COMPARISON OF NELDER-MEAD, PARTICLE SWARM OPTIMIZATION AND NM-PSO FOR CLUSTERING OF GENE EXPRESSION DATA

[1]Minny Joseph, [2]Jyothi Thomas
Department of Computer Science and Engineering
Christ University Bengaluru, Karnataka, India,
Email: [1]minny.joseph@mtech.christuniversity.in, [2]j.thomas@christuniversity.in

Abstract— The DNA microarray technology concurrently monitors the expression levels of thousands of genes during significant biological processes and across the related samples. The better understanding of functional genomics is obtained by extracting the patterns hidden in gene expression data. Many clustering algorithms have been proposed for the analysis of gene expression data, but little guidance is available in helping to choose one among them. In this paper, we perform clustering on the gene expression data using particle swarm optimization algorithm (PSO), Nelder Mead Simplex method (NM) and NMPSO Method. Here we evaluate the algorithms based on Number of clusters and fitness value. The results show that NM-PSO performs better than NM and PSO.

Keywords—Gene expression data, Particle swarm optimization (PSO), simplex method, fitness function, optimization, genomes, spread out, simplex, multi-minima

I. INTRODUCTION

Gene expression is the method in which information from gene is used for the generation of gene product. Gene expression data is used to interpret genetic code of a sample. The information regarding building and maintain of cells for an organism is carried by genes. The genes are encoded in long strands of DNA in most of the living organisms. Usually DNA is having a double helix structure. It consists of four types of nucleotide subunits to form a chain. The nucleotide subunits are namely adenine, cytosine, guanine and thymine. Guanine pairs with cytosine and adenine pairs with thymine. Transcription and translation are the two steps in gene expression, in which transcription produces messenger RNA from DNA. Messenger RNA or mRNA is single stranded. In the translation step, defined sequences of amino acids are produced from mRNA.

A Micro array experiment evaluates a large number of DNA sequences consisting of genes, cDNA clones or expressed sequence tags under different conditions. Gene expression data set from a micro-array experiment can be represented by a real-valued expression matrix [2]. In this matrix, rows represent expression pattern of genes, columns represent expression profile of samples or experimental conditions[16].

Data sets are represented as set of genes G = $\{g_1, g_2, g_3 \dots g_n\}$, where g_i represent i_{th} gene in the data set and w_{ij} represents expression profile of i_{th} gene at j_{th} samples/conditions[1].

Fig.1, represents dataset with n genes and m samples/conditions vector of real numbers represented as follows.

Sample S

Gene G
w11 w12 w13 …
… w1n w21 w22
w23 … … w2n
w31 w32 w33 …
… w3n

… … … … … …
… … … … … … Wn1 wn2
wn3 … … wnm

Fig. 1 Gene expression data matrix

The expression levels of various genes can be represented by using microarray technology. DNA molecules of various genes are placed in discrete spots of a microscope slide. A simple microarray is an N*M array, where N is the number of genes and the number of conditions is given by M. The row in the array represents a gene and columns represent the conditions [5] [1].

Data mining is an area, where we can extract knowledge from a large database. Knowledge extraction involves many tasks. Clustering is one of the important data mining task which is having a number of applications in the area of biology and other disciplines. Here similar objects are grouped in a cluster [12]. Clustering of gene expression data is helpful to understand gene regulation, gene function and cellular processes. While considering the case of gene expression data, the elements are genes. There is no previously defined class label for clustering.

Mainly two categories of clustering are hierarchical method and partitional method. Hierarchical clustering algorithms break up the data in to a hierarchy of clusters [15]. Partional algorithms divide the data set into disjoint partitions. Partitional method is faster than the hierarchical method but this method has a disadvantage that we have to mention the number of clusters in priori [9].

Clustering solution can be represented in two ways by integer encoding. In the first one, an integer vector of N position is considered as a genotype where N is the number of dataset objects. Each position corresponds to a particular object. The ith gene represents the ith dataset object provided that a genotype represents a partition formed by k clusters. Each gene will have a value between 1 and k and these values represent the cluster label, for example the clustered integer vector can be represented as [1111222233][5] [7].

Another way of representing integer encoding scheme is to make use of an array of k elements to provide a medoid based representation of dataset. Here each array element indicate the index of the object xi, i=1,2,…..N [10]

G1	G2	G3	G4	G5	G6	…	…	…	Gn
1	2	3	1	2	1	…	…	…	…

Fig. 2 Chromosome representation

A fitness function is type of objective function used to summarize, as a single figure of merit, how close a given design solution is to achieving the set aims.

$$Q(N,K)- 1/K! \sum_{I=0}^{k} (-1)i \; ({}^{K}{}_{i}) \; (K-i)^{N\}}$$

For example, for Q (25, 5) there are 2,436,684,974,110,751 ways of sorting 25 objects into 5 groups. If the number of clusters is unknown the objects can be sorted

$$\sum_{k-1}^{k} Q(N,K)$$

ways. For 25 objects this is over $4*10^{18}$. Clearly it is impractical for an algorithm to exhaustively search the solution space to find the optimal solution. Furthermore traditional clustering algorithms search relatively a less subset of the solution space. As a result, the probability of success of these methods is small and it requires for an algorithm with the potential to search large solution spaces effectively [15].

II. OVERVIEW OF ALGORITHMS

A. Nelder Mead's Simplex Method (NM)

Nelder Mead simplex algorithm, is an algorithm that exploits local information and converges to the nearest optimal point. It is an algorithm searching for local minimum and can be used for multi-dimensional optimizations. It does not have to compute derivatives to move along a function as gradient methods[3][11].

Nelder and Mead devised a simplex method for finding a local minimum of a function of several variables. A simplex is a triangle for two variables, and the method is a pattern search that compares function values at the three vertices of a triangle. The vertex where f (x, y) is largest is the worst vertex, which rejected and replaced with a new vertex[4]. A new triangle is formed and the search is continued. A sequence of triangle will be generated, which might have different shapes for which the function values at the vertices get smaller and smaller. The coordinates of the minimum point is found by reducing the size of the triangle. The algorithm will find the minimum of a function of N variables which is computationally compact and effective[1].

1) *Initial Triangle BGW*

Let f(x,y) be the function that is to be minimized. To Let the vertices of the triangle: $V_k= (x_k,y_k)$, k=1,2,3. The function f(x,y) is then evaluated at each of the three points: zk=f(xk,yk) for k=1,2,3. The subscripts are then reordered so that
z1≤z2≤z3. We use the notation

B=(x1,y1),G=(x2,y2) and W=(x3,y3) (2)

2) *Midpoint of the Good Side*

The construction process uses the midpoint of the line segment joining B and G. It is found by averaging the coordinates:

$$M= \frac{B+G}{2} = \left(\frac{x1+x2}{2}, \frac{y1+y2}{2} \right)$$

3) *Reflection using the point R*

The function decreases as we move along the side of the triangle from W to B, and it decreases as we move along the side from W to G. Hence it is feasible that f(x,y) takes on smaller values at points that lie away from W on the opposite side of the line between B and G. We choose a test point R that is obtained by "reflecting" the triangle through the side BG. First find the midpoint M of the side BG to determine R and then draw the line segment from W to M whose length is d. This last segment is extended a distance d through M to locate the point R.

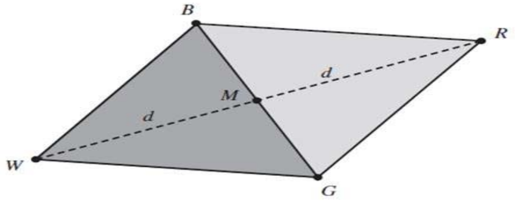

Fig. 3 Reflected point R for the Nelder-Mead method

The vector formula for R is

R=M+ (M-W) =2M-W (4)

4) *Expansion using the point E*

If function value R is lesser than function value of W, then the simplex has moved in the correct direction toward the minimum. There exists a possibility that the minimum is just a bit farther than the point R. Using this assumption we extend the line segment through M and R to the point E. This forms an expanded new triangle BGE in which the point E is found by moving an additional distance d along the line joining M and R. If the function value at R is greater than the function value at E, then we have found a better vertex than R. Then vector formula for E is

E=R+(R-M) =2R-M (5)

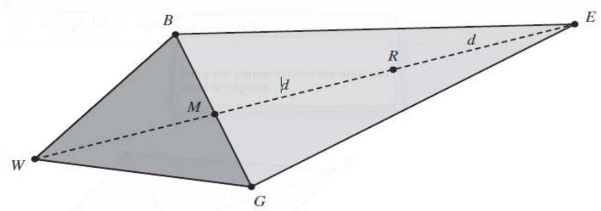

Fig. 4 Extended point E

5) *Contraction using the point C*

If the function values at R and W are the same, then another point must be tested. Perhaps M is having the smaller function, but we cannot replace W with M because we must have a triangle. Consider the two midpoints C1 and C2 of the line segments WM and MR, respectively. C is the point with the smaller function value and the new triangle is BGC. The choice between C1 and C2 may be inappropriate for the twodimensional case, but it is important in higher dimensions.

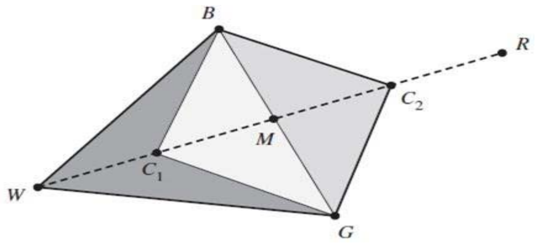

Fig. 5 The contraction point C1 and C2

6) *Shrink towards B*

I f the function value at w is not greater than the value at C, then the points G and W must be shrunk towards B. The point G and W is replaced with M and S respectively, which is the midpoint of the line segment joining B with W.

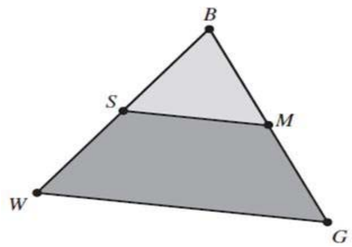

Fig. 6 Shrinking the triangle

7) *Logical Decision for Nelder Mead Algorithm*

```
IF $f(R) < f(G)$, THEN Perform
        Case (i) {either reflect or extend}
ELSE Perform
        Case (ii) {either contract or shrink}
    BEGIN {Case (i)}
        IF $f(B) < f(R)$ THEN
            Replace $W$ with $R$
        ELSE
            Compute $E$ and $f(E)$
                IF $f(E) < f(B)$ THEN
                    Replace $W$ with $E$
                ELSE
                    Replace $W$ with $R$
                ENDIF
        ENDIF
    END {Case (i)}

    BEGIN {Case (ii)}
        IF $f(R) < f(W)$ THEN
        Replace $W$ with $R$
        Compute $C= (W+M)/2$
        Or $C= (M+R)/2$ and
$f(C)$              IF $f(C) < f(W)$
THEN
            Replace $W$ with $C$
            ELSE
            Compute $S$ and $f(S)$
            Replace $W$ with $S$
            Replace $G$ with $M$
            ENDIF
    END {Case (ii)} ssss
```

B. *NM- PSO Method*

The NM-PSO optimization method integrates the constraint-handling methods, the Nelder-Mead simplex search method and the PSO algorithm. The PSO optimal method resists easily falling into the local best solution, but it requires many particles in an optimal process, which reduces the speed of computation. The Nelder-Mead simplex search method improves the efficiency of PSO due to its capacity for rapid convergence. However, the drawback of this method is that it easily falls into a local best solution. This drawback is improved by integrating the two algorithms. Combining the two algorithms and the gradient-based repair methods enables feasible optimal solutions to be found that satisfy the constraint conditions.

Using the advantages mentioned above, the NM-PSO method clearly overcomes the drawbacks of low convergence speed, the need for more particles, and the inability to deal with constraint conditions to accurately find optimal solutions.

The Pseudo code of the procedure

1. Initialization. Generate a population of size N (N > (n+1)).
   Repeat
2. Constraint handling method
   2.1 The Gradient Repair Method. Repair particles that violate the constraints by directing the infeasible solution toward the feasible region.
   2.2 Identify solutions that fulfill the constraint conditions and arrange them in the order of good to bad.
3. Nelder-Mead Method. Apply NM operator to the top n +1 particles and update the (n+ 1)th particle.
4. PSO Method. Apply PSO operator for updating the N particles.
   4.1 Selection. Select the global best particle and the neighbourhood best particle from the population.
   4.2 Velocity Update. Apply velocity updates to the N particles until the condition is fulfilled.

C. *Particle Swarm Optimization (PSO)*

PSO simulates the behaviours of bird flocking. Suppose the following scenarios a group of birds are randomly searching food in an area. There is only one piece of food in the area being searched. All the birds do not know where the food is. But they know how far the food is in each iteration [14] [8].

PSO is learned from the Scenario and used it to solve the optimization problems. In PSO each single solution is a “bird” in the search space. We call it “particle”. All of the particles have fitness value. 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 current optimum particles.

PSO is initialized with a group of random particles and the searches for optima by updating generations. In every iteration, each particle is updated by following two “best” values. The first one is the best solution (fitness) it has achieved so far. This value is called pbest. Another “best” value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When particle takes part of the population as its topological neighbours, the best value is a local best and is called lbest. After finding the two best values, the particle updates its velocity and positions with following equations [14].

V[]=V[]+C1*rand()*(Pbest[]Present[])+C2*rand()*(Gbest[]-Present[])
(6)
Present[]= Present[]+V[] (7)

Where V[] represent the particle velocity, present[] means current particle (Solution). Pbest[] and Gbest[] are defined as stated before, rand() means random number between (0,1).
C1,C2 are learning factors usually C1=C2=2.

The Pseudo code of the procedure

```
For each particle
    Initialize particle
END
Do
    For each particle
        Calculate fitness value.
        If the fitness value is better than the best fitness value (pbest) in history set current value as the new pbest.
    END
    Choose the particle with the best fitness value of all the particle as the gbest.
    For each particle
        Calculate particle velocity according equation (6)
        Update particle position according equation (7)
    END
```

III. EXPERIMENTAL RESULTS

In this section, the experiments that have been done to evaluate the performance of an NM, PSO and NM-PSO. NM method each and every iteration selects only one vertex from the search area and the best three coordinates are selected to form a new triangle. But for PSO, which contain with a group of random particles and searches for optima by updating generations. Here we are comparing NM-PSO with NM and PSO. Because of its global search ability and fast convergence speed compared with other global search algorithms, PSO is applied widespread in optimization. The drawback of PSO is that it easily falls into a local best solution. This drawback is improved by integrating the two algorithms.

A. Datasets

The Yeast Cell Cycle (YCC) dataset there are more than 6,000 genes during two cell cycles from yeast measured at 17times points. A subset of 698 genes is identified based on their peak times of five phases of the cycle and annotated. The resulting 692*72 data matrix is standardized (i.e., for each row the entries are scaled so that the mean is zero and the variance is one) and used for our experiment. Second one is Reduced Yeast Cell Cycle (RYCC). The data set originates in the one by Cho et al. Ka Yee Yeung extracted 384 genes from the yeast cell cycle data set in Cho et al. to obtain a 384*17 data expression matrix. It is to be pointed out that each gene in the RYCC data set appers also in the YCC data set. However, the dimensionality of the two data sets is quite different, and this may cause algorithms to behave differently. Third one is Reduced Peripheral Blood Monocytes (RPBM). We have randomly picked 10% of the cDNAs in each of the 18 original classes. Whenever that percentage is less than one, we have retained the entire class. The result is a 235*139 data matrix, and the true solution is readily obtained from that of PBM. The fourth dataset is Rat Central Nervous System (RCNS) which is a dataset obtained by reverse transcription coupled PCR to study the expression levels of 112 genes during rat central nervous system development over 9 time points. The result in a 112*9 data matrix Wen et al. studied it to obtain a division of the gene into 6 classes, in which 4 of them are composed of biologically functionally related genes. Such a division is assumed to be the true solution.

Figure 7, Figure 8, Figure 9, and Figure 10 correspondingly, show the results obtained from NM, PSO and NM-PSO for RPBM, RYCC, YCC and RCNS with varying cluster size 3 to 10. The results show that all the four datasets and varying cluster size the fitness value obtained from NM, PSO and NM-PSO. Here the NM-PSO performance is better than other two.

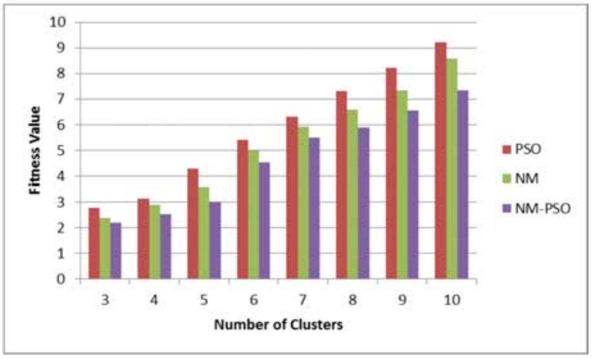

Fig. 7: Experiment results for RPBM data

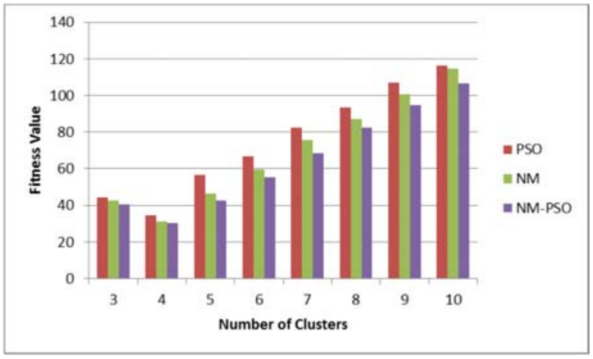

Fig. 8: Experiment results for RYCC data

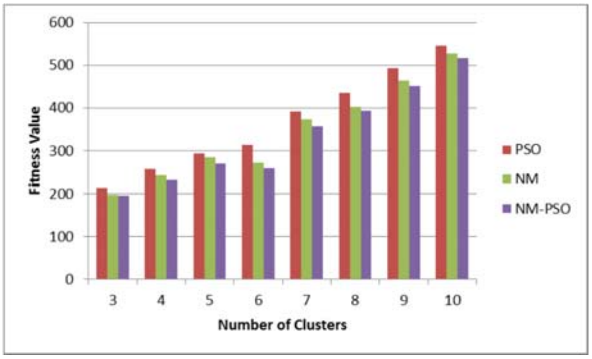

Fig. 9: Experiment results for YCC data

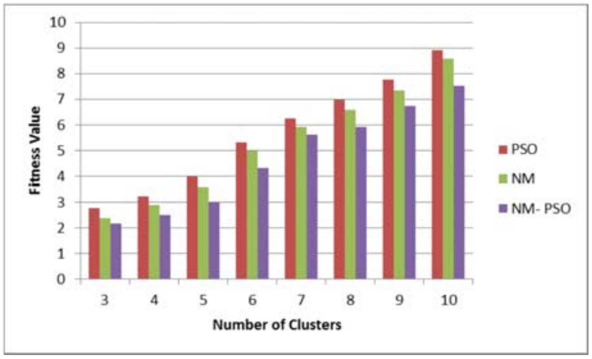

Fig. 10: Experiment result for RCNS data

IV. CONCLUSION

Microarrays are useful to simultaneously monitor the expression profiles of thousands of genes under various experimental conditions. Identification of gene cluster is the main goal in gene expression data analysis and is an important task in bioinformatics research. In this work the gene expression data are clustered using NM, PSO and NM-PSO. In this paper, the result shows that all the four datasets are varying cluster size with the fitness value obtained from NM, PSO and NM-PSO. NM-PSO performs better than NM and PSO. NM method each and every iteration selects only one vertex from the search area and the best three coordinates are selected to form a new triangle. The PSO optimal method resists easily falling into the local best solution, but it requires many particles in an optimal process, which reduce the speed of computation. The drawback of PSO is that it easily falls into a local best solution. This drawback is improved by integrating the two algorithms.

REFERENCES

[1] M. Pandi and K.Premalatha, “An Advanced Nelder Mead Simplex Method for clustering of Gene Expression data,” *International Journal of Computer, Information, Systems and Control Engineering*, vol. 8, no. 4, 2014.

[2] Alon, U. Barkai, N. Notterman, D. Gish, K. Y. S. Mack and D. Levine, “Broad Patterns of Gene Expression Revealed by Clustering Analysis of Tumor and Normal Colon Tissues Probe by Oligonucleotide Array,” *Proceedings of National Academy and Science*, vol. 96, no. 12, pp. 6745-6750, 1999.

[3] A. Liv and M.-T. Yang, “A New hybrid Nelder - Mead particle swarm optimization for coordination optimization of directional overcurrent relays,” *Mathematical problems in Engineering* , p. 18, 2012.

[4] J. Nelder and R.Mead, “A Simplex Method for function Minimization,” *Computer Journel*, pp. 308-313, 1965.

[5] Ben-Dor, A. Sharmir and Z. Yakhini, “Clustering Gene Expression Patterns,” *Journel of Computational Biology*, vol. 6, no. 3, pp. 281- 297, 1999.

[6] F. Gao and L.Han, “Implementing the Nelder- Mead Simplex Algorithm with Adaptive Parameters,” *Comput. Optim. Appl,* vol. 51, pp. 259-277, 2010.

[7] Fazel, F.Ganming and L. L. Ziying, “Evaluation and Optimization of clustering in gene expression data analysis,” *BMS Bioinformatics,* vol. 20, no. 10, pp. 1535-1545, 2004.

[8] J. Kennady and R. Eberhart, “Particle swrm optimization,” in *IEEE Internatinal Network on Neural Networks*, 1995.

[9] Kerr, G. Ruskin, H. Carne and D. P., “Techniques for clustering gene expression data,” *Computers in Biology and Medicine* , vol. 38, pp. 283-293, 2007.

[10] M. P. Chan, K. Yao and C. D.K.Y, “An Evalutionary clustering algorithm for gene expression microarray data analysis,” *IEEE Trans.Evolutionary Computations*, vol. 10, no. 3, pp. 296-314, 2006.

[11] N. Pham, A. Malinowski and T. Bartczak, “Comparative Study of Derivative Free Optimization Algorithm,” *IEEE Transaction on Industrial Informatics,* vol. 7, no. 4, 2011.

[12] R. Krovi, “Genetic Algorithms for clustering: A Preliminary Investigation,” in *In Proc. of the 25th Hawaii Int. Conference on System Sciences*, 1992.

[13] Wilamowski, N. Pham and B. M., “Improved Nelder- Mead Simplex method and applications,” *Journal of computing,* vol. 3, no. 3, pp. 55-63, 2011.

[14] Y. Li, X. Tian, L. Jiao and X. Zhang, “Biclustering of Gene Expression Data using Prticle swarm optimization integrated with pattern- driven local search,” in *IEEE Congress on Evolutionary Computation*, Beijing, China, 2014.

[15] Z. Du, YiweiWang and Z. Ji, “PK-means: A new algorithm for gene clustering,” *Computational Biology and Chemisty,* vol. 32, pp. 243-247, 2008.

[16] J. Thomas and G. Kulanthaivel, “Preterm birth prediction using Cuckoo search - based fuzzy min-max neural network,” vol. 8, no. 8, 2013.