International Journal of Engineering & Technology, 7 (3.27) (2018) 227-232

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Indexing Based Feature Selection by Applying Ant Colony Optimization Method for Improving Web Page Classification

A.M. James Raj[1*], F. Sagayaraj Francis[2]

[1]*Research Schloar, Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry, India.*
[2]*Professor, Department of Computer Science and Engineering, Pondicherry Engineering College, Pondicherry, India.*
E-mail:fsfrancis@pec.edu
**Corresponding author E-mail:jamdiva05@yahoo.co.in*

Abstract

In this information age many research work are carried out in web page classification to acquire the relevant and appropriate information. To be more specific, for enhancing the web page classification to obtain the optimized feature sets are chosen by utilizing the evolutionary algorithms. Normally, these algorithms are designed by the heuristic principles stimulated by natural evolution. After analyzing the significance of the various evolutionary algorithms deployed by several researchers in this domain so far, this work also intended to apply them to acquire the best solutions (enhanced features). In general, applying the evolutionary algorithms the fittest genes are generated and determined by the fitness function. Once the fittest genes are decided picking up the fittest individual genomes from a population for taking them to the next generations is the challenging task. In this article a novel approach is proposed to choose the best solutions.

Keywords: *Ant colony optimization, firefly algorithm, particle swarm optimization, cuckoo search algorithm, bat algorithms, wolf search and genetic algorithms or programming.*

1. Introduction

In general, the classification of web pages can be further improved by the way of effective feature selection approaches. Normally two classes of feature selection algorithms are found in the literature [1]. They are *traditional algorithms* and *evolutionary algorithms* [2]. This article focuses on effective feature selection for better classification by applying the most common evolutionary algorithms found in the literature [3] [4]. They are: *Ant Colony Optimization, Firefly Algorithm, Particle Swarm optimization* [5], *Cuckoo Search algorithm* [6], *Bat Algorithms, Wolf Search and Genetic algorithms or programming.* From the above, Ant Colony Optimization was chosen for exploration based on the reports in the literature.

Further, the evolutionary algorithms at a certain stage evaluate the choices of feature sets and filter them for further processing.

This filtering can be done by several selection methods. They are: *Roulette-wheel or Fitness proportionate selection, stochastic universal sampling, Tournament method, Truncation method, Elitism or elitist method and Threshold method.* Finally by comparing the empirical results of these methods this article also proposes a new selection method to acquire the optimum feature sets that enable the web page classification.

Basics of Evolutionary Algorithms

Recently, the evolutionary algorithms are utilized by many researchers in web page classification to acquire suitable solutions. Genetic algorithms are also principally part of this algorithm. Normally, these algorithms are designed by the heuristic principles stimulated by natural evolution. In general, they started the process with a set of solutions (chromosomes) called population. Solutions of one population are taken to the next level to form the population. It is commonly anticipated that the new population probably yielded better solutions than the prior one. New solutions (offspring) are formed from selected solutions based on their fitness values. That is the more suitable they are, more chances they have to reproduce. The cycle of evolution is repeated until a desired termination criterion is reached. This criterion can also be set by the number of evolution cycles or the total of variation of individuals between different generations or a predefined fitness value and so on[7].

Recombination is the process of mating the selected fittest in the reproduction stage to develop a new generation. Combination of fittest genes will lead to next generation. Various reproduction strategies and crossover methods are available. The simplest technique known as single-point crossover generates one or two child strings. Uniform crossover is used to obtain any combination of the two parent chromosomes.

Mutation denotes the arbitrary change of the value of a gene in the population. Random mutations alter a certain number of the bits in a mutant offspring. Single point mutation changes a bit one or zero and vice versa. Mutation creates influences to converge the algorithm. After analyzing the merits and demerits of the various evolutionary algorithms deployed by several researchers in this domain so far, this work also intended to apply them to find the best solutions (enhanced features) by utilizing the fitness functions are explained in the following sections.

General Procedure of Evolutionary Algorithms

The general steps involved in the genetic or evolutionary algorithms are summarized in Fig. 1.

1. *Start with a randomly generated population of n bit chromosomes (*candidate) solutions.
2. *Calculate the fitness f(x)* of each chromosome *x* in the population.
3. *Generate a fresh population* by repeatedly doing one or all the following till the population is finished.
   a. *Select a pair of parent chromosomes* from the current population. The choices of selection to be enhancing the fitness function. Selection is done *with replacement.* It refers that the same chromosome can be selected repeatedly to be chosen a parent.
   b. Apply the *crossover* probability p_c, to cross over the parents to form a new offspring (children). If no crossover was performed offspring is an exact copy of parent.
   c. Apply the *mutation* probability p_m, to mutate new offspring at each location in chromosome.
   d. *Place new offspring* in a new population.
4. *Apply fresh generated population* for further execution of algorithm.
5. *If (Terminating condition attained)*
   Stop the process *and submit* the enhanced solution
6. *Else*
   go to step 2.

Fig. 1: Algorithms for an optimal feature selection using the ant colony optimization

2. Related Work

Principles of Ant Colony Optimization

This method was proposed by Marco Dorigo and inspired by the social behavior of ants. In which, each ant moves randomly and the pheromone is deposited on the path. The idea behind this method is identifying the shortest path by the pheromone deposited on the path of ant's travels. Here the fundamental concept is that the higher pheromone on path augments chance of path being tracked easily. Finding a most favorable path between the ant's colony and source of food is the task of this method and it could be simulated by graph methods [7] [8].

Ant Based Solution Construction: Artificial ants construct solutions from a finite set of *n* available solution components. A solution construction starts with an empty partial solution $s^p = \emptyset$. Then, at each construction $C = \{c_{ij}\}$ step, the current partial solution s^p is extended by adding a feasible solution component from the set $N(s^p) \in C \setminus s^p$, which is defined by the solution construction mechanism. The constructing solutions process can be regarded as a path on the construction graph $G_C = (v, e)$. The set of solution components c may be associated either with the set v of vertices of the graph G_C, or with the set e of its edges. The allowed paths in G_C are implicitly defined by the solution construction mechanism that defines the set $N(s^p)$ with respect to a partial solution s^p .

The choice of a solution component from $N(s^p)$ is done probabilistically at each construction step. The exact rules for probabilistic choice of solution components vary across different variants of ACO. This is computed by the formula given in 1.

$$p\left(c_{ij}\middle|s^p\right) = \frac{\tau_{ij}^{\alpha}\cdot\eta(c_{ij})^{\beta}}{\sum_{c_{il}\in N(s^p)} \tau_{ij}^{\alpha}\cdot\eta(c_{il})^{\beta}}, \qquad \forall c_{ij} \in N(s^p) \tag{1}$$

where τ_{ij} is the pheromone value associated with component c_{ij}, $\eta(\cdot)$ is a weighting function that assigns at each construction step a heuristic value to each feasible solution component $c_{ij} \in N(s^p)$.

The values that are given by the weighting function are commonly called the heuristic information. Further α and β are positive parameters, whose values determine the relation between pheromone information and heuristic information.

Pheromone Update: The aim of pheromone update is to increase the pheromone values associated with good or promising solutions, and decrease those that are associated with bad ones. Usually, this is achieved by increasing the pheromone levels associated with chosen good solution S by a certain value Δ_τ, and by decreasing all the pheromone values through pheromone evaporation τ_{ij}. It is computed by the formula given in 2.

$$\tau_{ij} \leftarrow \begin{cases} (1-p)\tau_{ij} + p\Delta\tau, & if\ \tau_{ij} \in s_{ch} \\ (1-p)\tau_{ij}\,, & Otherwise \end{cases} \tag{2}$$

Where, $\rho \in [0, 1]$ denote the range of the evaporation.

Evaporation of the Pheromone is required to evade quick convergence of the problem. Usually, pheromone update algorithms are performed either the *iteration-best solution or the best-so-far solution.* The former method allows for more diversification of the search and the later one leads to quick convergence [9].

An ant colony optimization algorithm for finding the best solution using the fitness function is given in Fig. 2.

```
/*Algorithm for Ant Colony Optimization */
Steps
Initialization
while (stop condition not reached)
{
    position each ant in a first node
    do
    {
        for each ant do
        {
            choose next node by using rules of the state transition
                apply pheromone update step
        }
    } while (each ant make a solution) };
    update best solution
    apply offline pheromone update
}
```

Fig. 2: Algorithms for an optimal feature selection using the Ant colony optimization

Selection Methods in Evolutionary Algorithms

In general, applying the evolutionary algorithms the fittest genes are generated and determined by the fitness function. Once the fittest genes are decided picking up the fittest individual genomes from a population for taking them to the next generations is the challenging task. To pick the fittest solutions numerous methods exist in the literature. Few of them are: *Threshold method, Roulette wheel, Tournament selection, stochastic universal sampling, Elitist selection, Reward based selection, Ranking* and so on [10]. A brief note is given to understand about these selection methods as follows.

Threshold selection (THS): This method only considers solutions, of fitness value that is greater than a constant value which is fixed arbitrary. It means that it will eliminate a fixed percentage of the weakest candidates. It is also called as *truncation method.* With fitness proportionate selection there is a chance some weaker solutions may survive the selection process; this is an advantage, as though a solution may be weak, it may include some component which could prove useful following the recombination process.

Roulette-wheel selection (RWS): This selection method is also called as *or fitness proportionate method.* It is similar to a roulette wheel in a casino. In which, a sector of the wheel is fixed to each of the feasible options by their fitness value. It could be attained by dividing the fitness of a selection by the total fitness of all the selections. Then a random selection is made similar to how the roulette wheel is rotated. Here the candidate solutions with a higher fitness will be less likely to be eliminated.

Tournament selection (TS): This selection strategy involves in allowing several tournaments among a few individuals or chromosomes chosen at random from the population. The winner of each tournament is selected for crossover. The winner is a one among the chromosomes with the best fitness value. This method can be improved by simply changing the size of tournament. Here, the thin elements have a little opportunity to be picked up for the next generation when the tournament size is big. Once the selection is made the selected elements should be eliminated from the search domain. Otherwise, that selected individuals could be selected again and again.

Stochastic universal sampling (SUS): It is another popular selection method applied for deciding the useful solutions for recombination. SUS is a further enhancement of fitness proportionate selection (FPS). While the FPS selects various solutions among the population by using the repetitive sampling by arbitration the SUS applies a singular arbitrary constant value for all the solutions by selecting them at equally spaced intervals. It provides weaker members of the population an opportunity to be picked up and cut short the unfair method of FPS methods. FPS method performs badly when a member of the population has a really large fitness in comparison with other members.

3. Proposed Method

Principles of Indexing

In general, the prime aspect of organizing the data base or data repository is to facilitate the quick and proper retrieval of the vital information. Typically, an Indexing technique is among the simplest and efficient way of organizing such huge volume of data items to respond to queries at a faster rate. This can be achieved by reducing number of peripheral storage input-output operations. Indexes also facilitate reduction. An index may be defined as any extraneous structure apart from the actual data store. It facilitates storage retrieval search and analysis of data across classifications and categories without retrieving the entire data store. Thus, indexing is an efficient method that consists of both structure and process. Further, it enables quick access of data across classifications and categories without retrieving the entire data store. An index that is designed on a single attribute is called a *single dimensional index*. An index that is designed on multiple attributes is called a *multidimensional index*.

MBG-Tree Indexing Selection Method

To select the most advantageous or enhanced features from the given large dataset for increasing the performance of the web page classification is a demanding issue. Although, there were numerous techniques suggested by several researchers, this research domain needs greatly sophisticated methods to suit the present day complex applications. To facilitate the feature selection to be simple as well as efficient, one can adopt the evolutionary algorithms and that principles were deployed with several experiments earlier in this chapter in a primitive level. It provided satisfactory results with various kinds of datasets of the standard popular universities.

To supplement and to extend this work, a novel approach is proposed. It not only improves the feature selection, but also speeds up the performance of a web page classification. It is based on the principles of indexing technique and obtained in a modified way of *MB Trees for Genetic algorithms* to choose the chromosomes followed by the fitness functions estimation of the evolutionary algorithms. So, it is named as *MBG-trees*. The nodes of the MBG-Tree consist of elements along with the fitness value of elements combined together. Insertion is made at the leaf level. To insert an element apply *Search* to find out the leaf node where the element can be inserted.

The algorithm for Search, to find out the leaf node to insert it in the appropriate location is given in Fig. 3.

```
Algorithm Search (N, e)
/* Algorithm to insert an element e in Node N */
{
    if N is leaf and has space for element e
    {
        insert e into N
        update fitness value of N
    }
    while N has a parent M
    {
        update the fitness value of M by reckoning e
        assign M to N
    }
    If N is leaf and has no space for element
    {
        let M be the parent of N.
        let N1 be one of the nodes after splitting N by reckoning e
        let N2 be one of the nodes after splitting N by reckoning e
            call Insert(M, N1, N2)
    }
    if N is not a leaf
    {
        compute fitness value for every element x of N by reckoning e
        select N as the child node corresponding to the best fitness value
            call Search(N, e)
    }
}
```

Fig. 3: Algorithm to insert an element *e* in Node *N*

The algorithm for ***inserting an element in the location*** determined by the Search is given in Fig. 4.

```
Algorithm Insert(M, X, Y)
/* Algorithm to insert an element apply Search to find out the leaf node
   where the element can be inserted
   M be the parent of N to be split upon insertion of e.
   X be one of the nodes after splitting N by reckoning e
   Y be one of the nodes after splitting N by reckoning e
*/
{
    let x be the fitness value of X
    let y be the fitness value of Y
    if M has space for two elements
    {
    update fitness value of M by reckoning e
    create two child nodes X, Y for M
    }
    else if redistribution of elements possible without splitting the node M
    {
    redistribute the elements among M and its children
    update the fitness values of node M and its children
    propagate the updating to the ancestors of M.
    }
    else
    {
    Split M after reckoning X and Y
    redistribute the elements among children of M
    update the fitness values of node M and its children
    propagate the split to the parents of M
    }
}
```

Fig. 4: Algorithm to insert an element call Search procedure to find out the leaf node where the element can be inserted

4. Experimental Setup and Data Sets Description

A number of experiments were conducted on various real data sets downloaded from the UCI repository of the Internet in order to prove that this proposed work enhances the classification accuracy.

a) First of all, the entire experiments throughout this research work were coded and implemented using Java language, developed in Net Beans IDE 8.0 and JDK 1.7. This program was executed in an i7 processor with 8 GB RAM, by using Web KB data sets.
b) Throughout the experiments, the classifier was modelled by the percentage split method with the 60% of the input data as learning (training) and rest of 40% for testing (classification) process.
c) The primary intention of conducting the experiments was to know the efficiency of the classifiers by providing the ordinary way of selecting the feature sets.

The data sets are the main input components for doing the experiments in the classification model to examine and testify their performance. To do this several university computer science course based Web KB datasets down loaded from the UCI repository. They are the benchmarking datasets used in many research works and given as follows: They are: ***Texas University, Wisconsin University and Cornell University.***

Description of Texas University Data Set

The various components of Texas University data set is described as in Table 1.

Table 1: Texas University Data Set

S.No.	Broad Category	Number of Pages
1	Department	45
2	Course	300
3	Faculty	1750
5	Projects	1230
6	Announcements	180

In the above described data set, ***the Department category*** states about the particular department such as Computer Science, Engineering and Electronics etc. The ***Course category*** tells about the Course content, Grading, Problem sets, Required Text Book and so on. The ***Faculty category*** refers about the Professor or Instructors name, Teaching assistant name, Class Schedule (office hours), Address, Email and Phone Number and etc. The ***Projects category*** narrates about the project, project selection and submission details, evaluation methods and so on. Finally the ***Announcements category*** lists the various notifications to the students such as Room assignments(allotments), Course and Instructor Names, Lab and discussion selection schedule, Programming Assignment, Important News articles, Class News groups, Mid-term Test I and II and so on.

Description of Wisconsin University Data Set

The various components of Wisconsin University data set is described as in Table 2.

Table 2: Wisconsin University Data set

S. No.	Broad Category	Number of Pages
1	Courses	1111515
2	Staff Members	250
3	Seminars	30
4	Laboratories	42
5	Important Notifications	30

In the above described data set, The *Courses category* states about the Name of the course, Pre-requisites, Course content, Number of credits, Grading, Required Text Book and so on. The *Staff members category* refers about the Professor or Instructors name, Teaching assistant names, Class Schedule (office hours), Grading point, Lectures and discussion sections, Address, Email and Phone Number and etc. The *Seminars category* narrates about the Seminar topics, Resource Person details, Seminar duration, date and time, venue, schedule and so on. The *Laboratory category* states about the Name of the Laboratory, Program lists, Laboratory in-charge, Number of credits, Number of practical hours, Grading and so on. Finally the *Important Notifications category* lists the various notifications to the students such as Lectures, Course and Instructor Names, Computer Labs, Programming Assignment, Important News articles, Class News groups, and other pointers of Interest.

Description of Cornell University Data Set

The various components of Cornell University data set is described as follows in Table 3.

Table 3: Cornell University Data set

S. No.	Broad Category	Number of Pages
1	Courses s	145
2	Syllabus	150
3	Homework	445
4	Practicum	300
5	Exam	210

In the above described data set, The ***Courses category*** states about the Name of the course, Course content, Number of credits, Grading, Instructors name, Required Text Book, Course type and so on. The ***Syllabus category*** refers about the Name of the subject, Instructors name, teaching assistant names, Office hours, Grading point, Lectures and discussion sections and etc. The ***Homework category*** narrates about the Homework policies, Homework submission, Graded homework, Re-grade policy and so on. The ***Practicum category*** states about the Name of the practical, Handouts, practical lists, System facilities and configuration etc. The ***Exam category*** states about the Name of the Examination, Examination date, time and location and Examination supervisor etc.

Performance Evaluation Metrics

The standard performance evaluation metrics such as precision, recall, f-measures and accuracy are predominantly used in data mining applications to predict the efficiency of the work. In this classification model too, these measures are employed to analyze the performance efficiency. These performance evaluation metrics are determined by following formulas which are given in the equations from 3 to 6 respectively.

Precision

It is the ratio between the True Positive and the total sum of True and False positive values. It can be computed by the formula given in Eqn.3 using the confusion matrix given in Table 4.

$$Precision\ (p) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}} \quad (3)$$

Recall

It is the ratio between the True Positive and the total sum of True and False Negative values. It can be computed by the following formula given in Eqn. 4.

$$Recall\ (r) = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \quad (4)$$

F-measure

It can be employed in Information Retrieval to test the performance of the classifier. It is the harmonic mean of precision and recall. It can be calculated by the formula given in Eqn.5.

$$f - measures\ (f) = \frac{\mathrm{2xPxR}}{\mathrm{P} + \mathrm{R}} \quad (5)$$

Accuracy

It is the ratio of the total number of TP and TN of the total number of data. It can be calculated by the formula given in Eqn.6.

$$Accuracy\ (a) = \frac{TP + TN}{TP + TN + FP + FN} \qquad (6)$$

A confusion matrix listed in Table 4 can be used to determine the performance evaluation metrics described earlier.

Table 4: Confusion Matrix

Confusion Matrix	Predicted Positive	Predicted Negative
Actual Positive	TP-True Positive	FN –False Negative
Actual Positive	FP-False Positive	TN –True Negative

5. Experimental Results and Discussion

The experimental results of the various evolutionary algorithms utilized for enhancing the feature selection process together with the several university based data sets are summarized in the following section with a detailed discussion. On the whole, this *MBG-Tree Indexing principle* is utilized in all the experiments. The main idea of using this method is the improving the outcomes of the previous work based on selection method by the evolutionary algorithms. So, the same data sets were considered for conducting the experiments. The contribution of MBG-Tree indexing selection method is used for selecting the required number of features among the fittest solutions after applying the fitness functions. It is worth mentioning that, this new idea plays a vital role and leverages the tedious task of getting optimum features. It is evident that this MBG-Tree Indexing principle yielded enhanced results by comparing with the prior work results. Initially, the aforesaid four university based experiments were conducted together with the *MBG-Tree Indexing selection*. The obtained results were compared with the other conventional selection methods such as *Threshold Selection (THS), Roulette Wheel Selection (RWS), Tournament selection (TS) and Stochastic Universal Selection (SUS).* Eventually, the results are presented with the graphs to understand the performance of the classification process using the standard performance evaluation metrics with respect to Precision, Recall, F-measures and Accuracy.

Comparison of MBG-T Selection with other Methods Using Ant Colony Optimization Algorithm

It is envisioned that *if the MBG-T Selection Method is applied with the evolutionary ant colony optimization algorithms for selecting features, this proposed method yields superior classification performance than the other conventional selection methods.* This is proved by the experiments conducted earlier and their results which are presented by the following graphical representations. The results of Texas, Wisconsin and Cornell University are given in the following Figures6 and 7 respectively.[11][12]

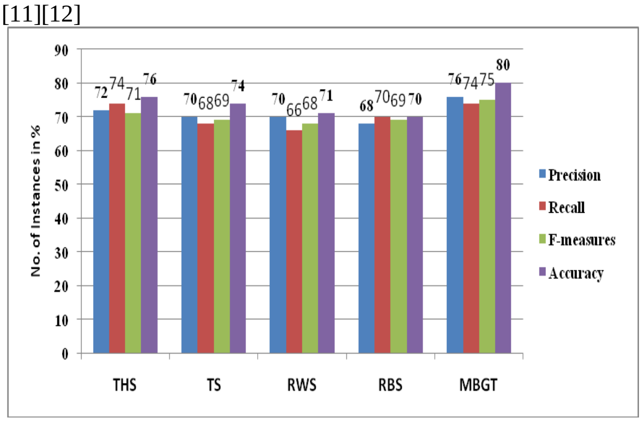

Fig. 5: Comparison chart of various traditional fitness selection methods with the MBG-T method in ACO using NBC for texas university data set

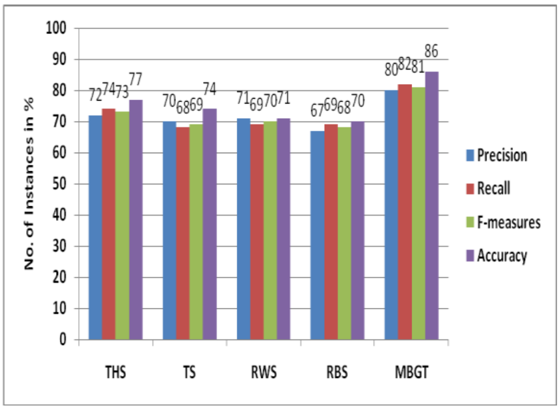

Fig. 6: Comparison chart of various traditional fitness selection methods with the MBG-T method in ACO using NBC for wisconsin university data set

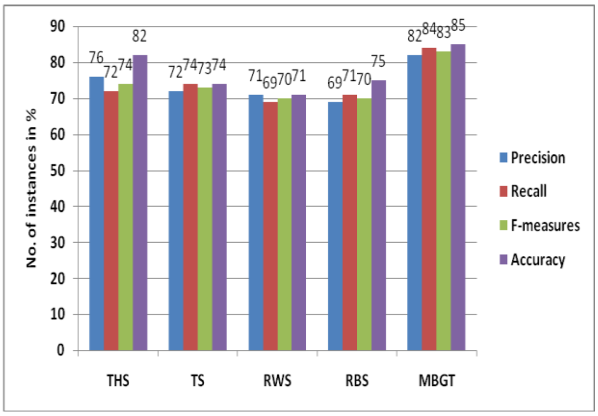

Fig. 7: Comparison chart of various traditional fitness selection methods with the MBG-T method in ACO using NBC for cornell university data set

6. Conclusion

Feature selection is an elementary but an essential task in data mining principle that extracts only the prominent features from the data sets. It's primary focus is not only reduces the features but also facilitates the efficient performance of the classifier. In this article a novice principle was explored *by applying evolutionary algorithms* to choose the optimized *feature set selection*. Initially, these algorithms utilized the fitness function to select the best features from the entire population by applying the various conventional selection methods. Later, these feature sets are supplied to the conventional classifiers for the classification. The experiments were conducted and the results were depicted with the bar charts. The proposed method was involved with the indexing techniques to acquire still more optimized features. In which, an innovative selection method was designed and implemented called as MBG-*Indexing technique incorporated with the Evolutionary Algorithms for selecting the good feature subsets*. The results were evaluated with the standard data mining performance metrics and compared with the results of the prior work. It was shown that an excellent classification results were attained by choosing only the optimized features. Moreover, this research article provides an abundant scope for further enhancement in numerous recent domains.

a) Since the World Wide Web is an elegant global distributed environment and certainly it will undergo to a lot of unpredicted progresses in the years marching in the next few decades. Obviously, that brings unimaginable challenges in this sphere and that disclose the way for further research works.

b) Further, by the impact of internet and its various kinds of facilities accumulated each and every day an enormous volume of inconsistent, irrelevant set features are generated.

c) It definitely provides potential place for researchers in web mining by proposing innovative and effective *feature selection mechanisms that eventually used to classify the web contents efficiently.*

References

[1] Chitra P & Venkatesh P, "Multiobjective evolutionary computation algorithms for solving task scheduling problem on heterogeneous systems", *International journal of knowledge-based and intelligent engineering systems*, Vol.14, No.1,(2010), pp.21-30.

[2] Yang XS & He X, "Firefly algorithm: recent advances and applications", *International Journal of Swarm Intelligence*, Vol.1, No.1,(2013), pp.36-50.

[3] Zhang Q & Richard S, "Web Mining: A Survey of Current Research, Techniques and Software", *International Journal of Information Technology & Decision Making*, (2008), pp.683–720.

[4] Wu S & Li Y, "Pattern-Based Web Mining Using Data Mining Techniques", *International Journal of e-Education, e-Business, e-Management and e-Learning*, (2013), pp.163-167.

[5] Li Y, Chen XZ & Yang BR, "Research on web mining-based intelligent search engine", *International Conference on Machine Learning and Cybernetics*, (2002), pp.386-390.

[6] Hendtlass T, "Particle Swarm Optimization and high dimensional problem spaces", *IEEE Congress on Evolutionary Computation*, (2009), pp.1988-1994.

[7] Yang XS & Deb S, "Cuckoo Search via Levy flights", *World Congress on Nature and Biologically Inspired Computing*, (2015), pp.61-68.

[8] San PE, "Classification of Web Pages using TF-IDF and Ant Colony Optimization", *International Journal of Scientific Engineering & Technology Research*, (2014), pp.61-68.

[9] Kim C & Shim K, "Text: Automatic template extraction from heterogeneous web pages", *IEEE Transactions on knowledge and data Engineering*, Vol.23, No.4,(2011), pp.612-626.

[10] Xue B, Zhang M & Brown WN, "Particle Swarm Optimization for Feature Selection in Classification: A multi-objective approach", IEEE Transactions on Cybernets, (2013), pp.1656-1671.

[11] A Mukanbetkaliyev, S Amandykova, Y Zhambayev, Z Duskaziyeva, A Alimbetova (2018). The aspects of legal regulation on staffing of procuratorial authorities of the Russian Federation and the Republic of Kazakhstan Opción, Año 33. 187-216.

[12] G Cely Galindo (2017) Del Prometeo griego al de la era-biós de la tecnociencia. Reflexiones bioéticas Opción, Año 33, No. 82 (2017):114-133