34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Optimization Strategies for Rapid Centroid Estimation

Mitchell Yuwono, Steven W. Su, Bruce D. Moulton, and Hung T. Nguyen

Abstract—**Particle swarm algorithm has been extensively utilized as a tool to solve optimization problems. Recently proposed particle swarm–based clustering algorithm called the Rapid Centroid Estimation (RCE) is a lightweight alteration to Particle Swarm Clustering (PSC). The RCE in its standard form is shown to be superior to conventional PSC algorithm. We have observed some limitations in RCE including the possibility to stagnate at a local minimum combination and the restriction in swarm size. We propose strategies to optimize RCE further by introducing RCE+ and *swarm* RCE+. Five benchmark datasets from UCI machine learning database are used to test the performance of these new strategies. In Glass dataset *swarm* RCE+ is able to achieve highest purity centroid combinations with less iteration (90.3%±1.1% in 9±5 iterations) followed by RCE+ (89%±3.5% in 65±62 iterations) and RCE (87%±5.9% in 54±44). Similar quality is also reflected in other benchmark datasets including Iris, Wine, Breast Cancer, and Diabetes.**

I. Introduction

Cluster analysis has been used in variety of fields including astronomy, medicine, physics, biology, archeology, geology, geography, psychology, and marketing [1]. Recent studies in clustering techniques include the use of computational intelligence [2-5]. Recently there are studies in the swarm intelligence clustering that highlight the advantages of swarm intelligence techniques to conventional K-means for Gaussian data clustering applications [2-5].

Particle swarm optimization (PSO) is a stochastic optimization approach originally proposed by Kennedy & Eberhart in 1995. It is a distributed behavioral model inspired by the behavior of flocks of birds and schools of fish [6]. Inspired by PSO, Particle Swarm Clustering (PSC) was proposed by Cohen & de Castro in 2006 which was shown to be superior to K-means for clustering benchmark datasets [2]. A modified PSC algorithm called Modified PSC (mPSC) was proposed by Szabo in 2010 to reduce PSC update complexity [3]. However Szabo concedes that the improvement is minimal. Addressing the problem of both algorithms in speed, we have previously proposed Rapid Centroid Estimation (RCE), a lightweight modification of PSC in 2012 [4, 5].

The objective of this paper is to further improve RCE by proposing two strategies which includes RCE+ and *swarm* RCE+. We will show the advantages of these strategies compared to standard RCE by performing comparative study of the performances of standard RCE, RCE+ and *swarm* RCE+ on benchmark datasets.

M. Yuwono, S.W. Su, B. Moulton, and H. Nguyen are with the Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo, 2007, NSW, Australia. (e-mail: mitchellyuwono@gmail.com).

Section II presents overview of the algorithm. Section III explains the optimization strategies. Section IV presents experimental results on benchmark datasets. Finally section V provides the conclusion.

II. Overview of The RCE Algorithm

RCE is proposed as a variant of PSC algorithm with reduced time complexity. This algorithm is capable of achieving the performance of PSC with higher stability and faster optimization speed. Whilst PSC algorithm slows down exponentially as dimension and volume increase, RCE reportedly has linear relationship between dimension and volume [4-5]. We have reported that RCE update routine is 274 times quicker than PSC and 270 times quicker than mPSC for a clustering task where the dataset has a dimension of 80 and a volume of 500 [5]. A Monte Carlo test on Synthetic two-class two-dimensional datasets with volume of 500 shows that RCE converged to the appropriate centers at 70 updates on average, compared to 19802 updates for PSC and 23006 updates for mPSC [5].

An RCE particle is a centroid prototype. An RCE group encodes a possible solution to the clustering problem [4-5].

The update rule for RCE can be summarized as follows. For each input pattern *j*, position of a particle *i* we calculate the terms known as *Cognitive* (1), *Social* (2), *Self-organizing* (3), and *Best position* (4):

$$X_i(t) = p_i(t) - x_i(t) \tag{1}$$

$$Y_i(t) = \frac{\sum_{\forall j \in x_i(t)} \hat{\varphi}_{i,j}^{Y} \otimes \left(g_j(t) - x_i(t)\right)}{N_j} \tag{2}$$

$$Z_i(t) = \frac{\sum_{\forall j \in x_i(t)} \hat{\varphi}_{i,j}^{Z} \otimes \left(y_j(t) - x_i(t)\right)}{N_j} \tag{3}$$

$$M(t) = \left[x_1^{best} \quad x_2^{best} \quad x_3^{best} \quad \dots \quad x_{nc}^{best}\right] \tag{4}$$

Where $\bar{\varphi}_{i,j}^{Y}$ and $\bar{\varphi}_{i,j}^{Z}$ are the subjectivity level towards an input pattern, modeled using uniform random numbers $0 \le \bar{\varphi}_{i,j}^{Y} \le 1$, $0 \le \bar{\varphi}_{i,j}^{Z} \le 2$. $x_i(t)$ and $p_i(t)$ denote the best position of a particle i in that makes it closest to an input pattern.

$g_j(t)$ represents the position of a particle that has been closest to the input pattern *j*; and

$M(t)$ represents the best position combination that has achieved global minimum according to a given fitness function *f*.

Δx is computed in (5), particle position update is computed in (6).

$$\Delta x_i(t+1) = \omega(t)\Delta x_i(t) + \varphi_1 \otimes \frac{X_i(t) + Y_i(t) + Z_i(t)}{3} \quad (5)$$

$$x_i(t+1) = x_i(t) + \Delta x_i(t+1) \quad (6)$$

Where $\omega(t)$ is the inertia weight, geometrically decreases on every iteration.

The RCE algorithm is given in Figure 1.

The fitness function f can be flexibly changed. In this paper we will use the fitness function to minimize *within-class* (S_w) and maximize *between-class* (S_b) ratio (7-9) [7].

$$S_w = \frac{1}{N}\sum_{\forall i}\sum_{\forall j \in x_i}(y_j - x_i)(y_j - x_i)^T \quad (7)$$

$$S_b = \frac{1}{N}\sum_{\forall i} N_{j \in x_i}(x_i - \mu)(x_i - \mu)^T \quad (8)$$

$$f(t) = \frac{S_w}{S_b} \quad (9)$$

Where N denotes the total number of data points in the set, y_j denotes input pattern j, x_i denotes particle i. $N_{j \in x_i}$ denotes the number of data points that belong to cluster/particle x_i. μ denotes the global mean of the dataset.

The distance used in RCE is obtained from a distance matrix, where columns represent the datum, and rows represents particle. The value 0 in the distance matrix indicates that the point is identical; the value higher than zero indicates that the point is farther away. In this paper we will use two distance metrics. The first metric is Euclidean distance (10), the second metric is Pearson distance (11).

$$D_{Euclidian} = \begin{bmatrix} d(y_1,x_1)^2 & d(y_2,x_1)^2 & \cdots & d(y_N,x_1)^2 \\ d(y_1,x_2)^2 & d(y_2,x_2)^2 & \cdots & d(y_N,x_2)^2 \\ \vdots & \vdots & \ddots & \vdots \\ d(y_1,x_{nc})^2 & d(y_2,x_{nc})^2 & \cdots & d(y_N,x_{nc})^2 \end{bmatrix} \quad (10)$$

$$D_{Pearson} = \begin{bmatrix} 1-\rho(y_1,x_1) & 1-\rho(y_2,x_1) & \cdots & 1-\rho(y_N,x_1) \\ 1-\rho(y_1,x_2) & 1-\rho(y_2,x_2) & \cdots & 1-\rho(y_N,x_2) \\ \vdots & \vdots & \ddots & \vdots \\ 1-\rho(y_1,x_{nc}) & 1-\rho(y_2,x_{nc}) & \cdots & 1-\rho(y_N,x_{nc}) \end{bmatrix} \quad (11)$$

Where D denotes distance matrix, $d(y_j, x_i)$denotes Euclidean distance between input pattern j and particle i, and $\rho(y_j, x_i)$ denotes Pearson correlation between input pattern j and particle i.

III. Optimization Strategies

A. Substitution Strategy

PSC-based algorithm such as RCE has the possibility to stagnate once equilibrium is reached. Based on this observation, we propose a strategy in order to gracefully break the equilibrium state we call the *substitution* method. The motive of *substitution* is to trigger the particles in current RCE neighborhood to reach equilibrium at alternate positions. We propose the name of "RCE+" for an RCE group with this strategy

Algorithm S = RCE(dataset, max_iter, s_max, ε , n_c)
Initialize n_c particles, randomize x,
Calculate distances of p, g for each particle and each datum.
while t < max_iter && s_c <s_max
Update distance matrix
$D(t) = d(y_j, x_i): \forall i, j$
Find the closest data point for each particle
$[Dx^{\min}(t) \quad Ix^{\min}] = \min(D, i)$
Find the closest particle for each data point
$[Dy^{\min}(t) \quad Iy^{\min}] = \min(D, j)$
Update $p_i(t)$, $g_j(t)$, and $M(t)$

$$p_i(t+1): \forall i = \begin{cases} y_{Ix^{\min}}(t) & if \quad Dx_i^{\min}(t) < Dx_i^{\min}(t-1) \\ p_i(t) & otherwise \end{cases}$$

$$g_j(t+1): \forall j = \begin{cases} y_{Iy^{\min}} & if \quad Dy_j^{\min}(t) < Dy_j^{\min}(t-1) \\ g_j(t) & otherwise \end{cases}$$

$$M(t+1) = \begin{cases} x_i(t): \forall i & if \quad f(x_i(t): \forall i) < f(M(t)) \\ M(t) & otherwise \end{cases}$$

Increment s_c if gradient is higher than $-\varepsilon$

$$s_c = \begin{cases} s_c + 1 & if \quad f(M(t+1)) - f(M(t)) > -\varepsilon \\ 0 & otherwise \end{cases}$$

Find the winning particle (the particle with the least Euclidean distance to an input pattern)

$$x_{most\_win}(t) = x(t) \in \min(d(p_i(t) - x_i(t))): \forall i$$

for each particle x
Get the elements which are the members of the particle i (cluster centroid).

$$y_i^{cluster} = \forall y \in x_i(t)$$

$$N_i = size(y_i^{cluster})$$

Calculate position update accordingly using (5-6) if N_i is greater than zero, otherwise redirect trajectory to most_win particle coordinate:

$$x_i(t+1) = \begin{cases} x_i(t) + \Delta x_i(t+1) & if \quad N_i > 0 \\ x_i(t) + \varphi_5 \otimes (x_{most\_win}(t) - x_i(t)) & otherwise \end{cases}$$

end

$$w(t+1) = 0.95\, w(t)$$

$$t = t + 1$$

end

Figure 1: RCE Algorithm

In each update episode, each particle inside the group has equal probability to make a decision to enter the *substitution* state and free itself from its current local group. The probability of a particle to enter this state is set to a very low chance around 1% to 5% per iteration. When a particle initiates the state, the particle ignores update rule (5) and will do update rule (12) for $T^{(S)}$ iterations. Superscript *(S)* indicates the *substitution* state.

$$\Delta x_i^{(S)}(t+1) = \omega(t)\Delta x_i^{(S)}(t) + \varphi_6 \otimes \left(x_{farthest}\left(t_0^{(S)}\right) - x_i^{(S)}(t)\right) \quad (12)$$
$$t_0^{(S)} < t < t_0^{(S)} + T^{(S)}$$

Where $x_i^{(S)}(t)$ denotes the particle that is currently on the *substitution* state. $x_{farthest}\left(t_0^{(S)}\right)$ denotes the farthest particle relative to $x_i^{(S)}$ at the initiation of *substitution* state.

The *substitution* strategy is illustrated in Figure 2.

B. Swarm Strategy

Swarm paradigm in RCE is different to that of conventional PSO. Instead of operating as a single swarm, RCE swarm operates as collaborative groups. A single RCE group is defined by *nc* particles where *nc* is the number of desired centroids. On top of the existing RCE group, we define additional *k-1* groups which are identical duplicates of the first RCE with particles initialized at different randomized positions. Hence an RCE swarm consists of *k* RCE groups working in parallel to solve a given clustering problem. Each group contributes to the whole swarm by sharing its knowledge about the best centroids locations and its best fitness value. We call this the *awareness* of the swarm. On each iteration the swarn synchronizes *awareness* knowledge. On synchronization, knowledge matrix *K* and its weight $K^{(w)}$ is updated. The matrix *K* is a collection of best position coordinates (13). The knowledge weight $K^{(w)}$ is a vector containing fitness value for each group (14).

$$K(t) = [M_1(t) \quad M_2(t) \quad \cdots \quad M_k(t)] \quad (13)$$

$$K^{(w)}(t) = [f(M_1(t)) \quad f(M_2(t)) \quad \cdots \quad f(M_k(t))] \quad (14)$$

Where $M_k(t)$ denotes matrix of best position known to RCE group *k* at iteration *t*. $M_k(t)$ is an *m* by *n* matrix, where *m* indicates dimensionality of the data, and *n* indicates number of particles in the group. *f* is the fitness function. *Awareness* factor *W* is calculated using (15):

$$W_i(t) = \frac{\sum_{\forall k \in x_i(t)} K_k^{(W)} \hat{\varphi}_{i,k}^{W} \otimes (K_k(t) - x_i(t))}{N_k} \quad (15)$$

K_k denotes a knowledge pattern closest to the particle x_i, N_k denotes the number of knowledge patterns closest to the particle x_i that are chosen and $\hat{\varphi}_{i,k}^{W}$ is subjectivity level towards a knowledge pattern modelled using uniform random number $0 \le \hat{\varphi}_{i,j}^{W} \le 1$.

Update rule (5) is then replaced with (16) to include *awareness* factor *W*:

$$\Delta x_i(t+1) = \omega(t)\Delta x_i(t) + \varphi_1 \otimes \frac{W_i(t) + X_i(t) + Y_i(t) + Z_i(t)}{4} \quad (16)$$

IV. Performance Analysis and Discussion

In order to assess the effects of the optimization method on RCE performance, a comparative study on benchmark datasets is performed. The clustering process is repeated 50 times. PCA is used on the experimental datasets to reduce their dimensionalities. The swarm size of *swarm* RCE+ is set to 5 groups, *substitution* probability is set to 2%, *substitution*

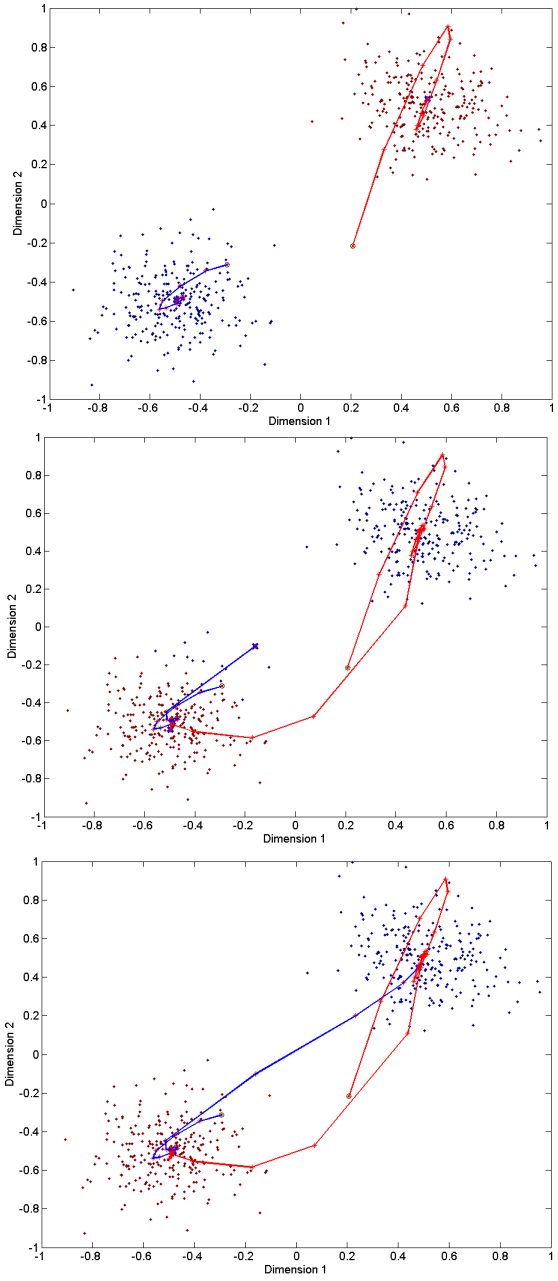

Figure 2: Position Substitution Strategy. Red particle breaks free from current equilibrium position. Blue particle is forced to adapt to this initiative by gravitating to an alternate equilibrium position.

iteration $T^{(s)}$ is set to 5 iterations. Maximum iteration is set to 500. Stagnation count before stopping s_c is set to 150 for RCE, 150 for RCE+, and 50 for *swarm* RCE+. The performances are measured using Entropy (16), Purity (17), and % misclassification (18).

$$E = -\frac{n_r}{n}\sum_{r=1}^{R}\sum_{\forall i \in x_r}\frac{n_r^i}{n_r}\ln\frac{n_r^i}{n_r} \quad (16)$$

$$P = \frac{n_r}{n}\sum_{r=1}^{R}\frac{\max\left(n_r^i\right)}{n_r} \quad (17)$$

$$Pm = \frac{n^{fp}}{n} \quad (18)$$

Entropy measures cluster homogeneity (16). Lower entropy shows that objects in the database are homogenous. r indicates cluster, R is the total number of classes in the cluster, n_r^i is the number of object of class i inside the cluster r. n_r is the number of objects in cluster r, n is the dataset volume.

Purity index measures the purity of the cluster by taking the ratio of the dominant class of the group in relation to the total number of objects inside the group (17). Higher purity is desirable for a good cluster.

Percent misclassification is the ratio of false positive classifications (*fp*) to the number of objects (18). Low percentage of misclassification is a criterion for a good cluster.

Experimental results can be seen in Table I.

TABLE I. EXPERIMENTAL RESULTS

Dataset[a,b] (N/Dim/PCA/[nc]) (distance metric)		**Algorithm**		
		RCE	***RCE+***	***Swarm RCE+ (5 groups)***
Iris (150/4/3/[3]) (Pearson)	E	0.23 ± 0.04	0.24 ± 0.022	0.24 ± 0.021
	P	87.5% ± 1.14%	91.44% ± 0.95%	91.6% ± 0.79%
	Pm	12.47% ± 1.14%	8.56% ± 0.95%	8.4% ± 0.79%
	iter	98 ± 89	102 ± 69	33 ± 25
	t	0.21 ± 0.189	0.218 ± 0.15	0.34 ± 0.2686
Glass (214/9/4/[2]) (Pearson)	E	0.24 ± 0.12	0.287 ± 0.07	0.31 ± 0.02
	P	87% ± 5.9%	89% ± 3.5%	90.3% ± 1.1%
	Pm	13% ± 5.9%	11% ± 3.5%	9.7% ± 1.1%
	iter	54 ± 44	65 ± 62	9 ± 5
	t	0.15 ± 0.13	0.12 ± 0.11	0.06 ± 0.04
Wine (178/13/4/[3]) (Euclidean)	E	0.54 ± 0.08	0.59 ± 0.05	0.60 ± 0.04
	P	62% ± 7%	67% ± 5.6%	68% ± 5%
	Pm	38% ± 7%	33% ± 5.6%	32% ± 5%
	iter	102 ± 87	92 ± 80	10 ± 7
	t	0.21 ± 0.168	0.18 ± 0.159	0.08 ± 0.062
Breast Cancer (699/9/6/[2]) (Euclidean)	E	0.16 ± 0.01	0.158 ± 0.015	0.162 ± 0.0127
	P	96% ± 0.49%	96.08% ± 0.7%	95.89% ± 0.63%
	Pm	4% ± 0.49%	3.92% ± 0.7%	4.11% ± 0.63%
	iter	15 ± 10	9 ± 7	7 ± 5
	t	0.019 ± 0.028	0.02 ± 0.0189	0.083 ± 0.063
Diabetes Pima Indian (768/8/4/[2]) (Euclidean)	E	0.43 ± 0.084	0.46± 0.075	0.48 ± 0.056
	P	65.64% ± 0.4%	65.64% ± 0.49%	65.6% ± 0.57%
	Pm	34.36% ± 0.4%	34.36% ± 0.49%	34.4% ± 0.57%
	iter	10 ± 8	27 ± 15	7 ± 3
	t	0.031 ± 0.016	0.0412 ± 0.0544	0.045 ± 0.023

a. N/Dim/PCA/[nc] denotes volume, dimension, reduced dimension after PCA, and [number of clusters]. Metric denotes distance metric: Pearson Distance/Euclidean Distance

b. E denotes entropy, P denotes purity, Pm denotes percentage misclassification, iter denotes number of iteration needed to obtain solution (n), t denotes time needed to obtain solution (seconds)

Table I suggests *swarm* RCE+ produces results with higher purity and lower standard deviation compared to the other algorithms. Low standard deviation suggests that the results have high repeatability.

Figure 3 visualizes the resulting centroid purity using different algorithms on benchmark datasets.

Figure 4 visualizes the number of iteration needed using different algorithms on benchmark datasets.

The *swarm* RCE+ strategy converges in less number of iteration compared to other algorithms. In Glass dataset *swarm* RCE+ is able to achieve highest purity centroid combinations with less iteration (purity of 90.3%±1.1% in 9±5 iterations) followed by RCE+ (purity of 89%±3.5% in 65±62 iterations) and RCE (purity of 87%±5.9% in 54±44 iterations).

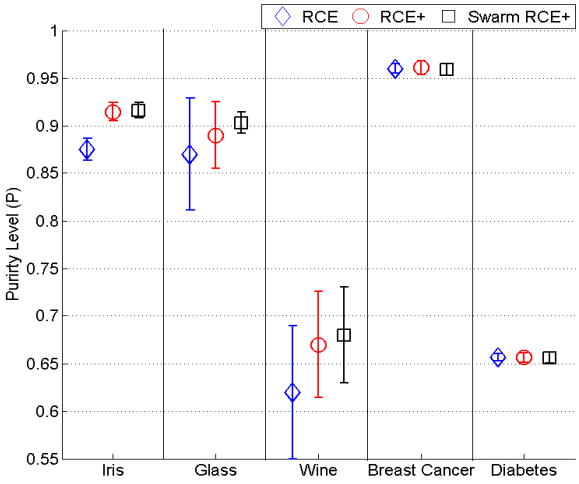

Figure 3: Purity comparison between algorithms

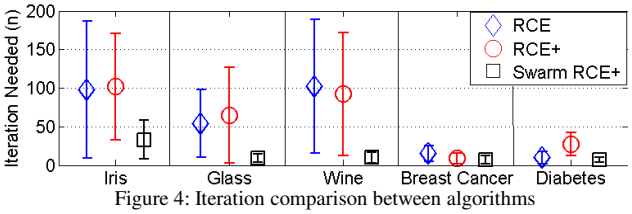

Figure 4: Iteration comparison between algorithms

V. CONCLUSIONS AND FUTURE DIRECTIONS

Two optimization strategies including *substitution* and *swarm* RCE+ have been proposed. Experimental results on benchmark datasets have shown that both optimization strategies improve the performance, repeatability and iteration needed to convergence of a standard RCE. In the future we aim to further reduce the complexity of the current algorithm.

REFERENCES

[1] H. P. Kriegel, P. Kröger, A, Zimek, "Clustering High Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering", in *ACM Transactions on Knowledge Discovery from Data*, vol. 3, no.1, March 2009, article 1.

[2] S.C.M. Cohen & L.N. de Castro, "Data Clustering with Particle Swarms," in *Proc 2006 IEEE Congress on Evolutionary Computation,* 2006, pp.1792-1798.

[3] A. Szabo, A.K.F. Prior, L.N. de Castro, "The Proposal of a Velocity Memoryless Clustering Swarm," in *Proc 2010 IEEE Congress on Evolutionary Computation*, 2010, pp.1-5.

[4] M. Yuwono, S.W. Su, B. Moulton, H. Nguyen, "Method for increasing the computational speed of an unsupervised learning approach for data clustering," in *Proc 2012 IEEE Congress on Evolutionary Computation,* 2012, to be published.

[5] M. Yuwono, S.W. Su, B. Moulton, H. Nguyen, "Fast unsupervised learning method for rapid estimation of cluster centroids," in *Proc 2012 IEEE Congress on Evolutionary Computation,*2012, to be published.

[6] D.W. Merwe and AP Engelhrecht, "Data Clustering using Particle Swarm Optimization," in *Congress on Evolutionary Computation*, vol.1, 2003, pp. 215 – 220.

[7] H. Sarvari, N Khairdoost, A. Fetanat, "Harmony Search Algorithm for Simultaneous Clustering and Feature Selection," In *Proc International Conference of Soft Computing and Pattern Recognition (SoCPaR),* 2010, pp.202-207.