Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 4953280, 11 pages
https://doi.org/10.1155/2017/4953280

Research Article

A Method for Entity Resolution in High Dimensional Data Using Ensemble Classifiers

Liu Yi,[1] Diao Xing-chun,[1] Cao Jian-jun,[2] Zhou Xing,[1] and Shang Yu-ling[1]

[1] *PLA University of Science and Technology, Nanjing, Jiangsu 210007, China*
[2] *Nanjing Telecommunication Technology Institute, Nanjing, Jiangsu 210007, China*

Correspondence should be addressed to Cao Jian-jun; jianjuncao@yeah.net

Received 31 October 2016; Accepted 17 January 2017; Published 15 February 2017

Academic Editor: Yaguo Lei

In order to improve utilization rate of high dimensional data features, an ensemble learning method based on feature selection for entity resolution is developed. Entity resolution is regarded as a binary classification problem, an optimization model is designed to maximize each classifier's classification accuracy and dissimilarity between classifiers and minimize cardinality of features. A modified multiobjective ant colony optimization algorithm is employed to solve the model for each base classifier, two pheromone matrices are set up, weighted product method is applied to aggregate values of two pheromone matrices, and feature's Fisher discriminant rate of records' similarity vector is calculated as heuristic information. A solution which is called complementary subset is selected from Pareto archive according to the descending order of three objectives to train the given base classifier. After training all base classifiers, their classification outputs are aggregated by max-wins voting method to obtain the ensemble classifiers' final result. A simulation experiment is carried out on three classical datasets. The results show the effectiveness of our method, as well as a better performance compared with the other two methods.

1. Introduction

Entity resolution (ER) is to find out the ambiguous denotations which refer to the same real world entity. ER has been researched for a long time and it is a crucial stage in data cleaning. It is also called record linkage in statistics, disambiguation in information retrieval [1], data matching and coreference disambiguation in computer science, and so forth [2].

In the big data era, ER's researches for big data have become a hot point [3–5]. Big data has some new characteristics such as big volume, fast velocity, and high dimension. And high dimension is one of the most important characteristics [6], which brings great challenges for current ER's methods. There are two existing ways to handle the high dimensional data: one is to adopt parallel technologies, such as crowdsourcing and MapReduce, to reduce time for computing similarity vector of each candidate pair by all features of two records so as to identify whether the two records are matches (similar) or nonmatches (dissimilar). The other way is to use feature selection method to reduce dimensions and calculate similarity vector for ER.

Crowdsourcing is a new ER approach [7] which distributes candidate records to human workers to identify matching records [8]. Abboura et al. [9] used crowdsourcing to find out matching records in training data and created matching dependencies by Apriori algorithm to identity matching records in testing data. Zhang et al. [10] proposed CrowdLink model to reduce human workers' difficulty in identifying similar records, and it could also tolerate human mistakes at the same time. The existing problems of crowdsourcing method are instability of human workers and the dependency on ER questions' setting way. Besides, human's judgment cannot guarantee the right answers, so algorithms need to handle with that.

Priya et al. [11] adopted Hadoop framework to design an ER system for stream data; they used thirteen similarity functions to measure similarity of records and generated matching rules by matching records based on the average of thirteen similarity functions. In order to solve ER problem

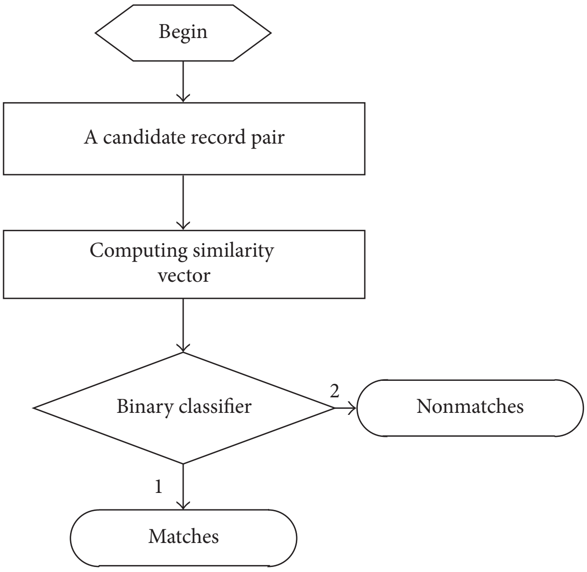


Figure 1: Process of entity resolution.

in high dimensional data by a parallel way, Fries et al. [12] proposed a method called parallel similarity self-join based on MapReduce, which simply applied MapReduce framework to reduce computing time. Besides, there are other ER models based on MapReduce such as HadoopDB [13], Hadoop++ [14], and PACT [15]. Though it is effective to reduce computing time through parallel technologies, it has a low efficiency to obtain similarity vectors of records by using all features. Besides, high dimensional data could likely contain irrelevant features which may obscure the effect of the relevant ones [16]. It is inefficient to improve algorithm's performance on ER by using all features which may even include noise information.

In order to overcome shortage of human participation for ER in high dimensional data, Cheng et al. [17] selected the key features from entity's descriptive features, and they are sorted and recombined to improve their readability and discriminant ability. For coping with the incomplete and incorrect publication information's effect on name disambiguation, Song et al. [18] employed the Named Entity Recognition model to choose organizations' features with publication features to improve ER's performance. Gueereiro et al. [19] developed a name disambiguation framework and applied five different types of features to implement ER process. Treerapituk et al. [20] adopted random forest method based on binary classifiers to select features, and experiment results showed that only using some key features could improve the ER's accuracy. Current feature selection methods cannot make full use of the rich information of high dimensional data effectively because the final number of selected features is usually no more than fifteen [21]. Besides, different classification results may be found in different subspaces, so global filtering of features is not sufficient (local feature relevance problem) [16]. Now, there are very few works considering big data high dimensional characteristic to address ER problem [22].

In order to overcome the difficulties of current ER methods for high dimensional data, we propose an ensemble learning method based on feature selection in this paper. We regard ER process as a binary classification problem, that is, classifying a record pair as matches (similar) or nonmatches (dissimilar). Then we define the measures of classification performance and similarity between binary classifiers which employ SVM as base classifier. Three objectives are applied to optimize each base classifier's performance, that is, maximizing classifier's classification accuracy rate, maximizing dissimilarity between classifiers, and minimizing cardinality of features. A modified multiobjective ant colony optimization (MOACO) is designed to solve the optimization model to select complementary feature subset which is adopted to train base classifier. In the end, several binary base classifiers are combined by ensemble learning method to improve performance for ER in high dimensional data.

The paper is organized as follows: Section 2 describes ER's concept and process; Section 3 defines binary classifier's classification performance and similarity measures; Section 4 shows our method's components and how it works; Section 5 makes an experiment to evaluate our method compared to other two methods; Section 6 closes with conclusions and discussions.

2. ER's Processing Description

According to machine learning technologies, ER's methods can be divided into four categories: methods based on probability; methods based on rules; methods based on clustering; and methods based on classification [23, 24].

In this paper, we regard ER as a binary classification problem, and the results contain two classes: matches class and nonmatches class, respectively. Firstly, a candidate record pair is represented by a similarity vector which is obtained by computing their corresponding features' similarity. Secondly, the similarity vector is adopted as input of a binary classifier to identify whether they are matches or nonmatches. The above process can be described as Figure 1.

Without loss of generality, we only discuss records in one table. Suppose a record has n features, and the set of features is denoted as $A = \{a_1, a_2, \ldots, a_n\}$. The value of kth feature of ith record is denoted as v_{ki}, $k = 1, 2, \ldots, n$; then the ith record can be denoted as $r_i = (v_{1i}, v_{2i}, \ldots, v_{ni})$. And the kth feature's similarity value between r_ith and r_jth records can be denoted as $s_{kij} = f_k(v_{ki}, v_{kj}) = f_k(v_{kj}, v_{ki})$; then we obtain the similarity vector of r_ith and r_jth records $V_{ij} = (s_{1ij}, s_{2ij}, \ldots, s_{nij})$. At last V_{ij} is input to the binary classifier which identifies the r_ith and r_jth records as matches or nonmatches.

3. Measures of Classification Performance and Classifier's Similarity

3.1. Measures of Binary Classifier's Classification Performance. As discussed above, we regard ER as a binary classification problem in this paper, so how to measure classifier's performance is the key to improving its classification effectiveness.

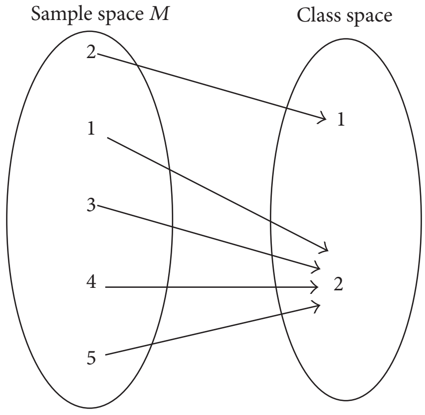


FIGURE 2: Mapping diagram of a binary classifier.

In this section, we define some indicators to measure binary classifier's performance.

Generally speaking, a binary classifier can be seen as a mapping function which is a many-to-one mapping classifier from sample space to class space. That is to say, it maps M classes in sample space to two classes in class space. As Figure 2 shows, the binary classifier maps two or more classes in sample space to the same class in class space. In order to facilitate the statement, we regard matches as positive class and nonmatches as negative class.

Now we define the measures of binary classifier for high dimensional ER.

Classification accuracy rate P:

$$P = \frac{\text{Number of samples correctly classified}}{\text{Number of samples}} \times 100\%. \quad (1)$$

False alarm rate R_{fa}:

$$R_{\text{fa}} = \frac{\text{Number of matches classified as nonmatches}}{\text{Number of matches}} \times 100\%. \quad (2)$$

Fault is not recognized rate R_{fn}:

$$R_{\text{fn}} = \frac{\text{Number of nonmatches classified as matches}}{\text{Number of nonmatches}} \times 100\%. \quad (3)$$

The binary classifier's output distribution matrix:

$$p = [p_{ii'}], \quad i, i' = 1, 2, \quad (4)$$

where

$$p_{ii'} = \frac{\text{Number of samples in class } i \text{ classified as class } i'}{\text{Number of samples in class } i} \times 100\%. \quad (5)$$

Then p_{ii} $(i = 1, 2)$ is the classification accuracy rate of class i, and it can be calculated by

$$p_{ii} = 1 - p_{ii'}. \quad (6)$$

Then the classification accuracy rate P can be given by

$$P = P_i p_{ii} + P_{i'} p_{i'i'}, \quad (7)$$

where P_i is the samples' prior possibility in class i. Given a testing set of samples, P_i can be expressed as

$$P_i = \frac{N_i}{N_i + N_{i'}}, \quad (8)$$

where N_i is number of samples in class i and $N_{i'}$ is number of samples in class i'.

So the false alarm rate R_{fa} of binary classifier can be written as (1 denoted matches; 2 denoted nonmatches)

$$R_{\text{fa}} = p_{12}. \quad (9)$$

The fault is not recognized rate R_{fn} of binary classifier can be expressed as (1 denoted matches; 2 denoted nonmatches)

$$R_{\text{fn}} = p_{21}. \quad (10)$$

Then P, R_{fa}, and R_{fn} have a relation represented by

$$1 - P = P_1 R_{\text{fa}} + (1 - P_1) R_{\text{fn}}. \quad (11)$$

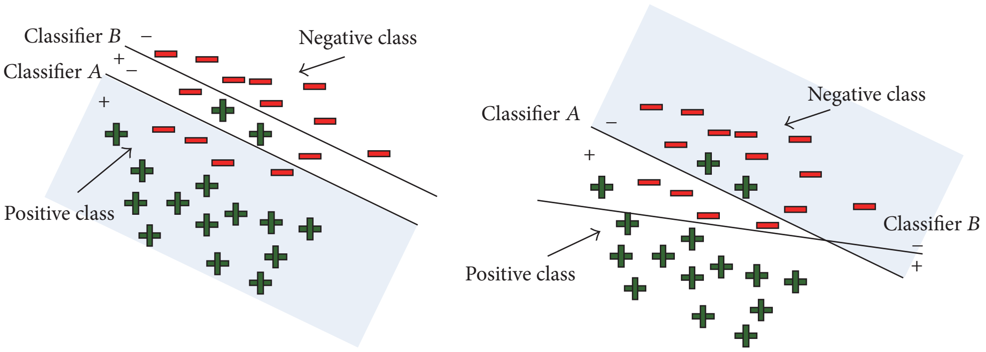
Based on (11) and the definitions of R_{fa} and R_{fn}, we can find that they have a conflict with each other. A high R_{fa} will lead to a low R_{fn}, and vice versa. And the classification accuracy rate can reflect both R_{fa} and R_{fn} effectively, so we adopt classification accuracy rate to measure binary classifier's classification performance.

3.2. Measures of Binary Classifier's Similarity. When many classifiers exist, the classifiers which have similar outputs may have similar results under the same ambiguous data, so their combination cannot improve the classification performance. On the other hand, when there are some differences between classifiers' results, their ensemble will improve the classification performance to a certain extent, which is illustrated in Figure 3.

In Figure 3(a), classifiers A and B have similar classification results, so their ensemble cannot identify the four negative samples in shadow correctly. But in Figure 3(b), classifiers A and B have a difference between their outputs, so their ensemble only cannot identify the two positive samples in shadow correctly, which lead to an improvement in classification accuracy rate. In this case, we say that their classification results are complementary to each other.

Given a set of samples and feature subset *subset* (denoting feature vectors of samples), if we use feature vectors of training samples to train a binary classifier Λ and feature vectors of testing samples to test it, then we can map *subset* into a fixed binary classifier Λ_{subset} and an output distribution matrix p:

$$\Lambda(subset) = (\Lambda_{subset}, p). \quad (12)$$



(a) Example of two similar classifiers

(b) Example of two dissimilar classifiers

Figure 3: Examples of classifiers' ensemble.

So the classifiers which have complementary results can be obtained by using complementary feature subsets to train them. It is also clear that the similarity of binary classifier Λ_{subset} can be measured by the similarity of subset and p which are also called input similarity and output similarity, respectively.

Next we define the measures of similarity between classifiers (similar to diversity) [25].

Definition 1. Two binary classifiers' input similarity is defined as the similarity of input feature subsets. For feature subsets $subset_1$ and $subset_2$ of two binary classifiers (both are nonempty), we adopt Tanimoto distance to measure their similarity [26]:

$$S_t\left(subset_1, subset_2\right) = 1 - \frac{\left|subset_1\right| + \left|subset_2\right| - 2\left|subset_1 \cap subset_2\right|}{\left|subset_1\right| + \left|subset_2\right| - \left|subset_1 \cap subset_2\right|}. \tag{13}$$

From (13), we can know that $S_t \in [0,1]$. When $S_t = 0$, it means that there is no common element between two subsets. When $S_t = 1$, it means that two subsets are identical and the classifiers trained by their corresponding feature vectors of training samples are also identical. So the bigger S_t, the higher similarity of two feature subsets and the two classifiers' input.

Definition 2. Two binary classifiers' output similarity is defined as the similarity of classifiers' output distribution matrix. Given two binary classifiers' output distribution matrices $p' = [p'_{ii'}]$ and $p'' = [p''_{ii'}]$, where $i = 1,2$, $i' = 1,2$, we use normalized Pearson's correlation coefficient to measure their similarity:

$$S_c\left(p', p''\right) = \frac{1}{2}\left(1 + \frac{\sum_{i=1}^{2}\sum_{i'=1}^{2}\left(p'_{ii'} - \overline{p'}\right)\left(p''_{ii'} - \overline{p''}\right)}{\sqrt{\sum_{i=1}^{2}\sum_{i'=1}^{2}\left(p'_{ii'} - \overline{p'}\right)^2 \sum_{i=1}^{2}\sum_{i'=1}^{2}\left(p''_{ii'} - \overline{p''}\right)^2}}\right), \tag{14}$$

where $\overline{p'}$ and $\overline{p''}$ are the average of matrices p' and p'', respectively:

$$\overline{p'} = \frac{1}{4}\sum_{i=1}^{2}\sum_{i'=1}^{2} p'_{ii'}, \tag{15}$$

where $S_c \in [0,1]$. When $S_c = 1$, the two output distribution matrices are full positive relative and the classification results of corresponding classifiers are identical. When $S_c = 0$, it means that the two output distribution matrices are full negative relative and the classification results of corresponding classifiers are completely different from each other.

Theorem 3. *If $\Lambda(subset_1) = (\Lambda_{subset_1}, p_1)$, $\Lambda(subset_2) = (\Lambda_{subset_2}, p_2)$, and $S_c(p_1, p_2) < 1$, then one has $S_t(subset_1, subset_2) < 1$.*

Proof. Suppose $subset_1 = subset_2$; then we have $p_1 = p_2$ according to (12) and precondition; that is to say, if $S_t(subset_1, subset_2) = 1$, then $S_c(p_1, p_2) = 1$. So the hypothesis is wrong and the theorem is correct. □

From above analysis, we know that the output similarity of classifiers is stronger than their input similarity, so we adopt output similarity of classifiers to measure the similarity of classifiers. It is also clear that the more dissimilarity between classifiers which are trained by complementary subsets, the more complementarity between them, which may improve utilization rate of high dimensional data features.

4. Ensemble Classifiers Based on Feature Selection

There is no single classification algorithm that can solve all kinds of problems according to "no free lunch theorems" [27]. The reason is that each classifier typically has a different domain of competence under different problems and conditions. However, if we have a pool of different classifiers and adopt ensemble learning method to combine them, which is a way independent of algorithms to improve classification performance, the classification accuracy would be improved efficiently [28]. So we use ensemble learning method to combine several binary classifiers to improve ER's performance.

4.1. Model of Ensemble Classifiers. For high dimensional ER, the proposed model based on feature selection which is used to train binary classifiers can be described as follows: Suppose there are ensemble classifiers which contain L (L is odd number) binary classifiers, P_l denotes the classification accuracy rate of lth classifier, q_l represents cardinality of lth classifier's input feature subset, and the input features of lth classifier are selected according to following optimization objectives:

$$\text{(i)} \quad \max \quad P_l \tag{16}$$

$$\text{(ii)} \quad \max \quad \left\{1 - \max_{j=1}^{l-1} \left\{S_c\left(p_j, p_l\right)\right\}\right\} \tag{17}$$

$$\text{(iii)} \quad \min \quad q_l. \tag{18}$$

Equation (16) maximizes the classification accuracy rate of lth classifier, which leads to a not poor classifier for ER. Equation (17) maximizes the dissimilarity between lth classifier and other $l-1$ classifiers, which means that the selected features constitute a complementary feature subset. Equation (18) minimizes the number of selected features, which leads to a better efficiency for classification. Since the three objectives have a conflict with each other, they are a multiobjective optimization problem.

The decision function adopted by our ensemble classifiers is "Max-Wins" voting method. Suppose f_{nl} is the output of the lth binary classifier for nth sample, and it can be denoted by

$$f_{nl} = \begin{cases} 1, & \text{matches} \\ 2, & \text{nonmatches.} \end{cases} \tag{19}$$

If the ensemble classifiers contain L (L is odd number) binary classifiers, then the final output of nth sample is decided by

$$\text{class} = f_{n1} \oplus f_{n2} \oplus \cdots \oplus f_{nl}, \tag{20}$$

where $\oplus$ denotes XOR operation. That is to say, we choose majority classification results as the final output.

4.2. MOACO for Solving Ensemble Classifiers' Model. A multiobjective optimization problem contains two or more than two objectives which have no order of priority. It can be stated as follows [29]:

$$\min \quad \mathbf{F}(\mathbf{x}) = \left(f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})\right)^T, \quad \mathbf{x} \in \Omega, \tag{21}$$

where the decision vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ belongs to the nonempty decision space Ω, the objective function vector $\mathbf{F} : \Omega \rightarrow \Gamma$ consists of m ($m \geq 2$) objectives, and Γ is objective space.

The solutions of multiobjective optimization problem are called Pareto optimal solutions which could not be further improved on any objective without harming the rest of objectives. And decision maker should choose one solution based on his or her preference. The goal of multiobjective optimization is to approximate the Pareto front which is composed of Pareto optimal solutions in the objective space [30].

MOACO is an excellent multiobjective evolutionary algorithm based on the foraging behavior of real ant species. The indirect communication of real ants in the colony uses pheromone trail lying on the ground to guide others to find the shortest path between their food source and the nest, which is called information positive feedback mechanism. As MOACO has that mechanism inherently and adopts reactive search optimization principle, that is, "learning while optimizing" principle [31], it has a better performance of searching Pareto optimal solutions than other multiobjective evolutionary algorithms especially for multiobjective combinatorial optimization problems [32]. Besides, it also has many other advantages such as robust, distributed computing and combination with other certain heuristics, which make it easily generalized.

As our model is a feature selection problem which is also a typical combinatorial optimization problem, we adopt a modified MOACO for solving it.

In order to solve the model by MOACO, we analyze it as follows:

(i) For a given binary classifier, it will have a better computing efficiency and classification accuracy rate if the number of its input features is set between 5 and 10. But we need more features to train the classifier to improve its performance in a high dimensional data. So the number of input features q is set between 1 and 20 without missing lower bound.

(ii) Transform (16) and (17) into two objectives f_1 and f_2 which must be optimized under a fixed cardinality of features (as MOACO must determine the number of selected features firstly). So the problem solved by MOACO is converted into a maximization optimization problem.

$$\max \quad \mathbf{F} = \left(f_1, f_2\right), \tag{22}$$

where $f_1 = \max P_l$ and $f_2 = \max\{1 - \max_{j=1}^{l-1}\{S_c(p_j, p_l)\}\}$. For a given binary classifier, we set up an archive which records its Pareto optimal solutions and the archive lasts until the cardinality of features increases to the upper bound. MOACO is adopted to search Pareto optimal solutions (feature subsets) and it also sets up an archive under a fixed feature number (determine q value firstly). After one cycle, the archive of MOACO is updated by solutions which are found by all ants. There are two situations for updating archive of MOACO. The value of objective f_2 does not exist when solving the first classifier, so the solutions are compared by values of objective f_1, and the solution with the highest f_1 is recorded. When solving the lth classifiers ($l > 1$), the Pareto relations between solutions are determined by their values of objectives f_1 and f_2, and the Pareto relations are applied to update the archive of MOACO (solution i replaces solution j if no component of $\mathbf{F}(i)$ is smaller than the corresponding component of $\mathbf{F}(j)$ and at least one component is larger). When iteration reaches to the upper bound, MOACO finishes and the solutions in its archive are employed to update current binary classifier's archive and the method for updating the archive is the same as MOACO's. After one cycle, the cardinality of features increases by one and a new MOACO starts.

(iii) When MOACO finishes for a given binary classifier under all q values, we need to choose one solution (feature subset) from current classifier's archive as the input feature subset. We select the final solution by the descending order of priority on those three objectives here.

(a) Compare values of all solutions on objective f_1 and choose a solution with max value on objective f_1 as current classifier's solution.

(b) If there are many solutions with the same max value on objective f_1, we compare their values on objective f_2 and choose a solution with max value as current classifier's solution.

(c) If there are many solutions with the same max values on objective f_1 and objective f_2, we choose the solution with least cardinality of features as current classifier's solution.

4.3. Components of MOACO. Based on the discussion of Section 4.2, the key stage is to solve the maximization optimization problem (22) which is a classical multiobjective subset problem.

Cao et al. [33] proposed a graph-based ant system for solving subset problem. They defined construction graph and equivalent routes and proposed a new updating pheromone policy based on strengthening the pheromone on equivalent routes. The effectiveness and superiority of the policy were illustrated with multidimensional knapsack problem. But it was used to solve single optimization problem, and we generalize it to solve multiobjective subset problem (22).

Lopez-Ibanez and Stutzle [34] made a comparison between several state-of-art MOACO algorithms and concluded that it would improve solutions' quality by using more than one pheromone matrices. So we adopt two pheromone matrices as a component of MOACO and one matrix per objective.

In MOACO, each ant selects route (feature) according to the transition probabilities. In the case of traveling salesman problem, the probability that ant k chooses to visit node j after node i is given by

$$H_{ij}^{k} = \frac{\left[\tau_{ij}\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}}{\sum_{l \in N_i^k} \left[\tau_{ij}\right]^{\alpha} \cdot \left[\eta_{ij}\right]^{\beta}} \quad \text{subject to } j \in N_i^k, \tag{23}$$

where τ_{ij} is pheromone value of edge (i, j) and η_{ij} is heuristic information which is a static greedy measure of the "goodness" of edge (i, j). N_i^k denotes the set of feasible choices available for ant k located in node i given its current partial solution. α and β are algorithm's parameters which represent the importance degree of pheromone and heuristic information, respectively. When there is more than one pheromone matrix, the values of them need to be aggregated into a single pheromone value to calculate transition probabilities. In this paper, we adopt weighted product method to aggregate values of two pheromone matrices [34].

$$\tau_{ij} = \left(\tau_{ij}^{1}\right)^{(1-\lambda)} \left(\tau_{ij}^{2}\right)^{\lambda}, \tag{24}$$

where λ is a weight which biases the aggregation towards one objective or the other, and it changes with iteration increases as below [34].

$$\lambda_i = 1 - \frac{(i-1)}{\left(N_{\text{weight}} - 1\right)}, \quad i = 1, \ldots, N_{\text{weight}}, \tag{25}$$

where N_{weight} is algorithm's parameter.

Besides, based on our model's characteristic, the heuristic information η_h of MOACO is defined as the Fisher discriminant rate of hth feature of records' similarity vector.

$$\eta_h = \frac{\left|\overline{\mu}_{1h} - \overline{\mu}_{2h}\right|}{\sqrt{\sigma_{1h}^2 + \sigma_{2h}^2}}, \tag{26}$$

where $\overline{\mu}_{1h}$ and $\overline{\mu}_{2h}$ are average of the hth feature of similarity vector in matching and nonmatching classes, respectively. σ_{1h}^2 and σ_{2h}^2 are variance of the hth feature of similarity vector in matching and nonmatching classes, respectively. Equation (26) shows that MOACO selects underlying features with high Fisher discriminant rate, which means they are easily classified.

After one cycle, MOACO uses solutions found by all ants to update its Pareto archive based on their Pareto relations. Then we adopt equivalent routes' policy and solutions in Pareto archive to update the two pheromone matrices.

$$\tau_{ij}{}^{t} = \begin{cases} \dfrac{(1-\rho) \cdot \tau_{ij}{}^{t-1} + \Psi\left(\Pr_s^t\right)}{Q} & (i, j) \in \Pr_s^t \\ (1-\rho) \cdot \tau_{ij}{}^{t-1} & \text{otherwise}, \end{cases} \tag{27}$$

where ρ is evaporation rate of pheromone and Q is a constant parameter to fine-tune increment value of pheromone. $\Pr_s^t$ represents sth Pareto solution in archive after t iterations, and $\Psi(\Pr_k^t)$ denotes evaluation value of $\Pr_s^t$ in corresponding objective.

In summary, the proposed modified MOACO can be described as Algorithm 1.

We now analyze the time complexity of MOACO in Algorithm 1. For a given value q, the time complexity of initialization is $O(q^2 + q)$. The time complexity for constructing solutions by ants is $O(M \times q^2)$, where M is number of ants. The time complexity for updating Pareto archive is $O(M \times N_{\text{ma}})$, where N_{ma} is the number of solutions in archive. The time complexity for updating pheromone matrices is $O(q^2)$. The overall time complexity of MOACO is $O(NC \times q^2 \times M)$, where NC is the iteration number of MOACO.

4.4. Pseudocode and Complexity Analysis of the Proposed Method. Based on above discussion, the pseudocode of our method is shown in Algorithm 2.

Now we analyze the time complexity of Algorithm 2. The time required by initializing parameters is $O(1)$. For each classifier and a fixed q, the time complexity for implementing MOACO is $O(NC \times q^2 \times M)$. The time complexity for updating classifier's Pareto archive is $O(M \times N_{\text{ca}})$, where N_{ca}

```
Begin
    Initialize parameters, pheromone matrices, and Pareto archive
While not stopping criteria met do
    Generate weight parameter $\lambda$ by Eq. (25)
    Aggregate values from two pheromone matrices by $\lambda$ by Eq. (24)
For each ant do
    Construct solution by Eq. (23)
End for
    Update Pareto archive
    Update pheromone matrices by solutions of Pareto archive and Eq. (27)
End while
End
```

Algorithm 1: Pseudocode of MOACO.

```
Begin
    Initialize parameters and each base binary classifier's Pareto archive
For each base binary classifier do
For each $q$ value do
    Search for the optimal solution (feature subset) by Algorithm 1 under current $q$ value (cardinality of features)
End for
    Update current base binary classifier's archive based on analysis (ii) in Section 4.2
End for
    Choose a solution (feature subset) as the current base binary classifier's input based on analysis (iii) in Section 4.2
    Apply max-wins voting method to aggregate classifiers
End
```

Algorithm 2: Pseudocode of ensemble classifiers' model.

Table 1: Characteristics of the three testing datasets.

Dataset	Dimension	Sample	Class
Colon	2000	62	2
GLIOMA	4434	50	4
GLI_85	22283	85	2

Table 2: Characteristics of experimental datasets.

Dataset	Dimension	Matches	Nonmatches
Colon_ER	2000	176	880
GLIOMA_ER	4434	200	1000
GLI_85_ER	22283	200	1000

is the number of solutions in classifier's archive. The time complexity for choosing classifier's final solution is $O(N_{\text{ca}})$. So the overall time complexity of our method is $O(L \times NC \times q^3 \times M)$, where L is the number of base binary classifiers.

5. Experiment Settings and Discussions

5.1. Data and Preprocessing. We applied three datasets (Colon, GLIOMA, and GLI_85) which come from a well-known website (available from http://featureselection.asu.edu/datasets.php.) to evaluate our method. The characteristics of the three datasets are shown as Table 1.

In order to make the three datasets fit for ER, we must translate them into datasets composed of similarity feature vectors. For each dataset, we first normalized its features' range between zero and one. Then we chose two samples from the same class as a matching record pair and two samples from different classes as a nonmatching record pair. We used absolute difference value between two records' corresponding features as their features' similarities which constitute records' similarity feature vector. Besides, there are more nonmatching records than matching records in a real world problem, so we adopted uniform resampling method to make the number of matches less than that of nonmatches. We can obtain three new datasets for our experiment after above preprocessing, and they are called Colon_ER, GLIOMA_ER, and GLI_85_ER. The characteristics of them are shown as Table 2.

5.2. Experiment Settings. We made a comparison between two methods and our proposed method in this section. We adopted a single SVM for ER, it calculated records' similarity vectors by all features, and this model is called method 1. Our model is called method 2. Cao et al. [33] proposed a model for ER based on feature selection, and it took classification accuracy rate, recall rate, and cardinality of features as optimization objectives, and the experiment results showed a good performance for ER. We name the

Table 3: Results of Colon_ER.

Number	Compared methods	P'	R	F_1
1	Method 1	0.4267	0.8000	0.5565
	Method 2	**0.7038**	**0.8738**	**0.7796**
	Method 3	0.6753	0.7785	0.7233
2	Method 1	0.4808	**0.9000**	0.6268
	Method 2	**0.7707**	0.8676	**0.8163**
	Method 3	0.7070	0.8053	0.7529
3	Method 1	0.4800	0.9000	0.6261
	Method 2	**0.7429**	**0.9192**	**0.8105**
	Method 3	0.7021	0.8483	0.7683
4	Method 1	0.4257	0.8000	0.5557
	Method 2	**0.7544**	**0.8449**	**0.7971**
	Method 3	0.7374	0.8185	0.7758
5	Method 1	0.3738	0.7000	0.4873
	Method 2	**0.7718**	**0.8659**	**0.8162**
	Method 3	0.7662	0.8209	0.7926

model proposed by Cao as method 3 for comparison. We took fivefold cross validation in each test and adopted average classification precision P', recall rate R, and F_1 measure as evaluation measures.

$$\begin{aligned} P' &= \frac{\text{Number of samples correctly classified matches}}{\text{Number of samples classified matches}} \\ R &= \frac{\text{Number of samples correctly classified matches}}{\text{Number of matching samples}} \\ F_1 &= \frac{2 \cdot P' \cdot R}{P' + R}. \end{aligned} \tag{28}$$

Method 1 applied SVM as a classifier, whose kernel function was "rbf" and $\delta = 0.4$ and $C = 100$.

The parameters of method 2 were set as follows: the base binary classifier was SVM whose parameters were set as method1, number of base binary classifiers $L = 5$, cardinality of features $q \in [1, 20]$, pheromone matrices number of MOACO $n = 2$, initial value of pheromone matrices $\tau_{ij}{}^{0} = 100$, factors of importance of pheromone values and heuristic values $\alpha = 1$, $\beta = 2$, evaporation rate $\rho = 0.2$, constant value $Q = 0.02$, number of ants $M = 20$, parameter of weight λ $N_{\text{weight}} = 6$, number of solutions in classifier's archive $N_{\text{ca}} = 40$, number of solutions in MOACO archive $N_{\text{ma}} = 80$, and the stopping criteria of MOACO were set as iterations $NC = 40$.

The parameters of method 3 were set the same as those in [33].

5.3. Results and Discussions. The results of three methods on three datasets are shown in Tables 3, 4, and 5 (part of results).

The results on dataset Colon_ER in Table 3 show that method 1 has a lower precision and higher precision rate than other two methods, since high dimensional data has irrelevant features and noise information, and adopting all features to train classifiers may lead to overfitting; that is, most samples are classified incorrectly as matches. From Tables 4 and 5, we can find that method 1 does not identify any matching records on datasets GLIMOA_ER and GLI_85_ER. It is because that those two datasets have higher dimensions than Colon_ER, which leads to having more irrelevant features and noise information that reduce performance of classifier for ER. The experiment results of method 1 demonstrate that it may reduce classifier's performance by using all features of records and make the classifier unavailable for high dimensional ER.

The results of method 3 on three datasets demonstrate that it has a higher performance than method 1. Feature selection could filter irrelevant features and noise information, which helps to improve the performance of classifier. Method 3 takes classification precision, recall rate, and cardinality of features as optimization objectives for feature selection. So the selected features used to train classifier could make it classify most samples correctly, which improves its classification precision and recall rate. The values of F_1 measure also show that method 3 has a better performance for high dimensional ER than method 1.

At last, we can find that method 2 has a higher performance on three datasets than method 1 and method 3 for high dimensional ER. There are three reasons causing this situation. Method 2 takes classification accuracy rate as optimization objective to consider about both false alarm rate and fault is not recognized rate of classifiers to get an improvement for its performance. And Table 6 also shows that method 2 has a better classification accuracy rate than that of two others, which means it can classify most positive (matches) and negative (nonmatches) classes correctly at a high level. Method 3 only applies one classifier and a few features of high dimensional data, which leads to a loss of rich information. The results on GLIMOA_ER show that high

Table 4: Results of GLIOMA_ER.

Number	Compared methods	P'	R	F_1
1	Method 1	/	0	/
	Method 2	**0.8712**	**0.4965**	**0.6325**
	Method 3	0.7778	0.4650	0.5821
2	Method 1	/	0	/
	Method 2	**0.8048**	**0.7537**	**0.7784**
	Method 3	0.6880	0.6828	0.6854
3	Method 1	/	0	/
	Method 2	**0.8227**	**0.6630**	**0.7343**
	Method 3	0.6772	0.6300	0.6528
4	Method 1	/	0	/
	Method 2	**0.8300**	**0.7401**	**0.7824**
	Method 3	0.7149	0.6065	0.6563
5	Method 1	/	0	/
	Method 2	**0.7953**	**0.7509**	**0.7725**
	Method 3	0.6981	0.6877	0.6929

Table 5: Results of GLI_85_ER.

Number	Compared methods	P'	R	F_1
1	Method 1	/	0	/
	Method 2	**0.9900**	**0.7266**	**0.8381**
	Method 3	0.9216	0.5000	0.6483
2	Method 1	/	0	/
	Method 2	**0.9514**	**0.7829**	**0.8589**
	Method 3	0.8000	0.7543	0.7765
3	Method 1	/	/	/
	Method 2	**1**	**0.7596**	**0.8634**
	Method 3	0.9037	0.6667	0.7673
4	Method 1	/	0	/
	Method 2	**0.9747**	**0.8800**	**0.9249**
	Method 3	0.7753	0.7886	0.7819
5	Method 1	/	0	/
	Method 2	**0.9852**	**0.7308**	**0.8391**
	Method 3	0.9000	0.6923	0.7826

Table 6: Classification accuracy rate of three methods.

Dataset	Method 1	Method 2	Method 3
Colon_ER	0.5347 ± 0.0027	**0.7724 ± 0.0216**	0.7293 ± 0.0326
GLIOMA_ER	0.7486 ± 0.0039	**0.8693 ± 0.0181**	0.8358 ± 0.0122
GLI_85_ER	0.8330 ± 0.0039	**0.9630 ± 0.0125**	0.9318 ± 0.0033

dimensional data causes local feature relevance problem and the performance of method 3 deteriorates with dimensions increasing. In contrast, method 2 adopts ensemble classifiers which take the dissimilarity between classifiers as an optimization objective. It chooses complementary features which maximize the dissimilarity between classifiers to make full use of the rich information in high dimensional data to overcome local feature relevance problem. Finally, method 2 adopts ensemble learning to combine base binary classifiers which are trained by complementary feature subsets to further improve the performance of ensemble classifiers for high dimensional ER.

Table 7 shows the final solutions' values of five base binary classifiers of method 2 in objectives f_1 and f_2. We can find that the f_2 values of solutions (except the first classifier as there were no other classifiers when it is trained) are all greater than 0.5, and it demonstrates that the five classifiers are dissimilar from each other because of being trained by complementary feature subsets. So method 2 can make full use of rich information in high dimensional data to further improve its classification performance. We can also obtain a conclusion that the solutions found by the modified MOACO can make classifiers achieve a tradeoff between classification precision and recall rate.

Table 7: Objectives' values of final solutions of method 2 on GLIOMA_ER.

Number	Classifier	f_1	f_2
1	1	0.8267	/
	2	0.8385	0.5886
	3	0.8339	0.5839
	4	0.8312	0.5819
	5	0.8394	0.5880
2	1	0.8475	/
	2	0.8757	0.6130
	3	0.8612	0.6052
	4	0.8448	0.5915
	5	0.8584	0.6009
3	1	0.8512	/
	2	0.8648	0.6067
	3	0.8675	0.6078
	4	0.8575	0.6006
	5	0.8621	0.6034
4	1	0.8267	/
	2	0.8385	0.5886
	3	0.8339	0.5839
	4	0.8312	0.5819
	5	0.8394	0.5880
5	1	0.8240	/
	2	0.8246	0.5800
	3	0.8267	0.5787
	4	0.8249	0.5775
	5	0.8185	0.5730

6. Conclusions

In order to improve features' utilization rate of high dimensional data and reduce the impact brought by irrelevant features, an ensemble learning method based on feature selection is proposed, and some conclusions through experiments can be obtained as follows.

(1) It could reduce the impacts brought by irrelevant features and noise information through applying feature selection, which improves classification performance for high dimensional ER.

(2) It can get a tradeoff between classification precision and recall rate by taking classification accuracy rate as an optimization objective.

(3) The complementary feature subsets obtained by maximizing dissimilarity between classifiers can address local feature relevance problem efficiently.

(4) Combining classifiers trained by complementary feature subsets through ensemble learning method can further improve algorithm's performance for high dimensional ER.

(5) Note that the proposed method is very generic. It can be applicable for solving ER from any databases provided there is a way to calculate the distance between two given records without considering the number of dimensions (our method is also suitable for the situation with low dimensions in order to further improve the performance of ER). In order to illustrate the effectiveness of our method, we have used three datasets, but any other datasets could have also been used with suitable distance measure.

(6) The proposed method is also very generic with respect to the solution framework. Thus instead of MOACO any other optimization technique could be used. And also the SVM could be replaced by other classifiers.

Though our method has a lot of advantages, there are more problems that need to be solved. We adopt evolutionary algorithm to solve model, but the running time increases with dimensions increasing (e.g., we may need more time to select 20 features from data with 10000 features rather than 5000). Second, there are more nonmatches than matches in ER, which is known as imbalanced data problem, but we do not consider it in our method. Finally, we need more researches to make use of complementary feature subsets to further improve utilization rate of high dimensional data features.

Competing Interests

The authors declare that there is no conflict of interests regarding the publication of this manuscript.

Acknowledgments

This work is supported by the Natural Science Foundation of China under Grant no. 61371196, the China Postdoctoral Science Special Foundation under Grant no. 201003797, the China Postdoctoral Science Foundation under Grant no. 2015M582832 and Grant no. 20090461425, and the Jiangsu Postdoctor Research Fund of China under Grant no. 1402138C.

References

[1] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate record detection: a survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, pp. 1–16, 2007.

[2] P. Christen, "A survey of indexing techniques for scalable record linkage and deduplication," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 9, pp. 1537–1555, 2012.

[3] V. K. Borate and S. Giri, "XML duplicate detection with improved network pruning algorithm," in *Proceedings of the IEEE International Conference on Pervasive Computing (ICPC '15)*, pp. 1–5, IEEE, Pune, India, January 2015.

[4] E. K. Rezig, E. C. Dragut, M. Ouzzani, and A. K. Elmagarmid, "Query-time record linkage and fusion over Web databases," in *Proceedings of the 31st IEEE International Conference on Data Engineering (ICDE '15)*, pp. 42–53, Seoul, Korea, April 2015.

[5] O. Peled, M. Fire, L. Rokach, and Y. Elovici, "Entity matching in online social networks," in *Proceedings of the IEEE International Conference on Social Computing*, vol. 10, No. 1, Washington, DC, USA, 2013.

[6] Z.-H. Zhou, N. V. Chawla, Y. Jin, and G. J. Williams, "Big data opportunities and challenges: discussions from data analytics perspectives," *IEEE Computational Intelligence Magazine*, vol. 9, no. 4, pp. 62–74, 2014.

[7] J. Wang, T. Kraska, M. J. Franklin, and J. Feng, "CrowdER: crowdsourcing entity resolution ," in *Proceedings of the VLDB Endowment*, vol. 5, no. 11, 2012.

[8] S. E. Whang, P. Lofgren, and H. G. Molina, "Question selection for crowd entity resolution," in *Proceedings of the VLDB Endowment*, vol. 6, no. 6, pp. 349–360, Trento, Italy, August 2013.

[9] A. Abboura, S. Sahrl, M. Ouziri, and S. Benbernou, "CrowdMD: crowdsourcing-based approach for deduplication," in *Proceedings of the 3rd IEEE International Conference on Big Data (IEEE Big Data '15)*, pp. 2621–2627, Santa Clara, Calif, USA, November 2015.

[10] C. Zhang, R. Meng, L. Chen, and F. Zhu, "CrowdLink: an error-tolerant model for linking complex records," in *Proceedings of the the Second International Workshop on Exploratory Search in Databases and the Web*, pp. 15–20, ACM, Melbourne, VIC, Australia, May 2015.

[11] P. A. Priya, S. Prabhakar, and S. Vasavi, "Entity resolution for high velocity streams using semantic measures," in *Proceedings of the 5th IEEE International Advance Computing Conference (IACC '15)*, pp. 35–40, IEEE, Banglore, India, June 2015.

[12] S. Fries, B. Boden, G. Stepien, and T. Seidl, "PHiDJ: parallel similarity self-join for high-dimensional vector data with MapReduce," in *Proceedings of the 30th IEEE International Conference on Data Engineering (ICDE '14)*, pp. 796–807, Chicago, Ill, USA, April 2014.

[13] A. Abouzeid, K. B. Pawlikowsk, D. Abadi, A. Silberschatz, and A. Rasin, "HadoopDB: an architectural hybrid of MapReduce and DBMS technologies for analytical workloads," *Proceedings of the VLDB Endowment*, vol. 2, no. 1, pp. 922–933, 2009.

[14] J. Dittrich, J. A. Q. Ruiz, A. Jindal, Y. Kargin, V. Setty, and J. Schad, "Hadoop++: making a yellow elephant run like a cheetah (Without It Even Noticing)," *Proceedings of the VLDB Endowment*, vol. 3, no. 12, pp. 518–529, 2010.

[15] M. Alexandrov, V. Heimel, V. Markl et al., "Massively parallel data analysis with PACTs on nephele," in *Proceedings of the VLDB Endowment*, vol. 3, no. 1, pp. 1625–1628, 2010.

[16] F. Masulli and S. Rovetta, "Clustering high-dimensional data," in *Proceedings of the 1st International Workshop (CHDD'12)*, pp. 1–13, 2012.

[17] G. Cheng, D. Xu, and Y. Qu, "C3D+P: a summarization method for interactive entity resolution," *Web Semantics: Science, Services and Agents on the World Wide Web*, vol. 35, pp. 203–213, 2015.

[18] M. Song, E. H.-J. Kim, and H. J. Kim, "Exploring author name disambiguation on PubMed-scale," *Journal of Informetrics*, vol. 9, no. 4, pp. 924–941, 2015.

[19] J. Guerreiro, D. Gonçalves, and D. M. D. Matos, "Towards a fair comparison between name disambiguation approaches," in *Proceedings of the International Conference in the Riao Series: Open Research Areas in Information Retrieval (OAIR '13)*, pp. 17–20, Lisbon, Portugal, May 2013.

[20] P. Treeratpituk and C. L. Giles, "Disambiguating authors in academic publications using random forests," in *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries (JCDL '09)*, pp. 39–48, Austin, Tex, USA, June 2009.

[21] J.-J. Cao, X.-C. Diao, Y. Du, F.-X. Wang, and X.-Y. Zhang, "Classification detection of approximately duplicate records based on feature selection using ant colony algorithm," *Acta Armamentarii*, vol. 31, no. 9, pp. 1222–1227, 2010.

[22] C. Kacfah Emani, N. Cullot, and C. Nicolle, "Understandable big data: a survey," *Computer Science Review*, vol. 17, pp. 70–81, 2015.

[23] I. P. Fellegi and A. B. Sunter, "A theory for record linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, pp. 1183–1210, 1969.

[24] F. Naumann and M. Herschel, "An introduction to duplicate detection," in *Synthesis Lectures on Data Management*, vol. 2, No. 1, 2010.

[25] H. Yu and J. Ni, "An improved ensemble learning method for classifying high-dimensional and imbalanced biomedicine data," *IEEE/ACM Transactions on Computational Biology & Bioinformatics*, vol. 11, no. 4, pp. 657–666, 2014.

[26] K.-Q. Li, X.-C. Diao, J.-J. Cao, and F. Li, "High precision method for text feature selection based on improved ant colony optimization algorithm," *Journal of PLA University of Science & Technology*, vol. 11, no. 6, pp. 634–639, 2010.

[27] D. H. Wolpert, "The supervised learning no-free-lunch theorems," in *Proceedings of the World Conference on Soft Computing*, pp. 25–42, 2002.

[28] B. Krawczyk and G. Schaefer, "Breast thermogram analysis using classifier ensembles and image symmetry features," *IEEE Systems Journal*, vol. 8, no. 3, pp. 921–928, 2014.

[29] B. Li, J. Li, K. Tang, and X. Yao, "Many-objective evolutionary algorithms," *ACM Computing Surveys*, vol. 48, no. 1, pp. 1–35, 2015.

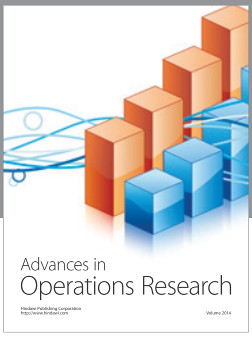
[30] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. Da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

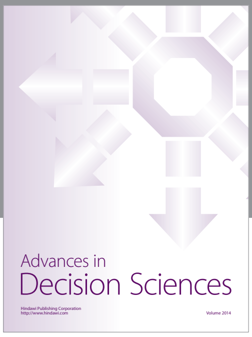
[31] L. Ke, Q. Zhang, and R. Battiti, "Using ACO in MOEA/D for Multiobjective Combinatorial Optimization," http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.720.644.

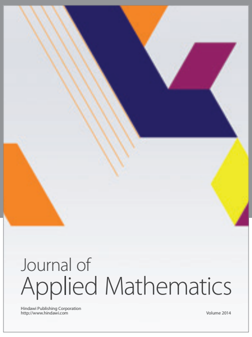
[32] M. López-Ibáñez and T. Stützle, "The impact of design choices of multiobjective ant colony optimization algorithms on performance: an experimental study on the biobjective TSP," in *Proceedings of the 12th Annual Genetic and Evolutionary Computation Conference (GECCO '10)*, pp. 713–720, Portland, Ore, USA, July 2010.

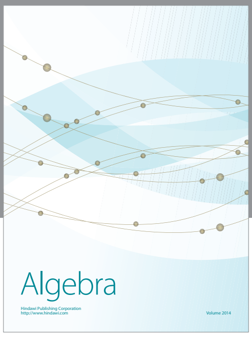
[33] J.-J. Cao, P.-L. Zhang, Y.-X. Wang, G.-Q. Ren, and J.-P. Fu, "Graph-based ant system for subset problems," *Journal of System Simulation*, vol. 20, no. 22, pp. 6146–6150, 2008.

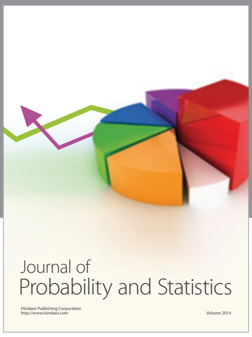
[34] M. Lopez-Ibanez and T. Stutzle, "The automatic design of multiobjective ant colony optimization algorithms," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 6, pp. 861–875, 2012.


Advances in
Operations Research

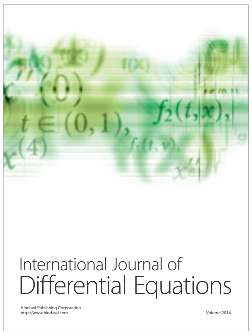

Advances in
Decision Sciences


Journal of
Applied Mathematics

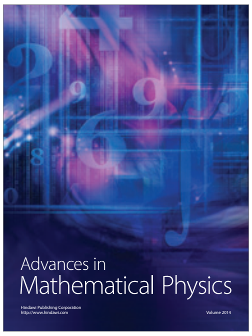

Algebra

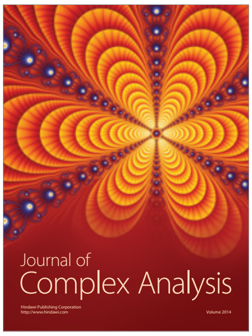

Journal of
Probability and Statistics

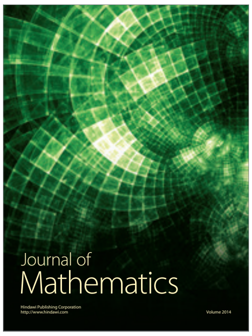
The Scientific
World Journal

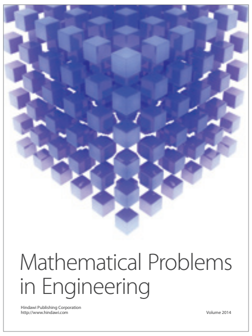

International Journal of
Differential Equations

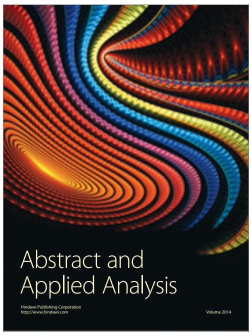
International Journal of
Combinatorics

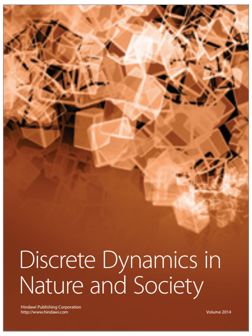

Advances in
Mathematical Physics

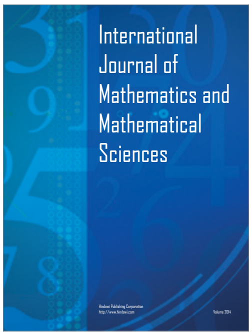

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

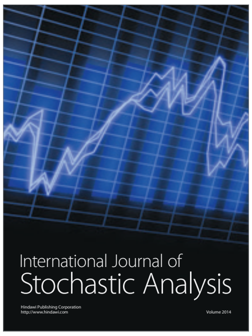

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization