search the database. Weights of the queries are optimized by an interactive GA with the relevance feedback mechanism and the

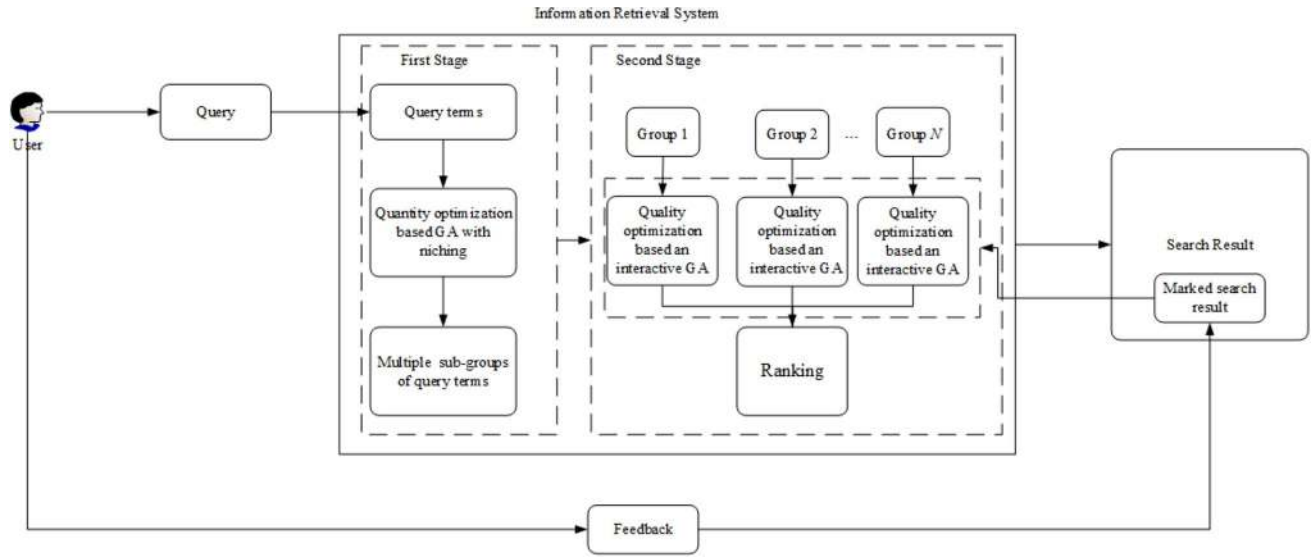

Fig. 2 The procedure of the information retrieval system

marked search result makes the optimization process towards to the optimal combination of query weights constantly. After this stage, several groups of sequential documents are acquired. All of these retrieved documents are integrated in descending order according to their frequencies to gain the best sequence of retrieved documents for user's browse. The flowchart of stage 1 and stage 2 are shown in Figs. 3 and 4 apart.

In order to clearly describe IMGA, two stages are detailed in the next subsections.

Stage 1: quantity control stage

The purpose of the quantity control stage is to decrease the scale of the optimization problem and control the number of retrieved documents in a suitable range by a multimodal GA. All the possible combinations of the query terms form the search space of the GA. As the aim is to find multiple combinations of query terms instead of a single combination, this paper proposes to use a special multimodal GA with niching. In this GA, if the number of documents retrieved by a query is acceptable, the fitness value of the query would be high. Otherwise, when a query returns too many or too few retrieved documents, it will be given a very low fitness value. Details of GA and niching strategy are described as follows.

Objective function

In the process of evolution, for each query combination or solution q_i, its fitness value is calculated as follows.

$$f(qi) = \begin{cases} \text{acount}(qi) - A & \text{account}(qi) \le B \\ \infty & B \le \text{account}(qi) \le B' \\ \text{acount}(qi) - A & B' \le \text{account}(qi) \end{cases}, \tag{7}$$

where account (q_i) denotes the number of retrieved documents by the query vector $\overrightarrow{q_i}$; A denotes a penalty value needed to be subtracted when the number of retrieved documents by a query vector is not in a certain interval; B and B' define a reasonable range of the number of retrieved documents, i.e., $[B, B']$. When the number of retrieved documents is in the range, the solution will be given an extremely high fitness value. The values of A, B, and B' are all associate with the size of the document database. ∞ denotes a maximum fitness value to flag that retrieval results by a query combination are promising.

Multimodal GA with niching method

Different from traditional GA-based query optimization methods that only return one solution, the first stage of the proposed approach aims to acquire several optimal solutions whose numbers of retrieved documents are suitable. We propose a multimodal GA with a niching method to obtain multiple solutions simultaneously. In the GA, closely spaced solutions gather to form multiple crowding. In other words, the solutions that are close in physical space constitute a crowding. Due to the high probability of a good solution to be found near a similarly good performing solution, this paper adopts the crowding technique to implement the niching method. Each crowding search for feasible solutions and

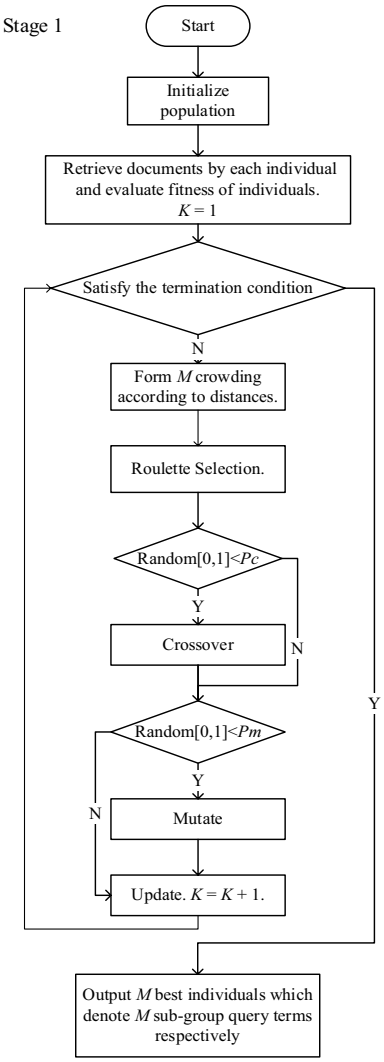

Fig. 3 The flowchart of the quantity control stage

multiple promising solutions will be obtained simultaneously by different crowding.

The flow of quantity control stage

At the quantity control stage, the value of the gene G_i of the binary-coded chromosome represents whether the query term is in the query vector. If the query vector contains the term t_i, then G_i is 0, else G_i is 1. N chromosomes are randomly generated to form an initial population. During GA evolution, parent chromosomes are selected by roulette selection and offspring are generated through real-valued multipoint crossover and polynomial mutation. In order to increase the diversity of the population, we take the niching strategy to updating the population and choose multiple chromosomes that perform promisingly. Based on a multimodal genetic algorithm, it is possible to maintain the diversity of the solution, as well to achieve high global optimality. The flow of the quantity stage is as follows.

1. N chromosomes are generated stochastically to form the initial population P_0, where the chromosome dimension is S which represents the number of initial query terms. Let $K=1$, where K counts the number of iterations of GA.
2. Calculate each chromosome's fitness based on the formula (7).
3. Form M crowding according to distances of chromosomes in the physical space. Roulette selection strategy is adopted to select pairs of chromosomes in each crowding and generate a random number m1 for each pair of chromosomes. If m_1 is less than the crossover rate P_c, randomly generate the number and the locations of crossover points. At each crossover-point, exchange their genes with each other.
4. Randomly generate a real value n_1 between 0 and 1 for each chromosome in each crowding, and if n_1 is less than the mutation rate P_m, the chromosome will perform mutation. That is, generate a random number of randomly-located mutation points. At each mutation-point, if the gene is 1, then it is transferred into 0. And vice, it is transferred into 1.
5. Choose the optimal chromosome in each crowding and compare these M chromosomes with the worst M chromosomes in the old population. The M chromosomes whose fitness values are higher replace the M poor ones. Let $K=K+1$.
6. If there are M chromosomes in the population that have the best fitness values, i.e., α, then stop it and go to Step 7, otherwise, go to Step 2. If K is equal to the maximum number of iterations, then the top M chromosomes with the largest fitness value are the best solutions and go to Step 7.
7. Convert M best chromosomes into M sub-groups of query terms.

The above process is summarized in Algorithm 2, where N is the population size. P_c and P_m are the crossover rate and the mutation rate respectively; M is the number of niches in this stage.

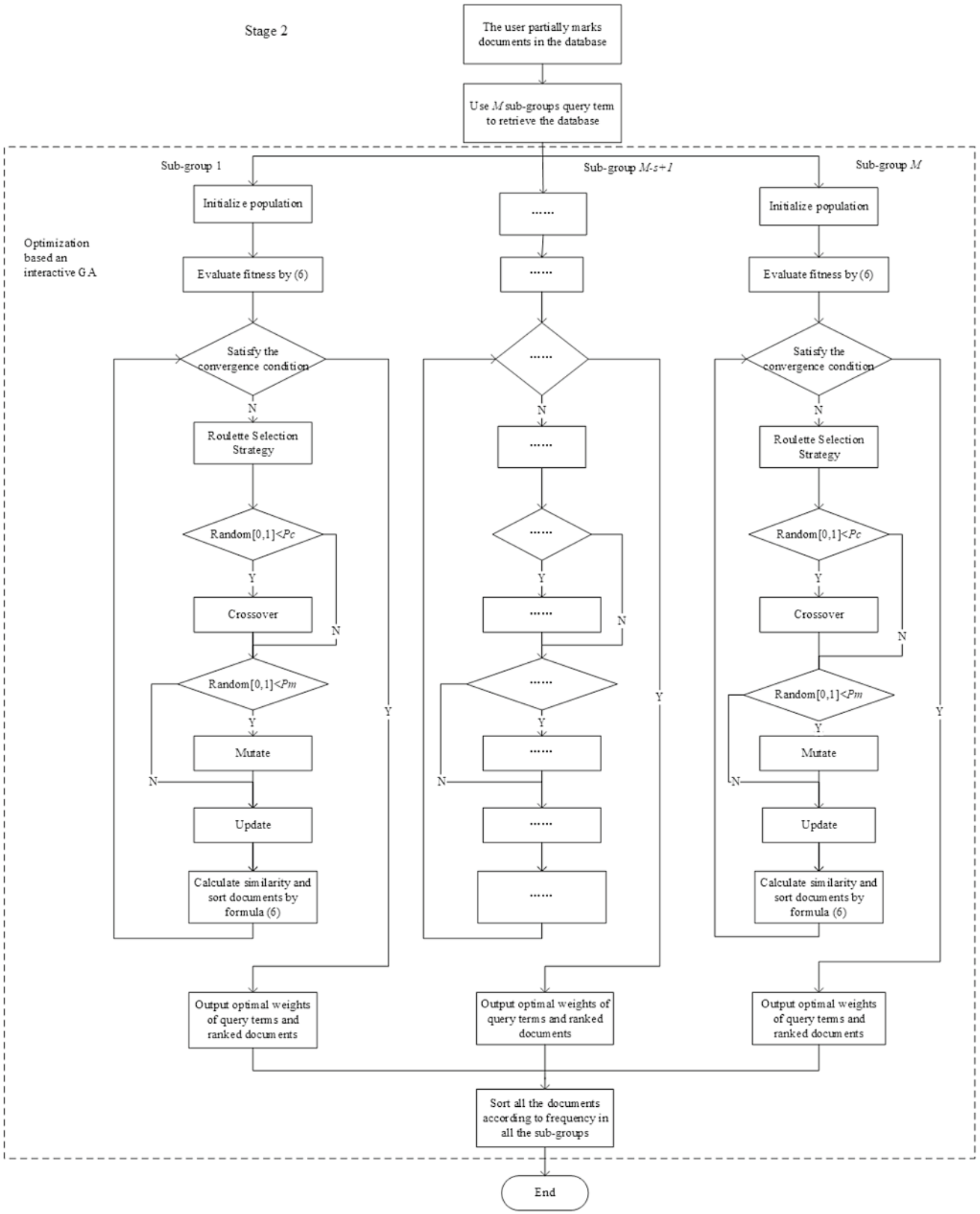

Fig. 4 The flowchart of the quality optimization stage

Algorithm 2: Quantity control algorithm

Input: Population P, niches number M, population size N, fitness function $f()$

Initialize P(I);
for i = 1 to $Max\_iterations$ **do**
  Calculate fitness f(P(I));
  Form M crowding by distance;
  for j=1 to N **do**
    Select();
    if(random() < P_c)
      Crossover();
    end if
    if(random() < P_m)
      Mutation();
    end if
    for k= 1 to M **do**
      Choose the optimum I_k;
    end for
    Calculate fitness f(P(I));
    Update();
    end for
end for

Output: The M best-so-far solutions

Stage 2: quality optimization stage

At this stage, all combinations of query terms acquired in the above stage are regarded as independent sub-queries. Each sub-query can be expressed as a query vector in VSM. Incorporating with the user's feedback mechanism, each query vector evolved as an individual in the population is optimized by an interactive GA. A surrogate model is built to evaluate the fitness values of individuals. Weighs of query terms optimized and the sequence of retrieved documents is yielded in the end. The details of the algorithm are described as follows.

Objective function

At the quality optimization stage, M sub-queries obtained in the previous stage are expressed as M query vectors. Weights of the query vectors are optimized by an interactive GA with the relevance feedback mechanism. The system sorts the documents in descending order according to the degree of similarity between the query vector and documents. For example, the document whose vector has a high degree of similarity with the query vector is ranked before the documents whose vector has a low degree of similarity with the query vector. Replacing user evaluation, the fitness values are calculated by a mathematical model. The mathematical model is represented as the score of the sequence of retrieved documents. The mathematical model of the objective function in interactive GA is described as formula (6).

The flow of quality optimization stage

The weights of query terms in each combination obtained in the quantity control stage are optimized by an interactive GA independently in this stage. After optimization, each sub-process can get a promising combination of weights of the query terms and a sequence of retrieved documents. That is, corresponding to M sub-queries, M sets of weights and M sequences of documents will be finally obtained. All the documents that appear in M sequences are sorted in descending order according to the document that frequency occurred in M sequences. At last, we can get a sequence of retrieved documents for the user's browse, and meanwhile, we can determine the weights of query terms. The detailed descriptions of the stage are as follows.

1. N' chromosomes are generated stochastically to form the initial population P_0' and each gene G_i of a chromosome gives the weight of the i-th query term t_i in the query vector. Let $K' = 1$.
2. Calculate degrees of similarity between query vectors and document vectors according to the formula (5). The fitness value of each chromosome is calculated by formula (6).
3. Adopt the roulette selection strategy to select parent chromosomes with higher fitness values into the next generation with a higher probability. Generate a random number m' for each pair of chromosomes. If m' is less than the crossover rate P_c', randomly generate the number and the location of crossover points. At each crossover-point, exchange their genes with each other.
4. Randomly generate a real value n' between 0 and 1 for each chromosome, and if n' is less than the mutation rate P_m', the chromosome accepts mutation. Generate the number and the location of mutation points. At each mutation-point, generate a random number which is in the range of 0 to 1 instead of the original gene.
5. If fitness values of all chromosomes which are in the old population are less than the fitness value of the one which is in the new population, replace the chromosome which performs worst in the old population by the new chromosome. Let $K' = K' + 1$.
6. If there is a chromosome with the best fitness (F = 1) in the new population, then the chromosome is the optimal solution, and to go to Step 7. Otherwise, go to Step 3. If K' is equal to the maximum number of iterations, then go to Step 7. The chromosome whose fitness value is the largest is the final result, otherwise, go to Step 3.
7. Use formula (5) to calculate degrees of similarity between the documents vectors and the user's queries to

get a descending order of retrieved documents in terms of degrees of similarity.

8. *M* groups of sequential documents are obtained. Then sort these documents in descending order all over again according to the document frequency appeared in *M* groups, and this sequence is the final sequence. Choose Top 10 or Top 20 of the retrieved documents to measure the performance of the proposed approach.

The above process is summarized in Algorithm 3, where *N'* is the population size; *D* is the database size; P_c' and P_m' are the crossover rate and the mutation rate respectively; *M* is the number of sub-query vectors, in other words, it is also the number of solutions returned in the first stage.

Algorithm 3: Quality optimization algorithm

Input: Population *P*, population size *N'*, database size *D*, fitness function *F* ()

```
Initialize P(I);
for i = 1 to Max_interations do
    for j = 1 to N' do
        for k = 1 to D do
            Similar(Q, d_k);
            Sort();
        end for
    Calculate fitness F(P(I));
    end for
    for j = 1 to N' do
      Select();
      if(random() < P_c)
          Crossover();
      if(random() < P_m)
          Mutation();
      for k = 1 to D do
          Similar(Q, d_k);
          Sort();
      end for
    Calculate fitness F(P(I));
    Update();
    end for
end for
```

Output: The best-so-far weight of query terms and the final sequence of retrieved documents

Experiment results

Experimental configuration

In the stage of the query weight optimization, due to the use of the relevance feedback mechanism, some documents need to be marked by their relevance to users, and common retrieval databases are inadequate for our work. As a result, a database that contains more than 2000 documents downloaded from the *IEEE XPLORE* [46] is used to test the performance of the proposed method. It includes 10 different categories of papers, and the number of each category of documents is between 185 and 226. The categories of documents are shown in Table 1. We use ten queries to analyze the performance of the proposed method. Table 2 shows ten test instances of queries.

The several vital parameters of the algorithm are set as follows. In the first stage, the population size is 30, and the dimension of individuals in the population is 10 because the number of query terms is 10 in each query. The maximum number of iterations of IMGA is 100. Referred to Lopez-Pujalte et al.'s experiments [21], the crossover rate and mutation rate are set to 0.8 and 0.2 respectively. The number of niches is set to 5, and the reasonable range of the number of retrieved documents is set from 20 to 50. In the second stage, the population size is 30, and the maximum number of iterations of the proposed algorithm is 100. The crossover rate and mutation rate are 0.8 and 0.2 respectively. To ensure fairness, IMGA and other methods for comparison both run 30 independent times.

Measure indicator

As two common performance indicators, the recall rate and the precision rate are used to measure the performance of algorithms in the IRS. The precision rate is the proportion of relevant documents in the returned result. The recall rate is the proportion of relevant documents that are returned to all relevant documents. The definition of the two indicators can be presented as follows

$$\begin{aligned} \text{Precision Rate} &= \frac{tp}{tp+fp}, \\ \text{Recall Rate} &= \frac{tp}{tp+fn}, \end{aligned} \tag{8}$$

where the definitions of *tp*, *fp,* and *fn* are presented in Table 3.

Table 1 Categories of documents

Category number	Category name	Category number	Category name
1	Evolutionary Algorithm	6	Neural Network
2	National Economy	7	Fuzzy set
3	Natural Language Processing	8	Face Recognition
4	Software Engineering	9	Information Retrieval
5	Object Oriented Database	10	Network Security

Table 2 Queries instances

Q0	Ant colony optimization particle swarm distributed genetic algorithm evolution computing
Q1	National economy economic finance model industry production development government contribution
Q2	Object oriented data set system structure rule model technology database
Q3	Mobile agent network graph security node autonomy synchronization localization adaptability
Q4	Natural language processing database data method artificial intelligence rule word
Q5	Software engineering knowledge area modeling analysis project education design requirement
Q6	Fuzzy set theory membership function
Boolean degree value operation logic	
Q7	Network security privacy firework access architecture control policy key attack
Q8	Face recognition human technology facial
image accuracy computer vision identity	
Q9	Information retrieval query weight frequency database search index retrieved private

Table 3 The relationship of relevant documents and the irrelevant documents

	Relevant	Nonrelevant
Retrieved	True positives (*tp*)	False positives (*fp*)
Not retrieved	False negatives (*fn*)	True negatives (*tn*)

Comparison results

Firstly, the Ricchio's relevance feedback algorithm [40] is implemented as a baseline to analyze the performance of the other state-of-the-art algorithms. Then, we conduct experiments on the comparison with the Chang-Chen-Liau's method [36], the TF/IDF method [47], and the TF-IDF-AP algorithm [48], and the with respect to the recall rates and precision rates of Top 10 and Top 20 retrieved documents. Moreover, a new ranking measure that combines the vector space measure and association rules technique (ranking measure with VSM and AR) which was proposed by Siham et al. [49] is also compared together.

Figures 5 and 6 illustrate the recall rates and the precision rates of the top 10 retrieved documents with respect to the proposed method and the five other algorithms. Figures 7 and 8 show the recall rates and the precision rates of the top 20 documents of the six algorithms.

From these figures, it can be observed that IMGA performs much better than the five other methods in terms of the recall rates and precision rates. While the Ricchio's relevance feedback method has just a higher recall rate of the top 20 than IMGA in the fourth query, and the TF/IDF method performs better than the four other algorithms on the precision rates of top 20 in the first query. The ranking measure with VSM and AR has better potential in the eighth query. Compared with the Chang-Chen-Liau's method, IMGA adds a process of quantity control and multiple sub-query terms are returned so that these weights of sub-query terms are

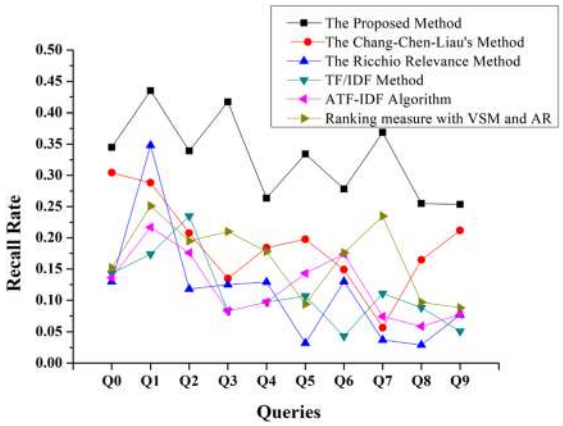

Fig. 5 Recall rates of the top 10 retrieved documents with respect to ten queries by different methods

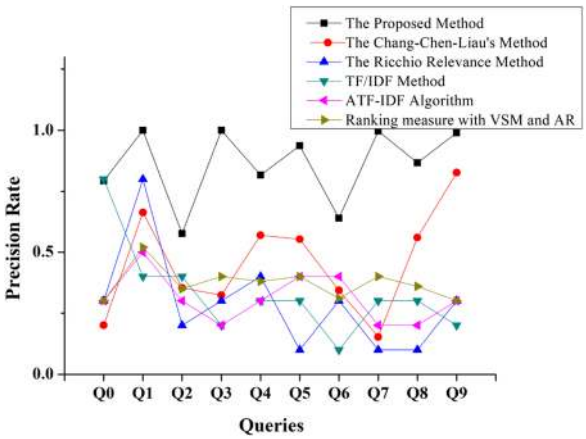

Fig. 6 Precision rates of the top 10 retrieved documents with respect to ten queries by different methods

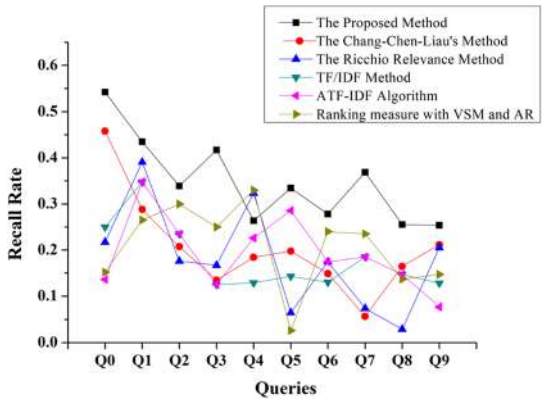

Fig. 7 Recall rates of the top 20 retrieved documents with respect to ten queries by different methods

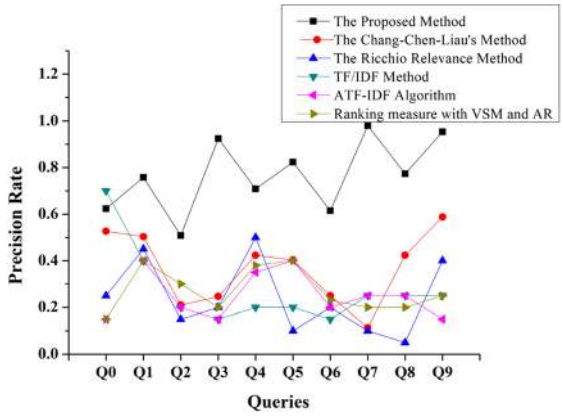

Fig. 8 Precision rates of the top 20 retrieved documents with respect to ten queries by different methods

optimized jointly in the weight optimization stage, which makes weights of query terms optimized more accurately.

Then we compare the average recall rates and the average precision rates of the six algorithms in terms of ten queries. Table 4 shows the average recall rates and the average precision rates of the top 10 and top 20 retrieved documents with respect to the four other methods. Obviously, IMGA achieves higher average recall rates and precision rates of the document retrieval, especially precision rates.

For a more intuitive observation of the experimental results,a significance test on the average recall rates and the average precision rates of 30 independent runs are implemented. Table 5 shows the t test with a confidence level of 0.95 of the recall rates and the precision rates between IMGA and the Chang-Chen-Liau's method by ten groups of user's queries. From the table, we can see that IMGA has performed much better than the Chang-Chen-Liau's method for all queries. The results of experiments have significant improvement in terms of the average recall rates and the average precision rates.

Analysis of the effectiveness of Niches

The number of niches is sensitive which is related to the population size. As the number of niches increases, more solutions can be obtained and the diversity of the solutions is achieved. Yet the way of merging the multiple solutions faces the challenge. On the other hand, the small number of niches will cause the diversity of solutions is not guaranteed. Therefore, choosing an appropriate niche size is significant to increase the performance of IMGA.

In order to analyze the impact of the niche size used in the GA during the first stage, we did a set of experiments on the five queries Q_0–Q_4, the number of niches is set to 3, 5, 7, 9, 12 respectively, and the performance of these algorithms is compared. Figures 9 and 10 show the recall rates and the precision rates of the top 10 documents with respect to different numbers of the niche (M) respectively. Figures 11 and 12 show the recall rates and the precision rates of the top 20 documents with respect to different numbers of the niche (M) respectively. From these four figures, we can see that IMGA performs better than any other when the number of niches is 5, whatever it returns the top 10 or 20 documents. What is more, when the population increases to 50, the situation remains the same. Figures 13, 14, 15 and 16 show the recall rates

Table 4 Average recall rates and average precision rates of the proposed system compared with other five methods

	TOP 10 Average Recall Rate	TOP 10 Average Precision Rate	TOP 20 Average Recall Rate	TOP 20 Average Precision Rate
IMGA	0.329	0.862	0.349	0.767
Chang-Chen-Liau's Method	0.190	0.455	0.205	0.369
The Ricchio's Relevance Feedback Algorithm	0.116	0.290	0.182	0.240
TF/IDF Method	0.113	0.330	0.182	0.275
TF-IDF-AP Algorithm	0.124	0.310	0.194	0.250
Ranking measure with VSM and AR	0.167	0.372	0.208	0.271

Table 5 Average recall rates and precision rates of different methods and the *t* test values of recall rates and precision rates between IMGA and the other methods with respect to ten users' queries

		Q_0		Q_1		Q_2		Q_3		Q_4		Q_5		Q_6		Q_7		Q_8		Q_9	
		k-means	t-test	*t* test	*t* test	k-means	t-test	k-means	*t* test	k-means	t-test	k-means	*t* test	k-means	t-test	k-means	*t* test	k-means	t-test	k-means	*t* test
The Proposed Method	TOP 10 Recall Rate	**0.345**	3.823	**0.435**	20.185	**0.339**	10.598	**0.417**	32.265	**0.263**	9.699	**0.334**	18.329	**0.278**	7.831	**0.369**	81.277	**0.255**	15.587	**0.254**	7.506
	TOP 10 Precision Rate	**0.793**	3.835	**1.000**	20.155	**0.577**	10.595	**1.000**	32.364	**0.817**	9.753	**0.937**	18.302	**0.640**	7.828	**0.997**	81.277	**0.867**	15.543	**0.990**	7.527
	TOP 20 Recall Rate	**0.542**	5.798	**0.435**	20.185	**0.339**	10.598	**0.417**	32.265	**0.263**	9.699	**0.334**	18.329	**0.278**	7.831	**0.369**	81.277	**0.255**	15.587	**0.254**	7.506
	TOP 20 Precision Rate	**0.623**	5.804	**0.758**	15.839	**0.508**	27.033	**0.925**	54.441	**0.708**	13.229	**0.823**	26.842	**0.615**	16.901	**0.980**	118.334	**0.773**	28.097	**0.953**	30.982
Chang-Chen-Liau's Method	TOP 10 Recall Rate	0.304	–	0.288	–	0.208	–	0.135	–	0.184	–	0.198	–	0.149	–	0.057	–	0.165	–	0.212	–
	TOP 10 Precision Rate	0.200	–	0.663	–	0.353	–	0.323	–	0.570	–	0.553	–	0.343	–	0.153	–	0.560	–	0.827	–
	TOP 20 Recall Rate	0.458	–	0.288	–	0.208	–	0.135	–	0.184	–	0.198	–	0.149	–	0.057	–	0.165	–	0.212	–
	TOP 20 Precision Rate	0.527	–	0.503	–	0.210	–	0.247	–	0.423	–	0.403	–	0.250	–	0.113	–	0.423	–	0.588	–
VSM	TOP 10 Recall Rate	0.130	–	0.348	–	0.118	–	0.125	–	0.129	–	0.032	–	0.130	–	0.037	–	0.029	–	0.077	–
	TOP 10 Precision Rate	0.300	–	0.800	–	0.200	–	0.300	–	0.400	–	0.100	–	0.300	–	0.1	–	0.1	–	0.3	–
	TOP 20 Recall Rate	0.217	–	0.391	–	0.176	–	0.167	–	0.323	–	0.065	–	0.174	–	0.074	–	0.029	–	0.205	–
	TOP 20 Precision Rate	0.250	–	0.450	–	0.150	–	0.200	–	0.500	–	0.100	–	0.200	–	0.1	–	0.05	–	0.4	–

Table 5 (continued)

		Q_0		Q_1		Q_2		Q_3		Q_4		Q_5		Q_6		Q_7		Q_8		Q_9	
		k-means	t-test	*t* test	*t* test	k-means	t-test	k-means	*t* test	k-means	t-test	k-means	*t* test	k-means	t-test	k-means	*t* test	k-means	t-test	k-means	*t* test
TF/IDF Method	TOP 10 Recall Rate	0.143	–	0.174	–	0.235	–	0.083	–	0.097	–	0.107	–	0.043	–	0.111	–	0.088	–	0.051	–
	TOP 10 Precision Rate	0.8	–	0.4	–	0.4	–	0.2	–	0.3	–	0.3	–	0.1	–	0.3	–	0.3	–	0.2	–
	TOP 20 Recall Rate	0.25	–	0.348	–	0.235	–	0.125	–	0.129	–	0.143	–	0.13	–	0.185	–	0.147	–	0.128	–
	TOP 20 Precision Rate	0.7	–	0.4	–	0.2	–	0.15	–	0.2	–	0.2	–	0.15	–	0.25	–	0.25	–	0.25	–
TF-IDF-AP Algorithm	TOP 10 Recall Rate	0.136	–	0.217	–	0.176	–	0.083	–	0.097	–	0.143	–	0.174	–	0.074	–	0.059	–	0.077	–
	TOP 10 Precision Rate	0.3	–	0.5	–	0.3	–	0.2	–	0.3	–	0.4	–	0.4	–	0.2	–	0.2	–	0.3	–
	TOP 20 Recall Rate	0.136	–	0.348	–	0.235	–	0.125	–	0.226	–	0.286	–	0.174	–	0.185	–	0.147	–	0.077	–
	TOP 20 Precision Rate	0.15	–	0.4	–	0.2	–	0.15	–	0.35	–	0.4	–	0.2	–	0.25	–	0.25	–	0.15	–
Ranking measure with VSM and AR	TOP 10 Recall Rate	0.152	–	0.251	–	0.195	–	0.21	–	0.178	–	0.094	–	0.176	–	0.235	–	0.097	–	0.088	–
	TOP 10 Precision Rate	0.3	–	0.52	–	0.35	–	0.4	–	0.38	–	0.4	–	0.31	–	0.4	–	0.36	–	0.3	–
	TOP 20 Recall Rate	0.152	–	0.265	–	0.3	–	0.25	–	0.33	–	0.026	–	0.24	–	0.235	–	0.137	–	0.147	–
	TOP 20 Precision Rate	0.15	–	0.4	–	0.3	–	0.2	–	0.38	–	0.4	–	0.23	–	0.2	–	0.2	–	0.25	–

The optimal results are bolded in the table

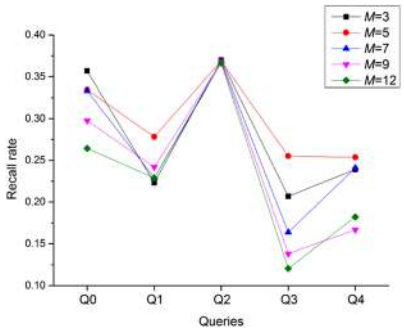

Fig. 9 Recall rates of top 10 retrieved documents with respect to five queries by different numbers of niches ($N=30$)

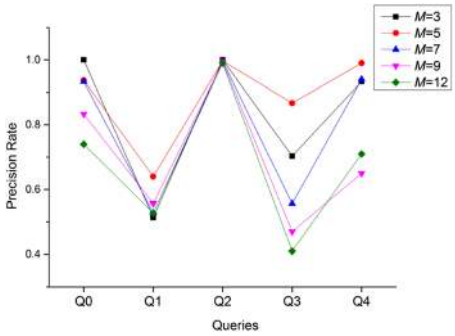

Fig. 10 Precision rates of top 10 retrieved documents with respect to five queries by different numbers of niches ($N=30$)

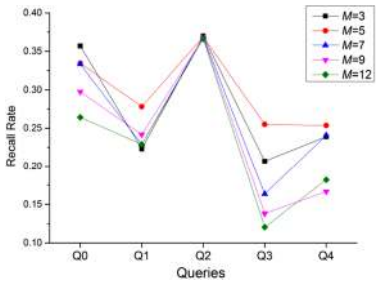

Fig. 11 Recall rates of top 20 retrieved documents with respect to five queries by different numbers of niches ($N=30$)

and the precision rates of the top 10 and top 20 retrieved documents when the population size increases to 50 with respect to different numbers of niches respectively. When the number of niches increases, the performance of IMGA gets worse for almost queries. In our algorithm, 5 niches are the best choice with respect to 30 and 50 population sizes.

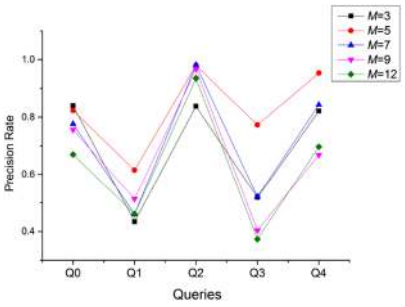

Fig. 12 Precision rates of top 20 retrieved documents with respect to five queries by different numbers of niches ($N=30$)

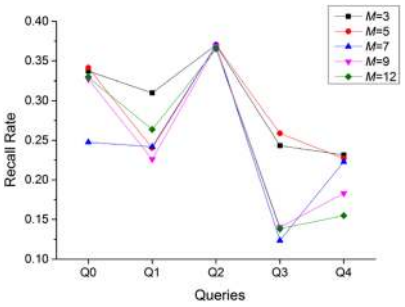

Fig. 13 Recall rates of top 10 retrieved documents with respect to five queries by different numbers of niches ($N=50$)

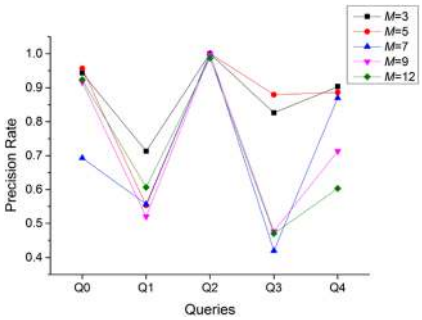

Fig. 14 Precision rates of top 10 retrieved documents with respect to five queries by different numbers of niches ($N=50$)

Conclusion

In this paper, we have proposed a two-stage information retrieval system based on an interactive multimodal genetic algorithm (IMGA) for query weight optimization. This system has a two-stage retrieval process: quantity control and quality optimization. In the quantity control stage, we adopt a multimodal genetic algorithm to obtain multiple feasible solutions to control the number of retrieved documents in an appropriate range. In this way,

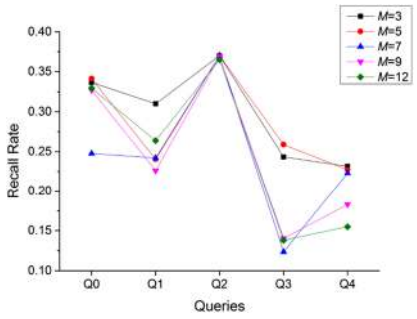

Fig. 15 Recall rates of top 20 retrieved documents with respect to five queries by different numbers of niches ($N=50$)

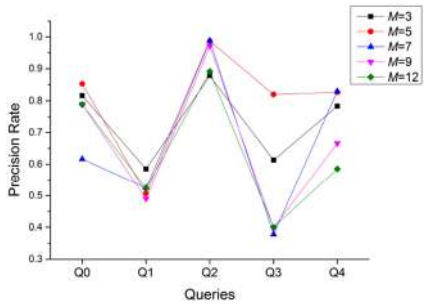

Fig. 16 Precision rates of top 20 retrieved documents with respect to five queries by different numbers of niches ($N=50$)

the algorithm decreases the dimension of the optimization problem to improve the accuracy of weight optimization in the next stage. In the quality optimization stage, an interactive genetic algorithm assisted with the user's relevance feedback mechanism is used to optimize the weights of query terms. Replacing user evaluation, a mathematical model is built to evaluate individuals. The retrieved document sequence which is suitable for the user's need is returned in the end. We did experiments on the document database to verify the effectiveness of the proposed IMGA. The experimental results show that the recall rates and the precision rates of the proposed two-stage method are much higher than Chang-Chen-Liau's method, the TF/IDF method, the TF-IDF-AP algorithm, and Ranking measure with VSM and AR.

In the future, we will continue to research the application of evolutionary algorithms in the field of information retrieval. The proposed method will be implemented on a larger dataset with more extensibility and flexibility. Other promising EAs will be studied to be embedded in the infomation retrieval system. What is more, the performance of the retrieval algorithm will be tested in another performance measures.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China under Grants 61976093 and 61873097. The research team was supported by the Guangdong-Hong Kong Joint Innovative Platform of Big Data and Computational Intelligence No. 2018B050502006, and the Guangdong Natural Science Foundation Research Team No. 2018B030312003. (Corresponding Author: Wei-Neng Chen, email: cwnraul634@aliyun.com).

References

1. Salton G, Mcgill MJ (2004) Introduction to modern information retrieval. Program 55(3):239–240
2. El-Bathy N, Azar G, El-Bathy M, Stein G (2011) Intelligent information retrieval lifecycle architecture based clustering genetic algorithm using SOA for modern medical industries.Electro/Information Technology (EIT) pp 1–7
3. Sanderson M, Croft WB (2012) The history of information retrieval research. Proc IEEE 100(6):1444–1451
4. Abualigah LMQ, Hanandeh ES (2015) Applying genetic algorithms to information retrieval using vector space model. Int J Comput Sci Eng Appl 5(1):19
5. Salton G, Mcgill MJ (1983) Introduction to modern information retrieval: McGraw-Hill. 41(4): 305–306
6. Banawan K, Ulukus S (2018) Multi-message private information retrieval: capacity results and near-optimal schemes. IEEE Trans Inf Theory 64(10):6842–6862
7. Junnila V, Laihonen T (2016) Information retrieval with varying number of input clues. IEEE Trans Inf Theory 62(2):625–638
8. Sun H, Jafar SA (2017) The capacity of private information retrieval. IEEE Trans Inf Theory 63(7):4075–4088
9. Tajeddine R, Gnilke OW, Rouayheb SE (2018) Private information retrieval from MDS coded data in distributed storage systems. IEEE Trans Inf Theory 64(11):7081–7093
10. Junnila V, Laihonen T (2014) Codes for information retrieval with small uncertainty. IEEE Trans Inf Theory 60(2):976–985
11. Nuij W, Milea V, Hogenboom F, Frasincar F, Kaymak U (2014) An automated framework for incorporating news into stock trading strategies. IEEE Trans Knowl Data Eng 26(4):823–835
12. Khemmarat S, Gao L (2017) Predictive and personalized drug query system. IEEE J Biomed Health Inform 21(4):1146–1155
13. Nassar MO, Mashagba FA, Mashagba EA (2013) Improving the user query for the boolean model using genetic algorithms. Int J Comput Sci Issues 8(5):66
14. Maron ME, Kuhns JL (1960) On relevance, probabilistic indexing and information retrieval. J ACM (JACM) 7(3):216–244
15. Salton G (1968) Automatic information organization and retrieval. McGraw Hill Text 8(1):1–2

16. Mistry K, Zhang L, Neoh SC, Lim CP, Fielding B (2017) A micro-GA embedded PSO Feature selection approach to intelligent facial emotion recognition. IEEE Trans Cybern 47(6):1496–1509
17. Han H, Lu W, Qiao J (2017) An adaptive multiobjective particle swarm optimization based on multiple adaptive methods. IEEE Trans Cybern 47(9):2754–2767
18. Mavrovouniotis M, Müller FM, Yang S (2017) Ant colony optimization with local search for dynamic traveling salesman problems. IEEE Trans Cybern 47(7):1743–1756
19. Cordón O, Herrera-Viedma E, López-Pujalte C, Luque M, Zarco C (2003) A review on the application of evolutionary computation to information retrieval ☆. Int J Approx Reason 34(2):241–264
20. Horng J-T, Yeh C-C (2000) Applying genetic algorithms to query optimization in document retrieval. Inf Process Manag 36(5):737–759
21. López-Pujalte C, Guerrero-Bote VP, Moya-Anegón FD (2003) Genetic algorithms in relevance feedback: a second test and new contributions. Inf Process Manag 39(5):669–687
22. López-Pujalte C, Guerrero-Bote VP, Moya-Anegón FD (2010) Order-based fitness functions for genetic algorithms applied to relevance feedback. J Am Soc Inform Sci Technol 54(2):152–160
23. Hideyuki T (2001) Interactive evolutionary computation: fusion of the capacities of ec optimization and human evaluation. Proc IEEE 89(9):1275–1296
24. Munetomo M, Bando S (2013) A scalable infrastructure of interactive evolutionary computation to evolve services online with data. In: IEEE International Conference on Big Data, pp 28–28
25. Guoyan Y, Zhen H, Chaoan L et al. (2006) The Application of Interactive Evolutionary Algorithm in Product Design. In: 6th World Congress on Intelligent Control and Automation, pp 6758–6762
26. Oliver A, Monmarche N, Venturini G (2002) Interactive Design of Web Sites with a Genetic Algorithm. In: Proceedings of the IADIS International Conference, pp 355–362
27. Funaki R, Sugimoto K, Murata J (2018) Estimation of influence of each variable on user's evaluation in interactive evolutionary computation. In: 9th International Conference on Awareness Science and Technology (iCAST), pp 167–174
28. Ohsaki M, Takagi H, Ohya K (1998) An input method using discrete fitness values for interactive GA. J Intell Fuzzy Syst 6:131–145
29. Lee JY, Cho SB (1999) Sparse Fitness Evaluation for reducing user burden in interactive genetic algorithm. Proceedings of the 1999 IEEE International Fuzzy Systems Conference, pp II-998–II1003
30. Watanabe Y, Yoshikawa T, Furuhashi T (2006) Proposal of Interactive Genetic Algorithm based on Evaluation of Paired Comparison. In: Proceedings of the 16th Intelligent System Symposium, pp 307–310
31. Sun X, Gong D (2009) Interactive genetic algorithms with individual's fuzzy and stochastic fitness. Chin J Electron 18(4):619–624
32. Wang SF, Wang SH, Wang XFJJODA et al (2003) Improved Interactive Genetic Algorithm Incorporating with SVM and Its Application. J Data Acquis Process 18(4):429–433
33. Li Y (2012) Adaptive learning evaluation model for evolutionary art. In: Proc. IEEE Congr. Evol. Comput., pp 1–8
34. Chugh T, Sindhya K, Hakanen J et al. (2015) an interactive simple indicator-based evolutionary algorithm (I-SIBEA) for multiobjective optimization problems. evolutionary multi-criterion optimization. Springer International Publishing
35. Gong D et al (2004) Hierarchical interactive evolutionary computation and its application in fashion design. Intell Control Autom 19(10):1117–1124
36. Chang YC, Chen SM (2006) A new query reweighting method for document retrieval based on genetic algorithms. IEEE Trans Evol Comput 10(5):617–622
37. Lopez-Pujalte C, Bote VPG (2002) A test of genetic algorithms in relevance feedback. Inf Process Manag 38(6):793–805
38. Bartell BT, Cottrell G, Belew R (1998) Optimizing similarity using multi-query relevance feedback. J Am Soc Inf Sci 49(8):742–761
39. Guerrero-Bote VP (2003) Order-based fitness functions for genetic algorithms applied to relevance feedback. J Am Soc Inform Sci Technol 54(2):152–160
40. Salton G (1971) The SMART Retrieval System—Experiments in Automatic Document Processing. Prentice-Hall, Inc.
41. Zhu Z, Chen X, Zhu Q, Xie Q (2007) A GA-based query optimization method for web information retrieval. Appl Math Comput 185(2):919–930
42. Holland JH (1975) Adaptation in natural and artificial systems. Q R Biol 6(2):126–137
43. Weiguo S, Swift S, Leishi Z, Xiaohui L (2005) A weighted sum validity function for clustering with a hybrid niching genetic algorithm. IEEE Trans Syst Man Cybern Part B 35(6):156–1167
44. Yang Q, Chen WN, Yu Z, Gu T, Li Y, Zhang H (2017) Adaptive multimodal continuous ant colony optimization. IEEE Trans Evol Comput 21(2):191–205
45. Li X, Epitropakis MG, Deb K, Engelbrecht A (2017) Seeking multiple solutions: an updated survey on niching methods and their applications. IEEE Trans Evol Comput 21(4):518–538
46. [DB/OL] http://ieeexplore.ieee.org/
47. Matta D, Verma M (2013) Evaluating Relevancy Of Words In Document Queries Using Vector Space Model. J Eng Comput Appl Sci (JEC&AS) 2(6): 2319–5606
48. Chen J, Chen C, Liang Y (2016) Optimized TF-IDF algorithm with the adaptive weight of position of word. Advanc Intell Syst Res 133:114–117
49. Jabri S, Dahbi A, Gadi T, Bassir A (2018) Ranking of text documents using TF-IDF weighting and association rules mining. In: 2018 4th International Conference on Optimization and Applications (ICOA), Mohammedia, 10(1109):1–6