ISSN: 2229 – 6956(ONLINE)
DOI: 10.21917/ijsc.2011.0020

DIVERSE DEPICTION OF PARTICLE SWARM OPTIMIZATION FOR DOCUMENT CLUSTERING

K. Premalatha[*] and A.M. Natarajan
Bannari Amman Institute of Technology, Tamil Nadu, India
E-mail: [*]kpl_barath@yahoo.co.in

Abstract
Document clustering algorithms play an important task towards the goal of organizing huge amounts of documents into a small number of significant clusters. Traditional clustering algorithms will search only a small sub-set of possible clustering and as a result, there is no guarantee that the solution found will be optimal. This paper presents different representation of particle in Particle Swarm Optimization (PSO) for document clustering. Experiments results are examined with document corpus. It demonstrates that the Discrete PSO algorithm statistically outperforms the Binary PSO and Simple PSO for document Clustering.

Keywords:
Particle Swarm Optimization, Document Clustering, Inertia Weight, Constriction Factor, Swarm Intelligence

1. INTRODUCTION

Document clustering is an automatic grouping of text documents into clusters so that documents within a cluster have high similarity in comparison to one another, but are dissimilar to documents in other clusters. Unlike document classification no labeled documents are provided in clustering; hence, clustering is also known as unsupervised learning. Document clustering is widely applicable in areas such as search engines, web mining, information retrieval and topological analysis. Document clustering has become an increasingly important task in analyzing huge numbers of documents distributed among various sites. The challenging aspect is to analyze this enormous number of extremely high dimensional distributed documents and to organize them in such a way that results in better search and knowledge extraction without introducing much extra cost and complexity. Clustering, in data mining, is useful to discover distribution patterns in the underlying data. A common document clustering method [1][12] is the one that first calculates the similarities between all pairs of the documents and then cluster documents together if their similarity values are above some threshold.

In this paper, a document clustering algorithm based on different representation of PSO is proposed. The remainder of this paper is organized as follows: Section 2 gives a general overview of the PSO. The different representation of PSO clustering algorithm is described in Section 3. Section 4 presents the detailed experimental setup and results for comparing the performance of the variant representation of PSO algorithms.

This template is designed to help you in preparing your manuscript in expected format. The guidelines include complete descriptions of the fonts, spacing and related information for creating your proceedings manuscripts. Please follow them properly.

2. PARTICLE SWARM OPTIMIZATION

PSO [3][7] is a population based stochastic optimization technique for the solution of continuous optimization problems. It is inspired by social behaviors in flocks of birds and schools of fish. In PSO, a set of software agents called particles search for good solutions to a given continuous optimization problem. Each particle is a solution of the considered problem and uses its own experience and the experience of neighbor particles to choose how to move in the search space. In practice, in the initialization phase each particle is given a random initial position and an initial velocity. The position of the particle represents a solution of the problem and has therefore a value, given by the objective function. While moving in the search space, particles memorize the position of the best solution they found. At each iteration of the algorithm, each particle moves with a velocity that is a weighted sum of three components: the old velocity, a velocity component that drives the particle towards the location in the search space where it previously found the best solution so far, and a velocity component that drives the particle towards the location in the search space where the neighbor particles found the best solution so far. PSO has been applied to many different problems in this work it is applied in Document Clustering.

2.1 BINARY PSO

In mathematical terms, Kennedy and Eberhart are proposing a model wherein the probability of an individual's deciding yes or no, true or false, or making some other binary decision, is a function of personal and social factors [7]:

$$P(x_{id}(t)=1)=f(x_{id}(t-1),v_{id}(t-1),p_{id},p_{gd}) \qquad (1)$$

where

$P(x_{id}(t)=1)$ is the probability that individual i will choose 1 for the bit at the dth site on the bitstring

$x_{id}(t)$ is the current state of the bitstring site d of individual i

t means the current time step, and $t-1$ is the previous step

$v_{id(}t-1)$ is a measure of the individual's predisposition or current probability of deciding 1

p_{id} is the best state found so far, for example, it is 1 if the individual's best success occurred when x_{id} was 1 & 0 if it was 0

p_{gd} is the neighborhood best, again 1 if the best success attained by any member of the neighborhood was when it was in the 1 state and 0 otherwise

If $v_{id}(t)$ is higher, the individual is more likely to choose 1, and lower values favor the 0 choice. Such a threshold needs to stay in the range [0.0, 1.0]. A straightforward function for accomplishing this is the sigmoid function.

A lot of randomness allows exploration of new possibilities, and a little bit allows exploitation by testing patterns similar to the best one found so far; thus the balance between those two modes of search by adjusting the uncertainty of decisions. The parameter $v_{id}(t)$, an individual's predisposition to make one or the other choice, will determine a probability threshold. If $v_{id}(t)$ is higher, the individual is more likely to choose 1, and lower values favor the 0 choice. Such a threshold needs to stay in the range [0.0, 1.0]. The sigmoid function

$$s(v_{id}) = \frac{1}{1+\exp(-v_{id})} \qquad (2)$$

squashes its input into the requisite range and has properties that make it agreeable to being used as a probability threshold.

To adjust the individual's disposition toward the successes of the individual and the community a formula for each v_{id} in the current time step that will be some function of the difference between the individual's current state or position and the best points found so far by itself and by its neighbors. To favor the best position, but not so much that the individual ceases searching prematurely, simply added ($p_{id} - x_{id}(t-1)$) and ($p_{gd} - x_{id}(t-1)$) to v_{id} (t), it would move upward when the difference between the individual's previous best and most recent states, or the difference between the neighborhood's best and the individual's most recent states, equaled 1, and would be attracted downward if either difference equaled −1. The probability threshold moves upward when the bests are ones and downward when they are zeroes. To weight them both with random numbers, then sometimes the effect of one, and sometimes the other, will be stronger. The symbol ϕ to represent a positive random number drawn from a uniform distribution with a predefined upper limit. In the binary version the limit is somewhat arbitrary, and it is often set so that the two ϕ limits sum to 4.0.

$$v_{id}(t)=v_{id}(t-1)+\varphi_1(p_{id}-x_{id}(t-1))+\varphi_2(p_{gd}-x_{id}(t-1)) \qquad (3)$$

$$\rho_{id} < s(v_{id}(t)) \; then \; x_{id}(t)=1; else\, x_{id}(t)=0 \qquad (4)$$

where ρ_{id} is a vector of random numbers, drawn from a uniform distribution between 0.0 and 1.0. These formulas are iterated repeatedly over each dimension of each individual, testing every time to see if the current value of x_{id} results in a better evaluation than p_{id}, which will be updated.

Initially all the particles of the swarm should be at the same place. But, after the first time increment, they will be dispersed randomly, because, in the absence of any information, this is still the best method. Therefore, to simplify that, this random distribution is the initial position of the swarm. This is also relates to the rates of travel of the particles will also initialize randomly, over a reasonable range of values, as a function of the size of the search space.

2.2 SIMPLE PSO

As the system is dynamic, each individual is presumed to be moving this should really be called changing at all times. The direction of movement is a function of the current position and velocity, the location of the individual's previous best success, and the best position found by any member of the neighborhood:

$$\vec{x}_i(t) = f(\vec{x}_i(t-1), \vec{v}_i(t-1), \vec{p}_i, \vec{p}_g) \qquad (5)$$

Further, just as in the binary version, change is a function of the difference between the individual's previous best and current positions and the difference between the neighborhood's best and the individual's current position. In fact the formula for changing the velocity is identical to the one used to adjust probabilities in the binary version, except that now variables are continuous, and what is adjusted is the particle's velocity and position in R^n:

$$\vec{v}_{id}(t)=\vec{v}_{id}(t-1)+\varphi_1(\vec{p}_{id}-\vec{x}_{id}(t-1))+\varphi_2(\vec{p}_{gd}-\vec{x}_{id}(t-1)) \qquad (6)$$

$$\vec{x}_{id}(t)=\vec{x}_{id}(t-1)+\vec{v}_{id}(t) \qquad (7)$$

The acceleration constants φ_1 and φ_2 in equation (6) represent the weighting of the stochastic acceleration terms that pull each particle toward *pbest* and *gbest* positions. Low values allow particles to roam far from target regions before being tugged back, while high values result in abrupt movement toward, or past, target regions.

The system as given thus far has a tendency to explode as oscillations become wider and wider, unless some method is applied for damping the velocity. The usual method for preventing explosion is simply to define a parameter $V_{\max}$ and prevent the velocity from exceeding it on each dimension *d* for

$$v_{id} = \begin{cases} V_{\max} & v_{id} > V_{\max} \\ -V_{\max} & v_{id} < -V_{\max} \end{cases} \qquad (8)$$

The effect of this is to allow particles to oscillate within bounds, although with no tendency for convergence or collapse of the swarm toward a point. Even without converging, the swarm's oscillations do find improved points in the optimal region. $V_{\max}$ is therefore an important parameter. It determines the resolution, or fineness, with which regions between the present position and the target position are searched. If $V_{\max}$ is too high, particles might fly past good solutions. If $V_{\max}$ is too small, on the other hand, particles may not explore sufficiently beyond locally good regions. In fact, they could become trapped in local optima, unable to move far enough to reach a better position in the problem space.

One approach to controlling the search is the implementation of an inertia weight. The maximum velocity V_{max} serves as a constraint to control the global exploration ability of a particle swarm. As stated earlier, a larger V_{max} facilitates global exploration, while a smaller V_{max} encourages local exploitation. The concept of an intertia weight was developed to better control exploration and exploitation. The motivation was to be able to eliminate the need for V_{max}. The inclusion of an inertia weight in the particle swarm optimization algorithm was first reported in the literature in 1998 (Shi and Eberhart 1998).

$$\vec{v}_{id}(t)=\omega\vec{v}_{id}(t-1)+\varphi_1(\vec{p}_{id}-\vec{x}_{id}(t-1))+\varphi_2(\vec{p}_{gd}-\vec{x}_{id}(t-1)) \qquad (9)$$

$$\vec{x}_{id}(t)=\vec{x}_{id}(t-1)+\vec{v}_{id}(t) \qquad (10)$$

2.3 DISCRETE PSO

In Discrete PSO, the search space is a finite set of states. The fitness function is a discrete function. V_{max} and - V_{max} also have discrete values. The velocity and position of discrete PSO velocity and position is calculated using equation (5) and (6).

2.4 PSO ALGORITHM

Algorithm

1. Initialize a population of particles with random positions and velocities on *N* dimensions in the problem space.
2. For each particle, evaluate the desired optimization fitness function in *N* variables. Compare particle's fitness evaluation with its *pbest*. If current value is better than *pbest*, then set *pbest* equal to the current value, and P_i equals to the current location X_i in *N*-dimensional space.
3. Identify the particle in the swarm with the best success so far, and assign its index to the variable *g*.
4. Change the velocity and position of the particle according to equations (5) and (6).
5. Loop to step 2 until a criterion is met, typically a sufficiently good fitness or a maximum number of iterations.

The Flowchart for PSO is given in Fig.1.

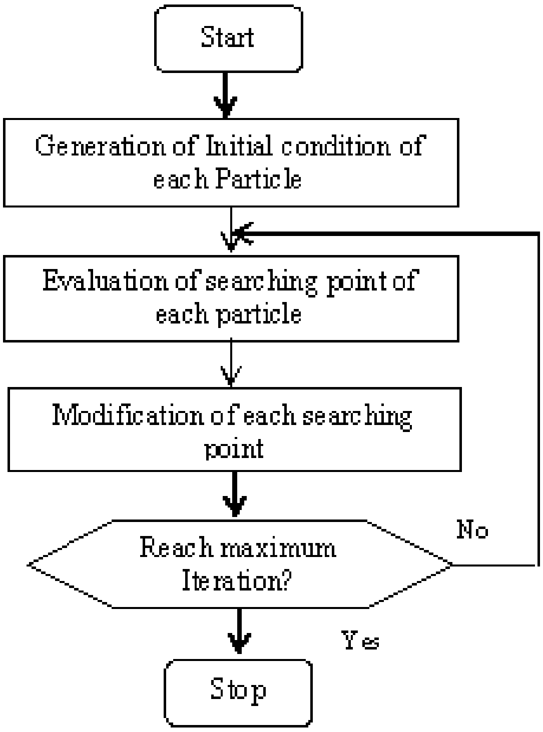
Start
Generation of Initial condition of each Particle
Evaluation of searching point of each particle
Modification of each searching point
Reach maximum Iteration?
No
Yes
Stop

Fig.1. PSO model

3. PSO FOR DOCUMENT CLUSTERING

3.1. PROBLEM STATEMENT

The clustering problem is expressed as follows:

The set of *N* documents *D* = {D_1, D_2,...,D_N} is to be clustered. Each $D_i \in \Re^{N_d}$ is an attribute vector consisting of N_d real measurements describing the object. The documents are to be grouped into non-overlapping clusters $C = \{C_1, C_2, \ldots, C_K\}$ (*C* is known as a clustering), where *K* is the number of clusters, $C_1 \cup C_2 \cup \ldots \cup C_K = D$, $C_i \neq \phi$, and $C_1 \cap C_2 = \phi$ for $i \neq j$.

Assuming $f : D \times D \rightarrow \Re^+$ is a measure of similarity between document feature vectors. Clustering is the task of finding a partition $\{C_1, C_2, \ldots, C_K\}$ of *D* such that $\forall i, j \in \{1, \ldots, K\}, j \neq i, \forall x \in C_i : f(x, O_i) \geq f(x, O_j)$ where O_i is one cluster representative of cluster C_i.

The goal of clustering is stated as follows:

Given,

1. A set of documents *D* = {D_1, D_2, ..., D_N},
2. A desired number of clusters K, and
3. An objective function or fitness function that evaluates the quality of a clustering, the system has to compute an assignment *g*: *D*→{1,2,.....,*K*}and maximizes the objective function.

The global maximization problem can be defined as follows [9]: Given $f : S \rightarrow \Re$ where $S \subseteq \Re^N$ and N is the dimension of the search space *S*. Find $y \in S$ such that $f(y) \geq f(z), \forall z \in S$. The variable *y* is called the global maximizer of *f* and *f*(*y*) is called the global maximum. The process of finding the global optimal solution is known as global optimization (Gray et al 1997). A true global optimization algorithm will find *y* regardless of the selected starting point $z_0 \in S$ [13]. The variable y_L is called the local maximizer of *L* because $f(y_L)$ is the largest value within a local neighborhood, *L*. Mathematically speaking, the variable y_L is a local maximizer of the region *L* if $f(y_L) \geq f(z), \forall z \in L$ where $L \subset S$.

For clustering, two measures of cluster quality are used. One type of measure allows comparing different sets of clusters without reference to external knowledge and is called an internal quality measure. The other type of measures evaluates how well the clustering is working by comparing the groups produced by the clustering techniques to known classes. This type of measure is called an external quality measure.

Internal criterion function focuses on producing a clustering solution that optimizes a particular criterion function that is defined over the documents. These documents are part of each cluster and do not take into account the documents assigned to different clusters. The criterion function for the vector-space variant of the K-Means algorithm is, each cluster is represented by its centroid vector and the goal is to find the clustering solution that maximizes the similarity between each document and the centroid of the cluster that is assigned to.

The proposed system applies global searching strategies for identifying optimal clusters in the exhaustive search space. Typical objective function in clustering formalizes the goal of achieving high intra-cluster similarity, where documents within a cluster are similar, and low inter-cluster similarity, where documents from different clusters are dissimilar. This is an internal criterion for the quality of a clustering.

The objective function used for document clustering in the proposed systems is given in equation (11) as follows:

$$h_f = \frac{1}{N_c}\sum_{i=1}^{N_c}\frac{S_i}{P_i} \qquad (11)$$

where

P_i - Number of documents, which belongs to cluster C_i

N_c - Number of clusters.

S_i is the cosine similarity measure, $\sum_{j=1}^{P_i}\frac{M_{ij}\bullet O_i}{\|M_{ij}\|\times\|O_i\|}$. It finds the similarity between the document vectors and centroid which belong to the cluster.

M_{ij} - j^{th} document vector belongs to cluster i.

O_i -Centroid vector of the i^{th} cluster, $\frac{1}{P_i}\sum_{\forall D_i \in C_i} D_i$

It finds the similarity between documents and centroid of cluster. While grouping, the documents within a cluster have high similarity and are dissimilar to documents in other clusters. The document is placed into a cluster based on high similarity with the cluster centroid using cosine similarity measure. Hence for obtaining an optimal solution for the proposed system is maximization of fitness function.

3.2 DOCUMENT VECTORIZATION

It is necessary to convert the document collection into the form of document vectors. Firstly, to determine the terms that is used to describe the documents, the following procedure is also used in earlier experiments [6]. Extraction of all the words from each document.

- Elimination of the stopwords from a stop word list generated with the frequency dictionary of [8]
- Stemming the remaining words using the Porter Stemmer which is the most commonly used stemmer in English [10]
- Formalizing the document as a dot in the multidimensional space and represented by a vector d, such as $d=\{ w_1, w_2, ..., w_n \}$, where w_i $(i = 1,2,...,n)$ is the term weight of the term $\boldsymbol{t}_i$ in one document. The term weight value represents the significance of this term in a document. To calculate the term weight, the occurrence frequency of the term within a document and in the entire set of documents must be considered. The most widely used weighting scheme combines the Term Frequency with Inverse Document Frequency (TF-IDF) [4][12]. The weight of term ***i*** in document ***j*** is given in equation (12)

$$W_{ji} = tf_{ji}\ x\, idf_{ji} = tf_{ji}\ x \log_2(n/df_{ji}) \qquad (12)$$

where tf_{ji} is the number of occurrences of term ***i*** in the document j; df_{ji} indicates the term frequency in the collections of documents; and n is the total number of documents in the collection.

3.3 DIFFERENT REPRESENTATION OF PSO

3.3.1 Simple PSO (SPSO)

The original PSO is basically developed for continuous optimization problems. In the context of clustering, a single particle represents the cluster centroid vectors. That is, each particle X_i is constructed as follows:

$$X_i = \lfloor M_{i1} \quad M_{i2} \quad \cdots \quad M_{ij} \quad \ldots \quad M_{iK} \rfloor$$

Fig.2. Simple PSO

Fig.2. M_{ij} refers the j-th cluster centroid vector of i-th particle in cluster C_j. Therefore, a swarm represents a number of candidates clustering for the current document vectors.

3.3.2 Binary PSO (BPSO)

Kennedy and Eberhart (1997) have adapted the PSO to search in binary spaces. In the proposed system, each particle maintains a 2-dimensional bit map of order K x N is used to represent the clustering where K is the number of clusters and N is the number of documents. Each particle X_i can be represented as shown in Fig.3. A 1 in row 2 column 3 stands for the document 3 belongs to cluster 2. Each column contains exactly single 1.

$$X_i = \begin{array}{c} \\ 1 \\ 2 \\ \vdots \\ K \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & \cdots & N \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 1 & 0 & \cdots & 0 \end{bmatrix} \end{array}$$

Fig.3. Representation of BPSO

3.3.3 Discrete PSO (DPSO)

The dimension of the particle is the label which the document belongs to. Specifically if the number of cluster is K, each dimension of the particle is an integer value in the range $\{1,2,\ldots,K\}$. That is a clustering of N documents as a string of N integers where the ith integer signifies the cluster number of the ith object. An example of particle is reported in Fig.4.

$$X_i = [1 \quad 2 \quad 1 \quad 3 \quad \ldots \quad 2]$$

Fig.4. Representation of DPSO

3.4 INITIAL POPULATION

One particle in the swarm represents one possible solution for clustering the document collection. Therefore, a swarm represents a number of candidate clustering solutions for the document collection. At the initial stage, each particle randomly chooses K different document vectors from the document collection as the initial cluster centroid vectors. For, each particle assigns a document vector from the document collection to the closest centroid cluster.

3.5 PERSONAL BEST AND GLOBAL BEST POSITIONS OF PARTICLE

The personal best position of partical is calculated as follows:

$$P_{id}(t+1) = \begin{cases} P_{id}(t) & if\ h_f(X_{id}(t+1)) < f(P_{id}(t)) \\ X_{id}(t+1) & if\ \ h_f(X_{id}(t+1) \geq f(P_{id}(t)) \end{cases} \qquad (13)$$

Here t represents the time epoch. The particle to be drawn toward the best particle in the swarm is the global best position of each particle. At the start, an initial position of the particle is considered as the *pbest* and the *gbest* can be identified with maximum fitness function value.

3.6 FINDING NEW SOLUTIONS

According to its own experience and those of its neighbors, the particle adjusts the centroid vector position in the vector space at each generation. The new velocity is calculated based on equation (1) for BPSO equation (5) for SPSO and DPSO. The position is updated based on equation (2) for BPSO and equation (6) for SPSO and DPSO. In DPSO the real values are discretized to integer values.

4. EXPERIMENT RESULTS

The search process of PSO algorithm is a process consists of both narrowing and expansion so that it can have the ability to escape from local maxima, and eventually find good enough solutions. The PSO with real, discrete and binary value representations are tested on the document collections which are given in Table 1.

Table.1. Test Document collection

Document Corpus Contents	Size	No. of Terms
Library Science	82	972
Information Science	1460	6965
Aeronautics	1400	6965

There is no hard and fast rule as to how many particles can be used to solve a specific problem. A large number of particles allow the algorithm to explore the search space faster; however, the fitness function needs to be evaluated for each particle. The number of particles will have a huge impact on the speed at which the simulation will run. From the earlier research, done by Eberhart and Shi (1998), it is proved that the performance of the standard algorithm is not sensitive to the population size but to the convergence rate. The size of the population also affects the convergence of the solution. Based on these results, the population is fixed at 20 particles in order to keep the computational requirements low.

The maximum number of iterations allowed for the fitness value to converge with the optimal solution is set as 40. The inertia weight *w*, in the velocity vector update equation is a scaling variable that controls the influence of the previous velocity while calculating the new velocity. Inertia weight values larger than one will typically cause the particle to accelerate and explore larger regions of the search space; while smaller values will cause the particle to gradually slow down and do a finer search of a region (Van den Bergh and Engelbrecht 2004). The parameters φ_1 and φ_2 are not critical for PSO's convergence. However, proper fine-tuning may result in faster convergence and lessening of local maxima. PSO has very little parameters to fine tune. Different inertia weights *w*, acceleration constants φ_2 and φ_2 have been chosen for simulation. Sensitivity analysis for parameters of PSO algorithm is carried out with different combinations of each parameter. To find the optimal parameters for population size and maximum number of iterations, a thorough sensitivity analysis is carried out for different combinations of the parameter settings, from which it can be observed that the maximum fitness value is at population size of 20 and the maximum number of iterations 40. For each selected *w*, c_1 and c_2, the fitness value obtained from simple PSO is recorded. It has been found that when $w = 0.9$, $\varphi_1 = 2.1$ and $\varphi_2 = 2.1$ the run finds better optimum than all other *w*, φ_1 and $\varphi_{2.}$ The number of clusters is set as 3.

The number of dimensions in each particle of SPSO is set as the cluster size *K*. The document collection size *N* is the DPSO particle dimension size. In BPSO, the particle dimension is *K* × *N*. In SPSO each dimension represents the cluster centroid and it has real value. The discrete value is assigned to the dimension of DPSO and it is derived from where the document coordinates. Binary values are assigned to the dimensions of BPSO based on the existence of document in the clusters. Thus the input size which represents the particle of BPSO is the highest among particle representation models. Fig.5. compares the fitness values obtained from the above PSO models. It shows that the DPSO finds an admissible solution from others. This is due to dimensions and values assigned to those dimensions. In this implementation, DPSO model outperforms the other proposed models SPSO and BPSO.

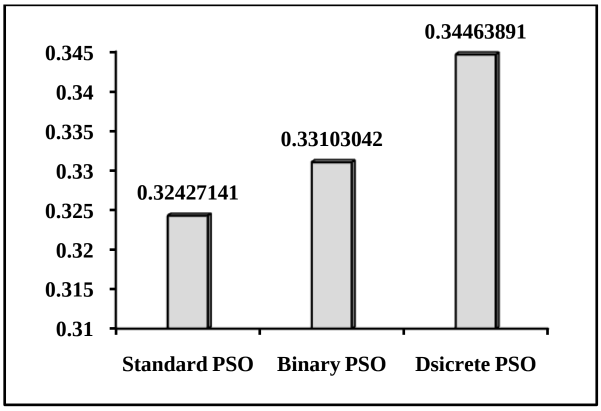

Fig.5. Fitness value obtained from SPSO, BPSO and DPSO

CONCLUSIONS

PSO methodology is examined for document clustering problem. It is found that the document clustering problem is effectively tackled with PSO methodology by optimizing for clustering operation. An important advantage of the PSO is its ability to cope with local optima by maintaining, recombining and comparing several candidate solutions simultaneously. In contrast, local search heuristics algorithm only refines a single candidate solution and is notoriously weak in cope with local optima. In general, PSO has very faster convergence in finding global optimal solutions for numerical optimization as well as document clustering.

REFERENCES

[1] Baeza-Yates R. and Ribeiro-Neto B, 1999, "Modern Information Retrieval", Addison Wesley, ACM Press, London.

[2] Eberhart R.C. and Shi Y., 2000, "Comparing Inertia Weights and Constriction Factors in Particle Swarm

Optimization", Proceedings of the Conference on Congress on Evolutionary Computation, San Diego, CA, pp 84-88.

[3] Eberhart. C Y.H. Shi., 1998, "Evolving Artificial Neural Networks", In Proc. Int. Conf. on Neural Networks and Brain.

[4] Everitt, B., 1980, "Cluster Analysis", 2nd Edition, Halsted Press, New York.

[5] Gray P., Hart W.E., Painton L., Phillips C., Trahan M. and Wagner J, 1997, "A Survey of Global Optimization Methods", Technical Report, Sandia National Laboratory.

[6] Guerrero Bote, V. P., & Moya Aneg_on, F, 2001, "Reduction of the dimension of a document space using the fuzzified output of a Kohonen network", Journal of the American Society for Information Science and Technology, Vol. 52, pp. 1234– 1241.

[7] Kennedy J. and Eberhart R.C. 1997, "A Discrete Binary Version of the Particle Swarm Algorithm", Proceedings of the Conference on Systems, Man, and Cybernetics, Piscataway, NJ: IEEE Service Center, pp. 4104-4109.

[8] Kucera, H., & Francis, N. 1967, "Computational analysis of present-day American English", Providence, RD: Brown University Press,

[9] Pardalos P.M., Migdalas A. and Burkard R.E. 2002, "Combinatorial and Global Optimization, Series on Applied Mathematics", World Scientific, Vol. 14.

[10] Porter, M. F. 1980, "An algorithm for suffix stripping. Program", Vol. 14, No. 3, pp. 130–137.

[11] Salton G. 1971, "The SMART Retrieval System - Experiment in Automatic Document Processing", Prentice-Hall, Englewood Cliffs, New Jersey.

[12] Salton G., Wong A. and Yang C. 1975, "A Vector Space Model for Automatic Indexing", Journal of Communications of the ACM, Vol. 18, pp. 613–620.

[13] Van den Bergh F. and Engelbrecht A.P. 2004, "A Cooperative Approach to Particle Swarm Optimization", IEEE Transactions on Evolutionary Computation, pp.225-239.