World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:2, No:6, 2008

# Exponential Particle Swarm Optimization Approach for Improving Data Clustering

Neveen I. Ghali, Nahed El-Dessouki, Mervat A. N., and Lamiaa Bakrawi



***Abstract*—**In this paper we use exponential particle swarm optimization (EPSO) to cluster data. Then we compare between (EPSO) clustering algorithm which depends on exponential variation for the inertia weight and particle swarm optimization (PSO) clustering algorithm which depends on linear inertia weight. This comparison is evaluated on five data sets. The experimental results show that EPSO clustering algorithm increases the possibility to find the optimal positions as it decrease the number of failure. Also show that (EPSO) clustering algorithm has a smaller quantization error than (PSO) clustering algorithm, i.e. (EPSO) clustering algorithm more accurate than (PSO) clustering algorithm.

***Keywords*—**Particle swarm optimization, data clustering, exponential PSO.

## I. Introduction

CLUSTERING is an important unsupervised classification technique. When used on a set of objects, it helps identify some inherent structures present in the objects by classifying them into subsets that have some meaning in the context of a particular problem. More specifically, objects with attributes that characterize them usually represented as vectors in a multidimensional space, are grouped into some clusters. Clustering algorithms have been applied to a wide range of problems, including exploratory data analysis, data mining, image segmentation and mathematical programming [4], [5].

PSO was originated from computer simulations of the coordinated motion in flocks of birds or schools of fish. As these animals wander through a three dimensional space, searching for food or evading predators, these algorithms make use of particles moving at velocity dynamically adjusted according to its historical behaviors and its companions in an n-dimensional space to search for solutions for an n-variable function optimization problem. The particle swarm optimization algorithm includes some tuning parameters that greatly influence the algorithm performance, often stated as the exploration exploitation tradeoff: Exploration is the ability to test various regions in the problem space in order to locate a good optimum, hopefully the global one. Exploitation is the ability to concentrate the search around a promising candidate solution in order to locate the optimum precisely [7], [10], [11].

El-Desouky et al., in [11] proposed a more enhanced particle swarm algorithm depending on exponential weight variation instead of varying it linearly which gives better results when applied on some benchmarks functions. In this paper we apply the exponential particle swarm (EPSO) algorithm in clustering data sets which when clustered using the linear PSO algorithm gives large number of failures when compared to the proposed method.

The rest of the paper is organized as follows. Section II introduces the standard linear PSO algorithm. EPSO can be found in section III. The EPSO and PSO clustering techniques are discussed in Section IV. Section V gives experimental configuration and results. Conclusions are drawn in the end.

Authors are with Faculty of Science, Al-Azhar University, Cairo, Egypt.

## II. Particle Swarm Optimization

PSO was inspired by the social behavior of a bird flock or fish school. In the PSO algorithm, the birds in a flock are symbolically represented as particles. These particles can be considered as simple agents "flying" through a problem space. A particle's location in the multi-dimensional problem space represents one solution for the problem. When a particle moves to a new location, a different problem solution is generated. This solution is evaluated by a fitness function that provides a quantitative value of the solution's utility [1], [7], [9].

Each particle represents a position in $N_d$ dimensional space, and is "flown" through this multi-dimensional search space, adjusting its position towards both the particle's best position; found thus far and the best position in the neighborhood of that particle.

Each particle *i* maintains the following information: $x_i$ the current position of the particle, $v_i$ the current velocity of the particle must be defined by parameters $v_{\min}$ and $v_{\max}$. The personal best position of the particle is represented by $y_i$.

So the particle's position is adjusted according to

$$v_{i,k}(t+1) = wv_{i,k}(t) + c_1 r_{1,k}(t)(y_{i.k}(t) - x_{i,k}(t)) + c_2 r_{2,k}(t)(\hat{y}_k(t) - x_{i,k}(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

Where *w* is the inertia weight whose range is [0.4, 0.9], $c_1$ and $c_2$ are the learning factors called, respectively, cognitive parameter and social parameter, $r_{1,j}(t), r_{2,j}(t) \sim \mathrm{U}(0,1)$, and k=1,…, $N_d$.

The velocity is thus calculated based on a fraction of the previous velocity, the cognitive component which is a function of the distance of the particle from its personal best position, and the social component which is a function of the distance of the particle from the best particle found thus far (i.e. the best of the personal bests).

The personal best position of particle *i* is calculated as

$$y_i(t+1) = \begin{cases} y_i(t) & \text{if } f(x_i(t+1)) \geq f(y_i(t)) \\ x_i(t+1) & \text{if } f(x_i(t+1)) < f(y_i(t)) \end{cases} \quad (3)$$

Two basic approaches to PSO exist based on the interpretation of the neighborhood of particles. Equation (1) reflects the globalbest (gbest) version of PSO where the neighborhood of each particle is the entire swarm. The social component then causes particles to be drown toward the best particle in the swarm. In the localbest (lbest) PSO model, the swarm is divided into overlapping neighborhoods, and the best particle of each neighborhood is determined so the social component of equation (1) changes to

$$c_2 r_{2,k}(t)(\hat{y}_{j,k}(t) - x_{i,k}(t)) \quad (4)$$

Where $\hat{y}_j$ is the best particle in the neighborhood of the $i^{th}$ particle.

The PSO is executed with repeated application of equation (1), (2) until a specified number of iterations has been exceeded or when the velocity updates are close to zero over a number of iterations [4], [7], [8].

## III. Exponential Particle Swarm Optimization EPSO

In linear PSO, the particles tend to fly towards the gbest position found so far for all particles. This social cooperation helps them to discover fairly good solutions rapidly. However, it is exactly this instant social collaboration that makes particles stagnate on local optima and fails to converge at global optimum. Once a new gbest is found, it spreads over particles immediately and so all particles are attracted to this position in the subsequent iterations until another better solution is found. Therefore, the stagnation of PSO is caused by the overall speed diffusion of newly found gbest [11]. An improvement to original PSO is constituted by the fact that *w* is not kept constant during execution; rather, starting from a maximal value, it is linearly decremented as the number of iterations increases down to a minimal value [4], initially set to 0.9, decreasing to 0.4 over the first 1500 iterations if the iterations are above 1500, and remaining 0.4 over the remainder of the run according to

$$w = (w - 0.4)\left(\frac{\text{MAXITER} - \text{ITERATION}}{\text{MAXITER}}\right) + 0.4 \quad (5)$$

MAXITER is the maximum number of iterations, and ITERATION represents the number of iterations.

EPSO has a great impact on global and local exploration it is supposed to bring out the search behavior quickly and intelligently as it avoid the particles from stagnation of local optima by varying this inertia weight exponentially, as given in equation (6), so that the movement of the particles will be more faster and distant from each other.

$$w = (w - 0.4)\, e^{\left(\frac{\text{MAXITER} - \text{ITERATION}}{\text{MAXITER}}\right)^{-1}} + 0.4 \quad (6)$$

## IV. The EPSO, PSO Clustering

In the past several years, PSO has been proven to be both effective and quick to solve some optimization problems. It was successfully applied in many research and application areas [1]. For the purpose of this paper, define the following symbols:

$N_d$ denotes the input dimension

$N_0$ denotes the number of particles to be clustered

$N_c$ denotes the number of cluster centroids as provided by the user

$Z_p$ denotes the $p^{th}$ data vector

$m_j$ denotes the centroid vector of cluster *j*

$n_j$ denotes the number of particles in cluster *j*

$c_j$ is the subset of data vectors that form cluster *j*.

The distance to the centroid is determined using

$$d(Z_p, m_j) = \sqrt{\sum_{k=1}^{N_d} (z_{pk} - m_{jk})^2} \quad (7)$$

where *k* subscripts the dimension.

To calculate the cluster centroid vectors, using

$$m_j = \frac{1}{n_j} \sum_{\forall Z_p \in C_j} Z_p \quad (8)$$

Each particle $x_i$ is encoded as follows:

$$x_i = (m_{i,1}, \ldots, m_{i,j}, \ldots, m_{i,N_c}) \quad (9)$$

Where $m_{i,j}$ refers to the $j^{th}$ cluster centroid vector of the $i^{th}$ particle in cluster $C_{ij}$. Therefore a swarm represents a number of candidate clusterings for the current data vectors.

The fitness of particles is measured as the quantization error is

$$q_e = \frac{\sum_{j=1}^{N_c} [\sum_{\forall Z_p \in C_{ij}} d(Z_p, m_j) / N_0]}{N_c} \quad (10)$$

Where *d* is defined in equation (7), and $N_0$ denotes the number of data vectors to be clustered.

### A. PSO Clustering

According to the standard global best PSO, data vectors can be clustered as follows

1) Initialize each particle to contain $N_c$ randomly selected cluster centroids.

2) for t = 1 to $t_{max}$ do

for each particle *i* do

for each data vector $Z_p$

calculate the Euclidean distance $d(Z_p, m_{i,j})$ to all cluster centroids $C_{ij}$

assign $Z_p$ to cluster $C_{ij}$ such that

$$d(Z_p, m_{ij}) = \min_{\forall c=1,\ldots,N_c} \{d(Z_p, m_{ic})\}$$

calculate the fitness using equation (10)

update the global best and local best positions

update the cluster centroids using equations (1) and (2)

.

### B. EPSO Clustering

In EPSO clustering algorithm we execute the PSO clustering algorithm but when we update the cluster centroids using equations (1) and (2) we use exponential inertia weight as given in equation (6) instead of linear inertia weight which given in equation (5).

## V. Experimental Studies

The main purpose is to compare the quality of the EPSO and PSO clustering, where the quality of the clustering is measured according to the quantization error, the intra–cluster distances, i.e. the distance between data vectors within a cluster, where the objective is to minimize the intra-cluster distances; and the inter–cluster distances, i.e. the distance between the centroids of the clusters, where the objective is to maximize the distance between clusters. The latter two objectives respectively correspond to crisp, compact clusters that are well separated [8].

### A. Classification Problems

We used five classification problems to compare the performance of the EPSO and PSO algorithms. The classification problems used for this paper are downloaded from Machine Learning Repository sited on http://archive.ics.uci.edu/ml/datasets

- Breast cancer: The Wisconsin breast cancer dataset contains 9 relevant attributes and 2 classes. The objective is to classify each data vector into benign or malignant tumors with 699 instances.
- Iris plants: the dataset contains 3 classes with 4 attributes, where each class refers to a type of iris plant with 150 instances.
- Yeast: this dataset contains 10 classes with 8 attributes, where each class refers to site of protein in the cell with 1484 instances.
- Glass: this dataset contains 5 classes with 9 attributes and 214 instances.
- Lenses: this dataset contains 3 classes with 4 attributes and 24 instances.

### B. Experimental Settings

EPSO and PSO clustering are applied on the five datasets, respectively. The Euclidian distance measure is used as the similarity metrics in each algorithm. For an easy comparison, the EPSO and PSO approaches run 1000 iterations in each experiment. For all the result reported, averages over 30 simulations are given. $c_1 = c_2 = 1.49$ and $w$ inertia weight is according to equation (6) and (5) respectively. We choose number of particles as a function of number of classes. In Breast cancer we choose 10 particles, in Iris plants database we choose 15 particles, in Yeast database we choose 50 particles, in Glass database we choose 25 particles, in Lenses database we choose 15 particles .These values were chosen to ensure good convergence.

### C. Results and Discussion

Table I summarizes the results obtained from the two clustering algorithms for the problems above. First, consider the fitness of solutions, i.e. the quantization error. For all problems, the EPSO algorithm had the smallest average quantization error in all problems, while the PSO algorithm had a large quantization error This table also illustrates that PSO fail to reach the optimal minimum in some runs, but the EPSO successes to reach the optimal minimum in all runs for all problems except for Iris plants database problem has less number of failures than the linear PSO. Even with these failures, we notice that EPSO reaches best minimum points better than the linear PSO.

TABLE I
PERFORMANCE COMPARISON OF EPSO, PSO ALGORITHMS

| Problem | Algorithm | Quantization Error | Failure |
|---|---|---|---|
| Breast cancer | PSO | 20.679 | 3 |
| | EPSO | 0.74576 | 0 |
| Iris | PSO | 16.9281 | 26 |
| | EPSO | 0.60702 | 2 |
| Yeast | PSO | 36.1873 | 20 |
| | EPSO | 1.49539 | 0 |
| Glass | PSO | 30.5105 | 23 |
| | EPSO | 1.38083 | 0 |
| Lenses | PSO | 17.8098 | 22 |
| | EPSO | 0.61398 | 0 |

Second, Fig. 1 to Fig. 5 illustrate the convergence behavior of the EPSO, PSO algorithms for the five classification problems. The linear PSO algorithm exhibited a faster, but premature convergence to a large quantization error, while the EPSO had a slower convergence, but to higher quantization error.

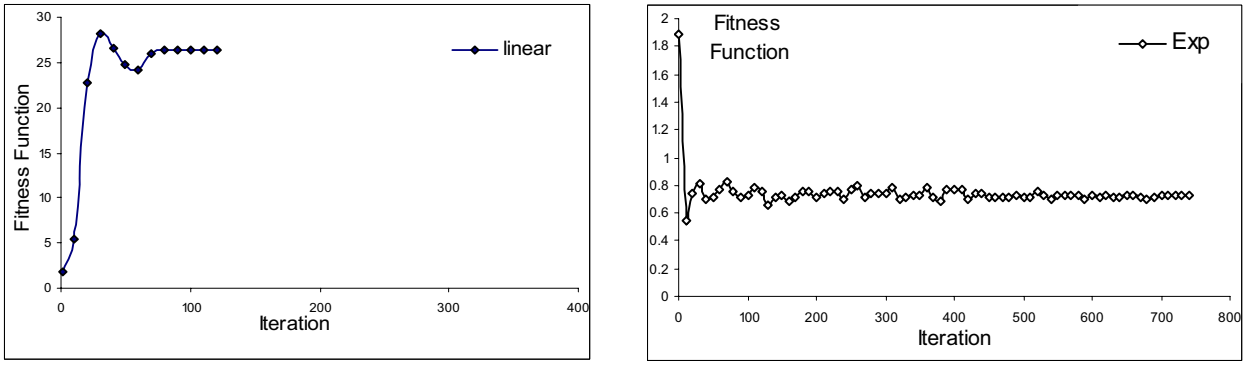


Fig. 1 Algorithm convergence for Breast cancer

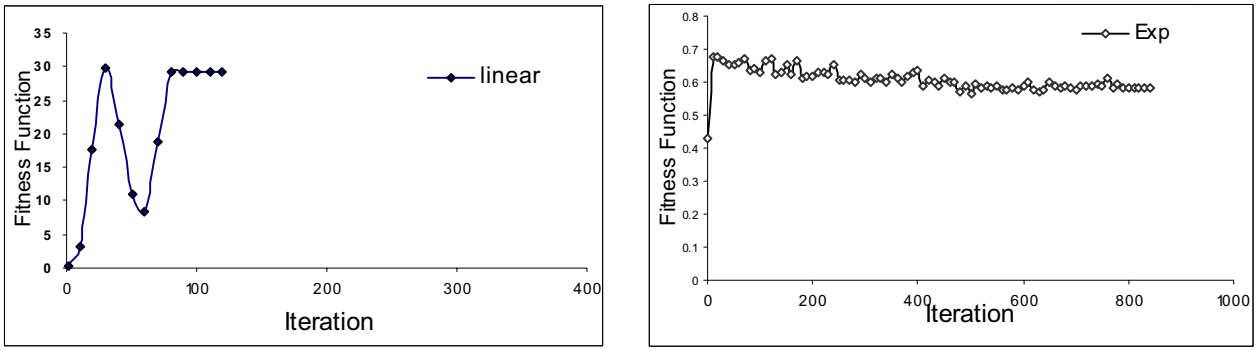


Fig. 2 Algorithm convergence for Iris plant

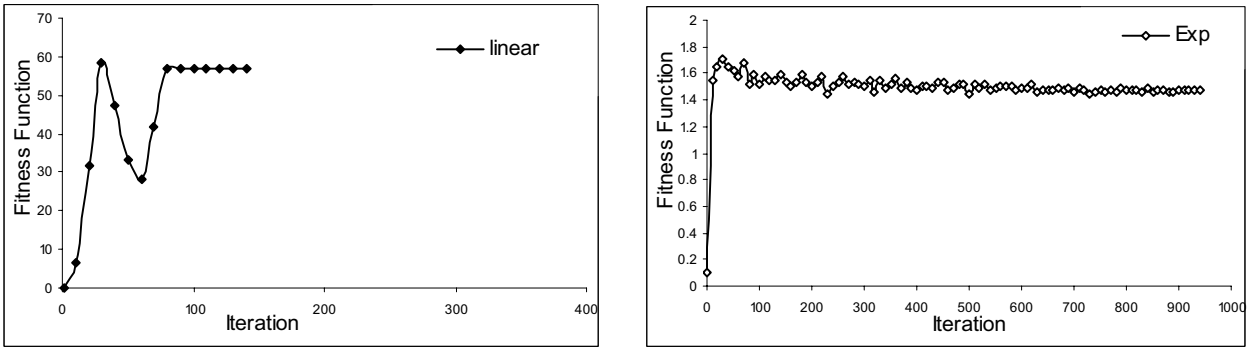


Fig. 3 Algorithm convergence for Yeast

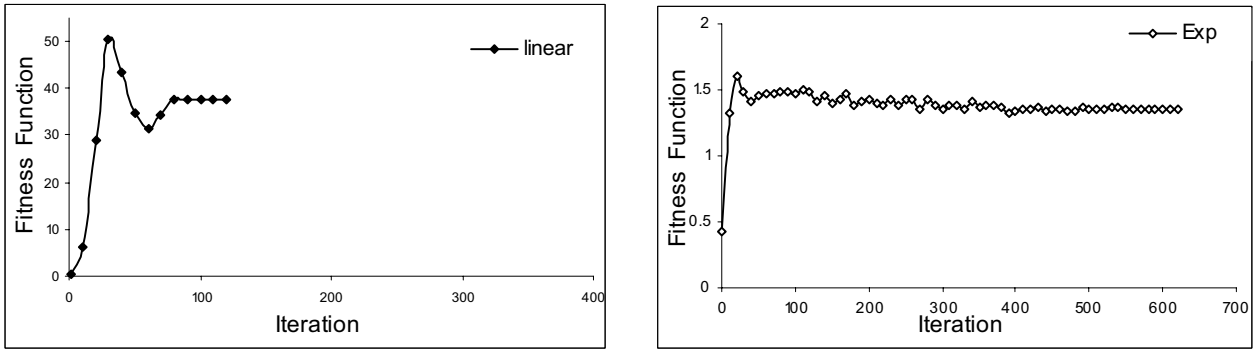


Fig. 4 Algorithms convergence for Glass

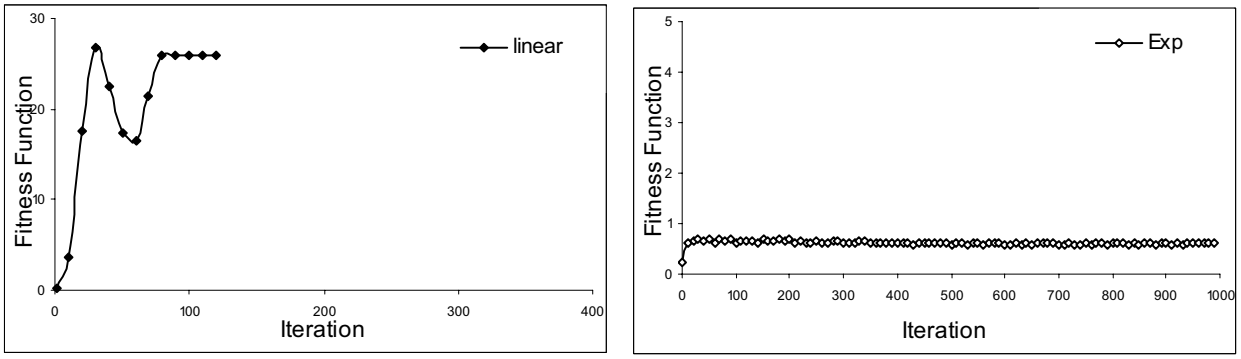


Fig. 5 Algorithm convergence for Lenses

## VI. Conclusion

This paper investigated the application of the EPSO to cluster data vectors. The EPSO algorithm was compared against the PSO clustering algorithm which showed that the EPSO convergence slower to lower quantization error, while the PSO convergence faster to a large quantization error. Also the proposed EPSO increases the possibility to find the optimal positions as it decrease the number of failure.

## References

[1] Cui, X., Potok, T., Palathingal, P., *Document Clustering using Particle Swarm Optimization*, Swarm Intelligence Symposium, 2005. Proceedings 2005 IEEE, pp. 185- 191

[2] Cui, X., Potok T., *Document Clustering Analysis based on Hybrid PSO+K-means Algorithm*, Journal of Computer Sciences (special issue), pp. 27-33, 2005.

[3] Falco, I., Cioppa, A., Tarantino, E., *Facing Classification Problems with Particle Swarm Optimization*, Applied Soft Computing, Vol.7, pp. 652-658, 2007

[4] Jain, A., Murty, M., Flynn, P., *Data Clustering: A Review*, ACM Computing Surveys, Vol. 31, No. 3, 1999.

[5] Kao, Y. –T. et al., *A Hybridized Approach to Data Clustering*, Expert systems and applications (2007), doi: 10.1016/j.eswa.2007.01.028

[6] Kennedy, J., Eberhart, R., *Particle Swarm Optimization*, proceedings of the IEEE International joint conference or Neural networks, vol.4, pp. 1942-1948, 1995.

[7] Li-ping, Z., Huan-jun, Y., Shang-xu, H., *Optimal Choice of Parameters for Particle Swarm Optimization*, Journal of Zhejiang University Science, Vol. 6(A)6, pp.528-534, 2004.

[8] Merwe DW., Engelbrecht AP., *Data Clustering using Particle Swarm Optimization*, IEEE Congress on Evolutionary Computation, Canberra, Australia, 215-220, 2003

[9] Shi, Y., Eberhart, R., *Parameter Selection in Particle Swarm Optimization*, proceedings of the 7th International Conference on Evolutionary Programming VII, pp. 591 – 600, 1998.

[10] Sousa, T., Silva, A., Neves, A., *Particle Swarm Based Data Mining Algorithms for Classification Tasks*, Parallel Computing 30, pp. 767-783, 2004.

[11] El-Desouky N., Ghali N., Zaki M., *A New Approach to Weight Variation in Swarm Optimization*, proceedings of Al-azhar Engineering, the 9th International Conference, April 12 - 14, 2007.