Int. J. Advance. Soft Comput. Appl., Vol. 2, No. 3, November 2010
ISSN 2074-8523; Copyright © ICSRS Publication, 2010
www.i-csrs.org

Hybrid PSO and GA models for Document Clustering

K. Premalatha, A.M. Natarajan

Bannari Amman Institute of Technology, Erode, TN, India
e-mail:kpl_barath@yahoo.co.in

Bannari Amman Institute of Technology, Erode, TN, India

Abstract

This paper presents Hybrid Particle Swarm Optimization (PSO) - Genetic Algorithm (GA) approaches for the document clustering problem. To obtain an optimal solution using Genetic Algorithm, operation such as selection, reproduction, and mutation procedures are used to generate for the next generations. In this case, it is possible to obtain local solution because chromosomes or individuals which have only a close similarity can converge. In standard PSO the non-oscillatory route can quickly cause a particle to stagnate and also it may prematurely converge on suboptimal solutions that are not even guaranteed to local optimal solution. This work proposes hybrid models that enhance the search process by applying GA operations on stagnated particles and chromosomes. GA will be combined with PSO for improving the diversity, and the convergence toward the preferred solution for the document clustering problem. The approach efficiency is verified and tested using a set of document corpus. Our results indicate that the approaches are feasible alternative to solve document clustering problems.

Keywords: *Particle Swarm Optimization, Genetic Algorithm, Stagnation, Convergence, Hybrid PSO and GA*

1 Introduction

A Simple Genetic Algorithm (SGA) is a computational abstraction of biological evolution that can be used to solve some optimization problems (Goldberg 1989, Holland 1975). The Genetic Algorithm (GA), proposed by Holland (1975), is a probabilistic optimal algorithm that is based on the evolutionary theories. This algorithm is population-oriented. Successive populations of feasible solutions are generated in a stochastic manner following laws similar to that of natural selection. PSO is a population-based search algorithm and is initialized with a population of random solutions, called particles (Kennedy and Eberhart 1997). Unlike in the other Evolutionary Computation techniques, each particle in PSO is also associated with a velocity. Particles fly through the search space with velocities which are dynamically adjusted according to their historical behaviours. Therefore, the particles have the tendency to fly towards the better search area over the course of search process.

The main problem with PSO is that it prematurely converges (Van den Bergh and Engelbrecht 2004) to stable point, which is not necessarily to be minimum. To prevent the occurrence, position update of the stagnated particles is changed. The position update is done through some hybrid mechanism of GA. The idea behind GA is due to its genetic operator's crossover and mutation. By applying crossover operation, information can be swapped between two particles to have the ability to fly to the new search area. The purpose of applying mutation to PSO is to increase the diversity of the population and the ability to have the PSO to avoid the local maxima. In addition to incorporating crossover and mutation operations into PSO, two different approaches to combine PSO with GA are proposed. Here PSO contributes to the hybrid approach in a way to ensure convergence faster. The hybrid mechanism of global search models PSO and GA enhances the search process by improving the diversity as well as converging. The paper is organized as follows: Section 2 provides a general outline of PSO. Section 3 gives an overview of the GA. The Hybrid models for document clustering are discussed in Section 4. Section 5 presents the detailed experimental setup and results for comparing the performance of the Evolutionary PSO algorithm with the standard PSO, GA and K-means approaches.

2 Particle Swarm Optimization

PSO is a population-based search algorithm and is initialized with a population of random solutions, called particles (Hu et al 2004). In PSO, each single solution is like a 'bird' in the search space, which is called 'particle'. All particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the particles with the best solutions so far. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating each generation.

The original PSO formulae define each particle as potential solution to a problem in *N*-dimensional space. The position of particle *i* is represented as $X_i = (x_{i1}, x_{i2}, \ldots, x_{iN})$. Each particle also maintains a memory of its previous best position, represented as $P_i = (p_{i1}, p_{i2}, \ldots, p_{iN})$. A particle in a swarm is moving; hence, it has a velocity, which can be represented as $V_i = (v_{i1}, v_{i2}, \ldots, v_{iN})$

Each particle knows its best value so far (*pbest*) and its position. Moreover, each particle knows the best value so far in the group (*gbest*) among *pbest*s. This information is analogy of knowledge of how the other particles around them have performed. Each particle tries to modify its position using the following information:

1. the distance between the current position and *pbest*
2. the distance between the current position and *gbest*

This modification can be represented by the concept of velocity.

Velocity of each agent can be modified by equation (1). The inclusion of an inertia weight in the PSO algorithm was first reported in the literature (Eberhart and Shi 1998),

$$V_{id} = w \times V_{id} + c_1 \times rand() \times (P_{id} - X_{id}) + c_2 \times rand() \times (P_{gd} - X_{id}), \qquad (1)$$

where
- i - Index of the particle , $i \in \{1, \ldots, n\}$
- n - Population size
- d - Dimension, $d \in \{1, \ldots, N\}$
- $rand()$ - Uniformly distributed random variable between 0 and 1
- V_{id} - Velocity of particle *i* on dimension *d*
- X_{id} - Current position of particle *i* on dimension *d*
- c_1 - Determine the relative influence of the cognitive component; Self confidence factor
- c_2 - Determine the relative influence of the social component; Swarm confidence factor
- P_{id} - Personal best or *pbest* of particle *i*
- P_{gd} - Global best *or gbest* of the group
- w - Inertia weight.

The use of the inertia weight *w* has provided improved performance in a number of applications. As originally developed, *w* often is decreased linearly from about 0.9 to 0.4 during a run. Suitable selection of the inertia weight provides a balance

between global and local exploration and exploitation, and results in less iteration on average to find a sufficiently optimal solution.

The constants c_1 and c_2 are known as learning factors. They represent the weighting of the stochastic acceleration terms that pull each particle towards the *pbest* and *gbest* positions. Thus, adjustments of these constants change the amount of stress in the system. Low values allow particles to travel far from target regions before being pulled back, while high values result in unexpected movement toward, the target regions. The cognitive parameter represents the tendency of individuals to duplicate past behaviours that have proven successful, whereas the social parameter represents the tendency to follow the success of others. Generally c_1 and c_2 are set to 2.0 which will make the search cover surrounding regions centered at *pbest* and *gbest*. Also, if the learning factors are equal, the same importance is given to social searching and cognitive searching.

The current position that is the searching point in the solution space can be modified by equation (2),

$$X_{id} = X_{id} + V_{id} \quad , \tag{2}$$

All swarm particles tend to move towards better positions; hence, the best position (i.e. optimum solution) can eventually be obtained through the combined effort of the whole population.

The PSO algorithm is simple in concept, easy to implement and computational efficient. The original procedure for implementing PSO is as follows:

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 (1) and (2).
5. Loop to step 2 until a criterion is met, typically a sufficiently good fitness or a maximum number of iterations.

Fig. 1 shows a general flow chart of PSO.

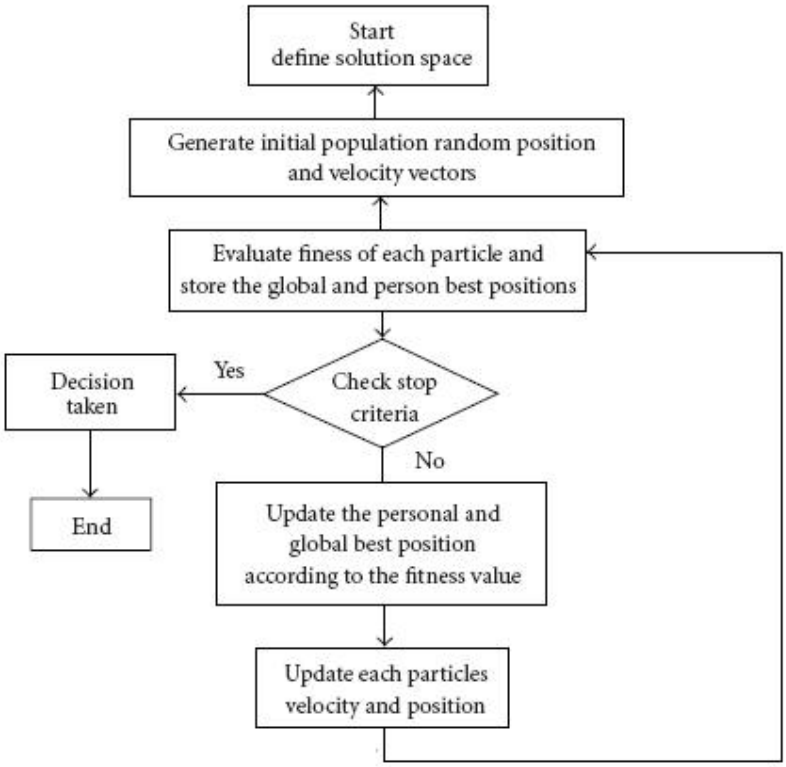

Fig. 1: Flowchart for PSO

3 Genetic Algorithm

A Simple Genetic Algorithm (SGA) is a computational abstraction of biological evolution that can be used to solve some optimization problems (Goldberg 1989, Holland 1975). The Genetic Algorithm (GA), proposed by Holland (1975), is a probabilistic optimal algorithm that is based on the evolutionary theories. This algorithm is population-oriented. Successive populations of feasible solutions are generated in a stochastic manner following laws similar to that of natural selection. These algorithms encode a potential solution to a specific problem on a simple chromosome-like data structure and apply recombination operators to these structures so as to preserve critical information. An implementation of a genetic algorithm begins with a population of (typically random) chromosomes. One then evaluates these structures and allocates reproductive opportunities in such a way that those chromosomes which represent a better solution to the target problem are given more chances to reproduce than those chromosomes which are poorer solutions. The goodness of a solution is typically defined with respect to the current population.

Usually there are only two main components of most genetic algorithms that are problem dependent: the problem encoding and the evaluation function. This can be viewed as a black box with different parameters. The only output of the black box is a value returned by an evaluation function indicating how well a particular combination of parameter settings solves the optimization problem. The goal is to set the various parameters so as to optimize some output. In more conventional terms the system has to maximize (or minimize) some function $F(X_1, X_2, ..., X_m)$

A population is created with a group of individuals created randomly. The individuals in the population are then evaluated. The evaluation function is provided by the programmer and gives the individuals a score based on how well they perform at the given task. It is helpful to view the execution of the genetic algorithm as a two stage process. It starts with the current population. Selection is applied to the current population to create an intermediate population. Then recombination and mutation are applied to the intermediate population to create the next population. The process of going from the current population to the next population constitutes one generation in the execution of a genetic algorithm.

In the construction of the intermediate population from the current population, in the first generation the current population is the initial population. There are a number of ways to do selection. After selection has been carried out the construction of the intermediate population is complete and recombination can occur. This can be viewed as creating the next population from the intermediate population. Crossover is applied to randomly paired strings with a probability denoted *Pc*. Pair of strings are picked. With probability *Pc* these strings are recombined to form two new strings that are inserted into the next population.

After recombination, a mutation operator is applied. For each bit in the population is mutated with some low probability *Pm*. Typically the mutation rate is applied with less than 1% probability. After the process of selection, recombination and mutation is complete, the next population can be evaluated. The process of evaluation, selection, recombination and mutation forms one generation in the execution of a genetic algorithm. Fig. 2 shows a Simple Genetic Algorithm model.

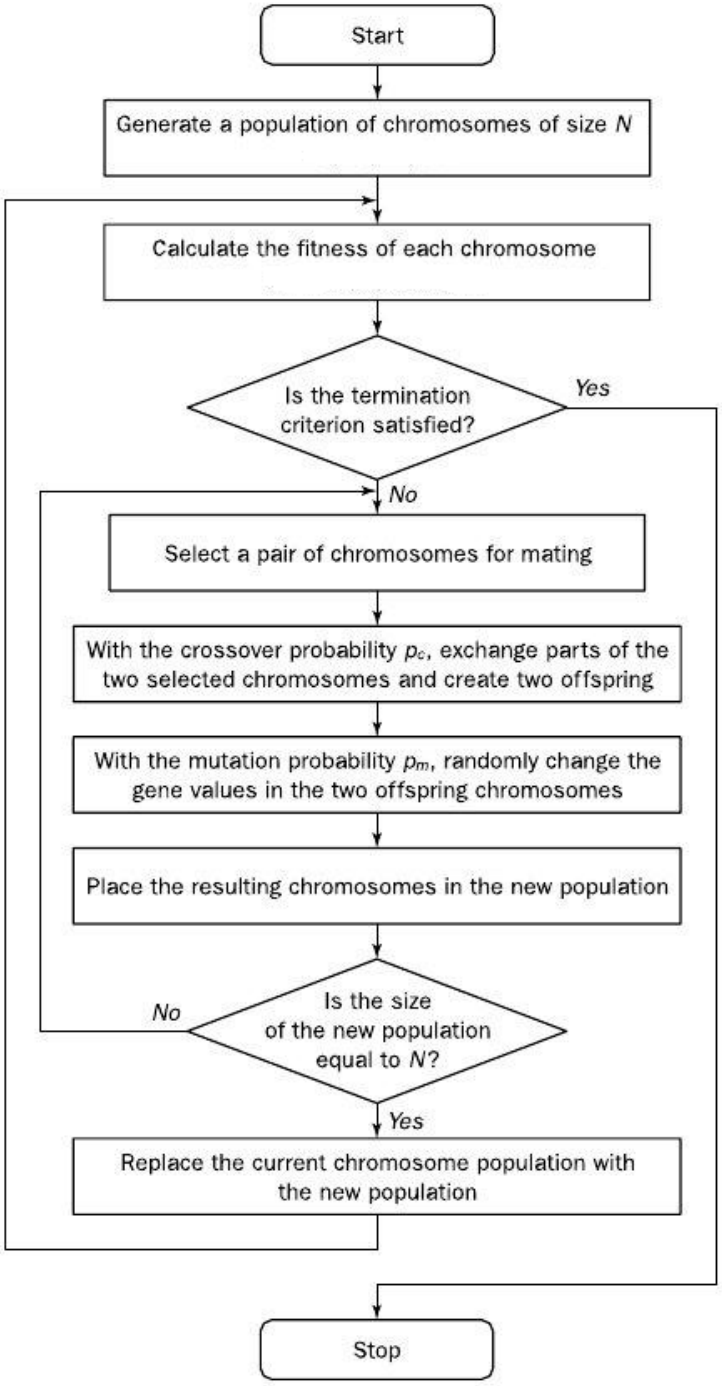

Fig. 2: Simple GA

4 Hybrid PSO and GA for Document Clustering

4.1 Need for Hybrid Mechanism in PSO

PSO is faster in finding quality solutions, however compared to other EC techniques; it faces some difficulty in obtaining better quality solutions while exploring complex functions. It faces premature convergence and suffers from poor fine-tuning capability of the final solution.

The drawback of PSO is that the swarm may prematurely converge. The underlying principle behind this problem is that, for the global best PSO, particles converge to a single point, which is on the line between the global best and the personal best positions. This point is not guaranteed for a local optimum (Van den Bergh and Engelbrecht 2004). Another reason for this problem is the fast rate of information flow between particles, resulting in the creation of similar particles with a loss in diversity that increases the possibility of being trapped in local optima.

Particles' velocities on each dimension are fixed firmly between maximum velocity V_{max} and minimum velocity, V_{min}. If the velocity on a dimension go beyond V_{max} or go below (V_{min}), then the velocity on that dimension is limited to V_{max} or -V_{max}. It consequences a loss in diversity and raises the opportunity of the swarm early convergence.

A further drawback is that stochastic approaches have problem-dependent performance. This dependency usually results from the parameter settings in each algorithm. The different parameter settings for a stochastic search algorithm result in high performance variances. In general, no single parameter setting can be applied to all problems. Increasing the inertia weight (w) will increase the speed of the particles resulting in more exploration (global search) and less exploitation (local search) or on the other hand, reducing the inertia weight will decrease the speed of the particles resulting in more exploitation and less exploration. Thus finding the best value for the parameter is not an easy task and it may differ from one problem to another. Therefore, from the above, it can be concluded that the PSO performance is problem-dependent. The problem-dependent performance can be addressed through hybrid mechanism. It combines different approaches to be benefited from the advantages of each approach.

4.2 Relationship Between PSO And GA

PSO shares many similarities with evolutionary computing techniques in general and GAs in particular. PSO and GA techniques begin with a group of a randomly

generated population; both utilize a fitness value to evaluate the population. They update the population and search for the optimum with random techniques.

In GA techniques, three main operators are involved. They are recombination, mutation and selection operator. PSO does not have a direct recombination operator. However, the stochastic acceleration of a particle towards its previous best position, as well as towards the best particle of the swarm, resembles the recombination procedure in GA (Eberhart and Shi 1998, Rechenberg 1994, Schwefel 1995). In PSO the information exchange takes place only among the particle's own experience and the experience of the best particle in the swarm, instead of being carried from fitness dependent selected parents to descendants as in GA's. Moreover, PSO's directional position updating operation resembles mutation of GA, with a kind of memory built in. This mutation-like procedure is multidirectional both in PSO and GA, and it includes control of the mutation's severity, utilizing factors such as the Vmax and k.

PSO is actually the only evolutionary algorithm that does not use the survival of the fittest concept. It does not utilize a direct selection function. Thus, particles with lower fitness can survive during the optimization and potentially visit any point of the search space (Eberhart and Shi 1998). A large inertia weight facilitates global exploration (search in new areas), while a small one tends to assist local exploration. Information sharing mechanism of PSO and GA are entirely different. In GA, chromosomes share information with each other. So the whole population moves like a one group towards an optimal area. In PSO, only global best gives out the information to others. It is a one-way information sharing mechanism. Compared with GA, all the particles tend to converge to the best solution quickly even in the local version in most cases. The advantages of PSO compared to GA are that PSO is easy to implement and there are few parameters to adjust.

4.3 Hybrid PSO And GA Models

Shi et al (2004) presented a variable population-size genetic algorithm (VPGA) by introducing the dying probability for the individuals and the war/disease process for the population. Based on this VPGA the PSO algorithm is combined.

Li et al (2006) proposed a parallel hybrid PSO-GA algorithm (PHPSO-GA) based on Parallel GA. In PHPSO-GA, subpopulations are classified as several classes according to probability values of improved adaptive crossover and mutation operators. Based on characteristics of different classes of subpopulations, different modes of PSO update operators are introduced for making use of the fast convergence property of particle swarm optimization. Adjustable arithmetic-progression rank-based selection is introduced to prevent the algorithm from

premature in the early stage of evolution and accelerate convergence rate in the late stage of evolution.

Li et al (2008) proposed a evolutionary learning algorithm based on a hybrid of Improved real-code Genetic Algorithm (IGA) and PSO. To overcome the drawbacks of standard GA and PSO, they applied non-linear ranking selection, competition and selection among several crossover offspring and adaptive change of mutation scaling are adopted in the genetic algorithm, and dynamical parameters are adopted in PSO. The new population is produced through three approaches to improve the global optimization performance, which are elitist strategy, PSO strategy and IGA strategy.

Tang et al (2010) proposed the Hybrid PSO/GA for job shop scheduling problem. In this model a new hybrid GA is used to solve the job shop scheduling problem. The particle swarm optimization algorithm is introduced to get the initial population, and evolutionary genetic operations are proposed.

4.4 Problem Statement

The clustering problem is expressed as follows:

The set of N documents $D = \{D_1, D_2, \ldots, D_N\}$ 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 $f : D \times D \to \Re^+$ 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 $D = \{D_1, D_2, \ldots, D_N\}$,
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 $g : D \to \{1, 2, \ldots, K\}$ and maximizes the objective function.

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 (3) as follows:

$$h_f = \frac{1}{N_c} \sum_{i=1}^{N_c} \frac{S_i}{P_i} \quad , \tag{3}$$

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 by maximizing the fitness function.

4.5 Document Vectorization

It is necessary to convert the document collection into the form of document vectors. The following steps are used to convert the document collection into document vectors.

1. Extraction of all the words from each document.
2. Elimination of the stopwords from a stopword list generated with the frequency dictionary of (Kucera, 1967)
3. Stemming the remaining words using the Porter Stemmer which is the most commonly used stemmer in English (Frakes, 1992)

4. 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 t_i in one document. The term weight value represents the significance of the document. To calculate the term weight, the occurrence frequency of the term within a document and entire set of documents must be considered. The most widely used weighting scheme combines the Term Frequency with Inverse Document Frequency (TF-IDF) (Salton, 1975). The weight of term *i* in document *j* is given below

$$W_{ji} = tf_{ji} \times idf_{ji} = tf_{ji} \times \log_2\left(n / df_{ji} \right),$$

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.

4.6 Chromosome (Particle) Representation

The algorithm uses chromosomes (particles) which codify the whole partition P of the data set in a vector of length n, where n is the size of the dataset. Thus, each gene (dimension) of the chromosome (particle) is the label where the single item of the dataset belongs to; in particular if the number of cluster is *k* each gene (dimension) of the chromosome (particle) is an integer value in the range {1..., K}. An example of chromosome is reported in Figure 3.

1	2	3	4	5	6	…	…	n
1	*2*	*1*	*1*	*2*	*3*	…	…	*2*

Fig. 3: Chromosome (Particle) representation

4.7 Initial Generation (Population)

At the initial stage, each individual randomly chooses k different document vectors from the document collection as the initial cluster centroid vectors. For, each individual, a gene (dimension) assigns a document vector from the document collection to the closest centroid cluster. The allele of gene (value of dimension) represents the cluster where the document is present. The initial population can be partitioned into two equivalent divisions for PSO and GA. PSO and GA can be evaluated mutually. The objective function for each individual can be calculated based on the equation (3).

4.8 GA Operators

4.8.1 Selection

In selection the offspring producing individuals are chosen. The first step is fitness assignment. Each individual in the selection pool receives a reproduction probability depending on the own objective value and the objective value of all other individuals in the selection pool. This fitness is used for the actual selection step afterwards. The simplest selection scheme is roulette-wheel selection, also called stochastic sampling with replacement. This is a stochastic algorithm and involves the following technique: The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness. A random number is generated and the individual whose segment spans the random number is selected. The process is repeated until the desired number of individuals is obtained.

4.8.2 Crossover

The interesting behavior arises from genetic algorithms because of the ability of solutions to learn from each other. Solutions can combine to form offspring for the next generation. Sometimes they will pass on their worst information, but doing crossover in combination with a forceful selection technique perceives better solutions result. Crossover occurs with a user specified probability called, the crossover probability P_c. In single point crossover, a position is randomly selected at which the parents are divided into two parts. The parts of the two parents are then swapped to generate two new offspring.

4.8.3 Mutation

The purpose of mutation is to diversify the search direction and prevent convergence to the local optimum. Mutation is a genetic operator that alters one ore more gene values in a chromosome from its initial state. This can result in entirely new gene values being added to the gene pool. With these new gene values, the genetic algorithm may be able to arrive at better solution than was previously possible. Mutation is an important part of the genetic search as helps to prevent the population from stagnating at any local optima. It prevents local searches of the search space and increases the probability of finding global optima Mutation occurs during evolution according to a user-definable mutation probability P_m

4.8.4 Evaluation

After producing offspring they must be inserted into the population. This is especially important, if less offspring are produced than the size of the original

population. Another case is, when not all offspring are to be used at each generation or if more offspring are generated than needed. By a reinsertion scheme is determined which individuals should be inserted into the new population and which individuals of the population will be replaced by offspring. The used selection algorithm determines the reinsertion scheme. The elitist combined with fitness-based reinsertion prevents this losing of information and is the recommended method. At each generation, a given number of the least fit parent is replaced by the same number of the fit offspring.

4.9 PSO Operators

Each particle knows its best value so far *pbest* and its position. This information is equivalence of personal experiences of each agent. Moreover, each agent knows the best value so far in the group *gbest* among *pbest*s. This information is correspondence of knowledge of how the other agents around them have performed.

4.9.1 Personal best & Global best for particles

The personal best position of particle is calculated as follows,

$$P_{id}(t+1) = \begin{cases} P_{id}(t) & if \ \ f(X_{id}(t+1)) \leq f(P_{id}(t)) \\ X_{id}(t+1) & if \ \ f(X_{id}(t+1) > f(P_{id}(t)) \end{cases}.$$

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 ***personal best*** and the ***gbest*** can be identified with minimum fitness function value. The modification of the particles can be represented by the velocity is shown in equation (1) and the current position of the particles can be modified by the equation (2).

4.10 Finding New Solution in Hybrid PSO models

The steps involved in HPSO are similar to standard PSO except in the stagnation behaviour of the particles. After updating the particle position, it is checked for stagnation. The particle swarm system is said to be in stagnation, if arbitrary particle history best position and the total swarm's history best position assign constant over time steps. This situation is named as stagnation behaviour, because after a point, algorithm finishes to generate alternative solutions. When $X_i = pbest_i = gbest$, then the velocity update depends only on the wV_i. If this condition persists over some time steps, then wV_i becomes 0. To prevent the occurrence, position update of the global best particle is changed. Here, to avoid the premature convergence of the swarm, the particles use a hybrid mechanism when they stick at the local maximum and find new particles. The stagnated particles are replaced

by the new particles only if the fitness values of the new particles are high thereby keeping the population size fixed. If the new particles replace the parent, the customized velocity vector is assigned to the new particles.

4.11 Hybrid models with GA operators

In this model each particle's *pbes*t position can be checked for stagnation over a designated time steps. If it does not change its *pbest* position, then it is marked and placed it in selection pool. In PSO with Crossover (PSO-CO), from the pool of marked stagnated particles, two random particles are selected for reproduction and crossover is performed using one point crossover with the user-defined crossover rate on their positions and velocities. This is done until the pool of marked particles is empty. The reason for choosing crossover is its interesting behaviour of the ability of the solutions to learn from each other. Elitist selection is applied on current population and new particles that are obtained from crossover for selecting the particles for next iteration.

Finding new solution in PSO with Mutation (PSO-M) is done on stagnated particles which are stored in selection pool. The selection pool contains the particles do not change their *pbest* positions over designated time steps. The mutation operation with user-defined mutation rate is applied on the selection pool particles positions and their velocities. The mutation operation is significant to the success of GAs, since it expands the search directions and avoids convergence to local optima. Current population particles and new particles are applied to elitist selection for choosing the particles for next population.

In the hybrid model of PSO with Crossover and Mutation (PSO-CM), both crossover and mutation are applied for finding new solution. Particle that doesn't change its *pbest* position can be marked as stagnated and placed in the selection pool. From the pool of marked particles, two random particles are selected and crossover is performed using one point crossover with the user-defined crossover rate on their positions and velocities. Mutation operator is applied on those particles positions and velocities with user defined mutation rate. This is finished until the pool of marked particles is empty. For the next iteration, the particles are selected from the current population and set of newly created particles obtained from GA operators crossover and mutation by Elitist selection

In the previous hybrid models, the GA operators are incorporated in PSO for improving the diversity of the particles. In this HPSO with GA using Swap Operation (HPSO-Swap) model, PSO is combined with GA for good knowledge sharing. The initial population is equally divided for PSO and GA and their operations are done in parallel. In PSO, the particles do not change their *pbest* positions over designated time steps are identified. Those stagnated particles are replaced by the chromosomes from GA and random velocities are assigned. These

chromosomes are selected based on roulette-wheel selection. Similar to HPSO-Swap, the initial population is equally divided for PSO and GA and their operations are done in parallel. But in this HPSO with GA using Crossover Operation (HPSO-CO), the new particles are generated by crossover operation which is performed between the stagnated particle and chromosome from GA. Random Velocity is assigned to the new particles. These kinds of updating result in improving their scores of the fitness.

5 EXPERIMENTAL RESULTS WITH DISCUSSION

The proposed hybrid models are experimented with document collections (Silvia et al 2003) which are shown in table 1 and tested with K-Means and PSO+K-Means proposed by Cui and Potok (2005). The Parameters and their values are shown in Table 2.

Table 1: Test Document collection

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

Table 2: Parameters and their values

Parameter	**Value**
No. of **C**lusters	3
No. of Particles	10
No. of Chromosomes	10
Maximum no. of Populations / Generations / Iterations	40
Designated iterations for stagnation	10
c_1	2.1
c_2	2.1
w	0.9
P_c	0.9
$\boldsymbol{P_m}$	0.01

The results obtained from the experiments are shown in Fig. 4 shows that the fitness value obtained from HPSO-CO and PSO-CO models outperforms the other proposed models. HPSO-CO algorithm generates the highest clustering compact result. In practice, crossover is the principal genetic operator which attempts to preserve the beneficial aspects of candidate solutions and to eliminate undesirable components. Another source of the algorithm is the implicit parallelism inherent in the evolutionary technique. By restricting the reproduction of weak candidates, GA eliminates not only that solution but also all of its descendants. This tends to make the algorithm likely to converge to high quality solutions within a few generations. PSO-CM has the poor fitness value within the proposed HPSO (PSO-CO, PSO-M, HPSO-Swap and HPSO-CO) models. This is because of the random nature of mutation. It may degrade a strong candidate solution than to improve it.

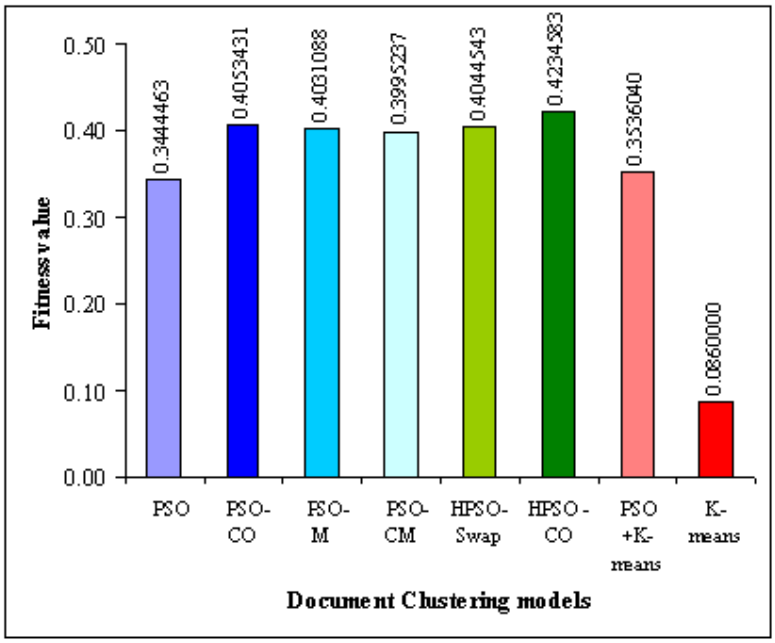

Fig. 4: Evaluation of proposed HPSO systems

In PSO+K-Means (Cui and Potok 2005) global search method, PSO is combined with local search method K-Means. The result of PSO when used as the initial seed of K-Means, the K-Means quickly locates the optima with better fitness value than with random initial seed. It is a result of the K-Means algorithm greatly reliant on initial partition. But the hybrid PSO+K-Means fitness value observed is inferior to the proposed systems. This is because the standard PSO had stagnation, which caused the premature convergence. However, the proposed hybrid models handle the stagnation behaviour and they aim to avoid the premature convergence

of the particles. The hybrid mechanism may not always avoid the stagnation behaviour of the particles. But it tries to diversify the particles position; it avoids the stagnation, which is the source for the improvement in the particles position.

6. Conclusion

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. For a large high dimensional dataset, conventional PSO conducts a globalized searching for the optimal clustering, but it may be trapped in a local optimal area. The HPSO algorithm combines the ability of fast convergence of the PSO algorithm with the competence of ease to exploit previous solution of GA for avoiding the premature convergence. Its success lays in their abilities to extent a large subset of search space. Due to their simplicity and efficiency in navigating large search spaces for optimal solutions, PSO and GA are used in this research to develop efficient, robust and flexible algorithms to solve a document clustering problem.

References

[1] Cui X. and Potok T.E. "Document Clustering Analysis based on Hybrid PSO+K-means Algorithm", *Journal of Computer Sciences (Special Issue)*, (2005). pp. 27-33.

[2] Eberhart R.C. and Shi, Y., "Comparison between Genetic Algorithms and Particle Swarm Optimization", *Evolutionary Programming VII, Lecture Notes in Computer Science*, Springer, New York, Vol. 1447, (1998), pp. 611-616.

[3] Frakes W.B. and Baeza-Yates R. *Information Retrieval: Data structures and Algorithms*, Prentice-Hall, New Jersey, USA, (1992).

[4] Goldberg D.E., *Genetic Algorithms-in Search, Optimization and Machine Learning*, Addison- Wesley Publishing Company Inc., London, (1989).

[5] Holland J., *Adaption in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI, (1975).

[6] Hu X., Shi Y. and Eberhart R.C., "Recent Advances in Particle Swarm", *Proceedings of Congress on Evolutionary Computation*, Portland, Oregon, (2004), pp. 90-97.

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

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

[9] Li G, Zhao F, Guo C and Teng H, "Parallel Hybrid PSO-GA Algorithm and Its Application to Layout Design", *Advances in Natural Computation*, Springer Berlin, Vol. 4221, (2006), pp. 749-758

[10] Li W. T., Shi X. W., Xu L. and Hei Y.Q. , "Improved GA and PSO Culled Hybrid Algorithm for Antenna Array Pattern Synthesis", Progress In Electromagnetics Research, PIER 80, (2008), pp. 461–476

[11] Rechenberg I., "Evolution Strategy", *Computational Intelligence: Imitating Life*, J. M. Zurada, R. J. Marks II, and C. Robinson, (Eds.), IEEE Press, Piscataway, NJ, (1994), pp. 147-159.

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

[13] Schwefel H. P. ,*Evolution and Optimum Seeking*, John Wiley and Sons, New York. (1995),.

[14] Shi, X.H., Liang Y.C., Lee H.P., Lu C. and Wang L.M., "An Improved GA and a Novel PSO-GA-Based Hybrid Algorithm", *Information Processing Letters*, Vol. 93, No. 5, (2005), pp. 255-261

[15] Silvia Acid, De Campos L., Fernandez-Luna J. and Huete J., "An Information Retrieval Model Based on Simple Bayesian Networks", *International Journal of Intelligent Systems*, Vol. 18, (2003), pp. 251-265.

[16] Tang J, Zhang G, Lin B and Zhang B, "A Hybrid PSO/GA Algorithm for Job Shop Scheduling Problem", *Advances in Swarm Intelligence*, Springer Berlin, Vol. 6145, (2010), pp. 566-573

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