Clustering Using Annealing Evolution : Application to Pixel Classifcation of Satellite Images

Ujjwal Maulik
Dept. of Comp. Sc. and Tech.
Kalyani Govt. Engg. College
Kalyani - 741235, India
ujjwal_maulik@kucse.wb.nic.in

Sanghamitra Bandyopadhyay
Machine Intelligence Unit
Indian Statistical Institute
Kolkata 700108, India
sanghami@isical.ac.in

Malay K. Pakhira
Dept. of Comp. Sc. and Tech.
Kalyani Govt. Engg. College
Kalyani - 741235, India
mkp@kucse.wb.nic.in

Abstract

In this article an efficient clustering technique, that utilizes an effective integration of simulated annealing and evolutionary programming as the underlying search tool, is developed. During the evolution process, data points are redistributed among the clusters probabilistically so that points that are farther away from the cluster center have higher probabilities of migrating to other clusters than those which are closer to it. The superiority of the new clustering algorithm over the widely used K-means algorithm and those based on simulated annealing and conventional evolutionary programming is demonstrated for some real life data sets. Another real life application of the developed clustering technique in classifying the pixels of a satellite image of a part of the city of Mumbai is also provided.

1. Introduction

Clustering [1, 15, 8, 11, 2] is an exploratory data analysis tool where a set of patterns, usually vectors in a multi-dimensional space, are grouped into clusters in such a way that patterns in the same cluster are similar in some sense and patterns in different clusters are dissimilar in the same sense. Different distance criteria, e.g., Euclidean, Mahalanobis etc. [15], are often used for defining the similarity measures. One of the widely used and intuitively simple clustering technique is the K-means algorithm [15], which optimizes the sum of the intra cluster distances corresponding to the K clusters. However, the major drawback of the K-Means algorithm is that it often gets stuck at local minima and the result is largely dependent on the choice of the initial clustering of the data set [14].

In order to overcome the above mentioned limitation, in this article, an efficient clustering technique, that utilizes an effective integration of simulated annealing [10, 3] and evolutionary programming [6, 7] as the underlying search and optimization tool, is developed. The resulting clustering technique is called the ANEV-clustering. During the evolution process, data points are redistributed among the clusters probabilistically so that points that are farther away from the cluster center have higher probabilities of migrating to other clusters than those which are closer to it. Experimental results comparing the performance of the proposed clustering technique with those of the K-Means algorithm, SA and the conventional evolutionary programming approaches are provided for several real-life data sets. Moreover the utility of the technique for classifying pixels in a satellite image of a part of the city of Mumbai for differentiating different landcover regions is demonstrated.

2. Clustering Using Annealing Evolution

The searching capabilities of simulated annealing and evolutionary programing have been used in this article for the purpose of appropriately determining a fixed number K of cluster centers in $\mathbb{R}^N$; thereby suitably clustering the set of n unlabelled points. The clustering metric that has been adopted is the sum of the Euclidian distances of the points from their respective cluster centers.

2.1. Basic Principle

Like the K-means algorithm, the annealing-evolution process tries to minimise the intra cluster spread (ICS). Mathematically, ICS is defined for the K clusters $C_1, C_2, \ldots, C_k$ as follows:

$$ICS(C_1, C_2, ..., C_k) = \sum_{i=1}^{K} \sum_{x_j \in C_i} ||x_j - z_i||^2 \qquad (1)$$

Here z_i represents the center of cluster C_i and $x_j \in C_i$ represents a data point x_j belonging to cluster C_i. The goal is to search for the appropriate cluster centers $z_1, z_2, \ldots, z_K$ such that the clustering criterion ICS is minimised.

A configuration in ANEV-clustering corresponds to a set of K cluster centers, where K, the number of clusters, is assumed to be known *a priori*. The data points are initially assigned to K clusters (where K is known *a priori*) randomly, and the cluster centers are computed using Eqn. 2.

$$z_i = \frac{1}{n_i}\Sigma_{x_j \in C_i} x_j \quad i = 1, 2, \ldots, K. \tag{2}$$

This is analogous to the way the centers are computed in the K-means algorithm. The set $\{z_1, z_2, \ldots, z_K\}$ constitute a configuration $\mathcal{C}$ of ANEV-clustering. Once the cluster centers are computed, the ICS is calculated as in Eqn. 1. This is considered to represent the energy of the above configuration $\mathcal{C}$.

Each element in a cluster is assumed to have a certain degree of membership to it which is inversely proportional to its Euclidian distance from the cluster center. So elements that are at a larger distance from the center are the fittest candidates for redistribution to another cluster. We have redistributed an element x_i in cluster C_j to cluster C_k with probability

$$\exp(\frac{-[D_{ik} - D_{ij}]+}{T_g}) \tag{3}$$

where $[x]+ = \max(x, 0)$, and $D_{ik} = ||x_i - \mathbf{z}_k||$ and $k \neq j$. T is a temperature schedule, which is a sequence of strictly positive numbers such that $T_1 \geq T_2 \geq \ldots \geq T_g = 0$. The suffix g of T indicates the number of generations through the annealing process. Note that $T_g \rightarrow 0$ as $g \rightarrow \infty$.

2.2. The Algorithm

In the proposed clustering algorithm, a pool, P, of probable solutions or configurations is maintained. Each member of the pool (also called population) is represented as a string of K vectors, each identifying a cluster center. For each member of the pool, the data points are initially assigned to K clusters, known *a priori*, randomly, and the cluster centers are computed as in Eqn. 2. Thus we get a number of clustering solutions equal to the population size ($POPSIZE$). For each configuration (solution string), we compute its energy value as the corresponding ICS. The fitness value of the string is correspondingly calculated as $1/ICS$. The solutions which are potentially better, having lower energy values, are selected with higher probability and used in the mutation phase. At each temperature, T, a new pool is generated from the older one by the processes of selection and mutation. The annealing-evolution algorithm is shown below.

The Annealing-Evolution Clustering Algorithm

```
begin
    $g = 0$
    initialise $(T_g = T_0, P(g))$
    evaluate $P(g)$
    $termination\_condition = false$
    while (NOT $termination\_condition$) do
        begin
            $g = g + 1$
            select $P(g)$ from $P(g-1)$
            for $i$ = 1 to POPSIZE do
            begin
                evolve configuration $\mathcal{C}_i^{'}$ with energy $E^{'}$
                mutating $\mathcal{C}_i$ with energy $E$
                if $(E^{'} - E \leq 0)$ then
                        begin
                            $C_i \leftarrow C_i^{'}$
                            flag = true
                        end
                else if exp $\left[-\frac{E^{'}-E}{T_g}\right] \geq$ random [0,1) then
                        begin
                            $C_i \leftarrow C_i^{'}$
                            *flag* = true
                        end
            endfor
                if *flag* then
                    begin
                        replace old $C_i$ in $P(g)$ by new $C_i$
                        *flag* = false
                    end
            $T_g = T_0 * (1 - \alpha)^g$ /*$0 < \alpha < 1$, or, lower $T_g$ */
        end
end
```

2.3. *Selection*

$Selection$ is the process where the strings, or, chromosomes, with relatively higher fitness are reproduced to form a pool for the next generation. A number of commonly used strategies are there. Two of them are the $roulette - wheel$ selection scheme (also called the proportionate selection scheme) and the $tournament$ selection scheme. In both the cases, the strings are selected with probabilities proportional to their fitness values. We have used the tournament selection scheme with a tournament size of 10. Here, for selection of a single string, a set of 10 randomly identified string is picked up from $P(g)$, of which the best string is placed in $P(g+1)$.

In addition, the proposed annealing-evolution algorithm uses the *elitist strategy*, which ensures that the best solution string of $P(g)$ survives to $P(g+1)$.

2.4. *Mutation*

Here, we try the possibilities to generate new solution strings from the existing ones. A solution string represents one cluster configuration, encoding a set of cluster centers. Corresponding to this solution string a partitioning of the data exists, and each element of a cluster is assigned a certain degree of membership to it which is inversely proportional to its Euclidian distance from the cluster center. Therefore, elements that are at a larger distance from the center are the fittest candidates for redistribution to another cluster. We have redistributed an element x_i in cluster C_j to cluster C_k with probability given by equation 3. In this way, a new cluster configuration $C^{'}$ with energy $E^{'}$ is generated from an existing configuration C with energy E. Thus at each generation a number of alternative solution strings are generated which are accepted or rejected according to probability exp $\left[-\frac{E^{'}-E}{T_g}\right]$ as shown in the algorithm. Mutation is performed stochastically, with probability P_m.

For the purpose of comparison, we have also implemented two versions of evolutionary programming (EP) based clustering. As earlier, the chromosomes encode a set of cluster centers, and the ICS is used as a measure of the fitness values. In the mutation phase the centers are perturbed by adding small quantities of noise generated using some probability distribution. Here, we have used Gaussian and Cauchy distributions for this purpose. These mutation schemes [16] are mentioned briefly below.

Gauss and Cauchy mutation strategies

Let $\{x_1, x_2, \cdots, x_k, \cdots, x_m\}$ be a chromosome, and the element x_k be selected for mutation. The result will be a vector $\{x_1, x_2, \cdots, x_k{}^{'}, \cdots, x_m\}$

In classical evolutionary programming Gaussian mutation is commonly used. Gaussian mutation requires two parameters: a mean value and a standard deviation. Generally, mutation in this approach was achieved by transforming the component x_i of a vector X into $x_i^{'}$ by the following relation:

$$x_i^{'} = x_i + \eta_i(j) N_j(0,1) \tag{4}$$

$$\eta_i^{'}(j) = \eta_i(j) \exp\left(\tau^{'} N(0,1) + \tau N_j(0,1)\right) \tag{5}$$

where η_i's are standard deviations for Gaussian mutations. $N(0,1)$ denotes a normally distributed one-dimensional random number with mean zero and standard deviation one. $N_j(0,1)$ indicates that the random number is generated anew for each value of j, where j varies over the number of dimensions. The factors τ and $\tau^{'}$ are commonly set to $\left(\sqrt{2\sqrt{n}}\right)^{-1}$ and $\left(\sqrt{2n}\right)^{-1}$ respectively, where n represents the length of a chromosome.

In [16] it is shown that the Cauchy mutation function performs better than the Gaussian mutation function for most of the benchmark problems used for EPs. Cauchy mutation can perform longer jumps with high probability and hence its search step size is considerably larger than that of the Gaussian.

The Cauchy distribution function is given by $F_t(x) = \frac{1}{2} + \frac{1}{\pi} arctan\left(\frac{x}{t}\right)$, where $t > 0$ is a scale parameter.

The Cauchy mutation follows the following relation

$$x_i^{'} = x_i + \eta_i(j)\delta_j \tag{6}$$

where, $\delta_j = t \tan\left[\pi\left(F_t - \frac{1}{2}\right)\right]$ Generally t is taken to be 1. δ_j is computed for each j anew. The relation for changing η remains same as in the case of Gaussian mutation and is given by Equation 5.

2.5. *The Temperature Schedule*

The asymptotic convergence (i.e., at $t \to \infty$) of the SA is guaranteed for a logarithmic annealing schedule of the form $T_t = T_0/(1 + \ln t)$, where $t \geq 1$ and T_0 is the initial temperature. However, in practice, the logarithmic annealing is far too slow and hence we have used a geometric schedule of the form $T_t = (1-\alpha)^t * T_0$, where α is a positive real number close to zero. As $T_t \to 0$, no more perturbation of the cluster configuration is possible and hence termination condition is assumed to be reached. In practice, the state of the system configuration is found to be frozen well before this.

3. Implementation Results

The experimental results comparing the ANEV-clustering algorithm with the K-means algorithm, SA based technique, and two EP based schemes (with Cauchy and Gaussian mutations) are provided for three real-life data sets (*Vowel, Iris* and *Crude Oil*) respectively. These are first described in this section. Subsequently, the effectiveness of the ANEV-clustering technique for classifying pixels is demonstrated for a satellite image of a part of the city of Mumbai.

3.1. Comparative Performance of ANEV-clustering

The three real-life data sets are described below.

Vowel Data: This data consists of 871 Indian Telugu vowel sounds [13]. These were uttered in a consonant-vowel-consonant context by three male speakers in the age group of 30-35 years. The data set has three features F_1, F_2 and F_3, corresponding to the first, second and third vowel format frequencies, and six overlapping classes $\{\delta, a, i, u, e, o\}$. The value of K is therefore chosen to be 6 for this data. Fig. 1 shows the distribution of the six classes in the $F_1 - F_2$ plane.

Iris Data: This data represents different categories of irises having four feature values. The four feature values represent the sepal length, sepal width, petal length and the petal width in centimeters [5]. It has three classes (with some overlap between classes 2 and 3) with 50 samples per class. The value of K is therefore chosen to be 3 for this data.

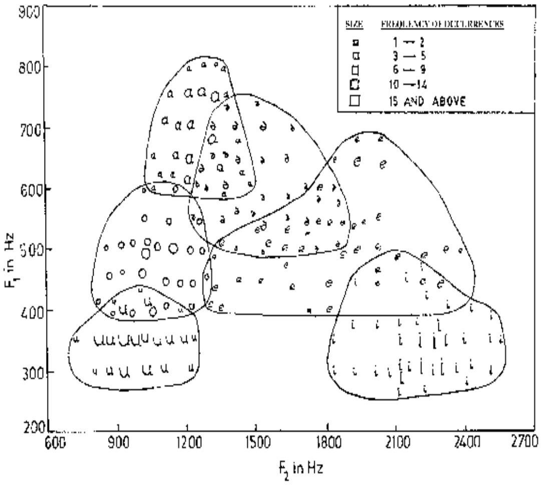

Figure 1: *Vowel* data in the $F_1 - F_2$ plane

Crude Oil Data: This overlapping data [9] has 56 data points, 5 features and 3 classes. Hence the value of K is chosen to be 3 for this data set.

The clsutering algorithms K-means algorithm, conventional EPs, and the annealing-evolution-clustering algorithm are executed for a maximum of 20 generations. The SA based method was however executed for 200 generations in order to obtain comparable performance. The values of α and T_1 are chosen to be 0.05 and 50 respectively.

Tables 1-3 show the comparative results for *Vowel, Iris* and *Crude Oil* respectively. Although all the algorithms were run for 100 simulations, for the purpose of demonstration, results are provided for only five different initial configurations.

For *Vowel* (Table 1), the K-means algorithm attains the best value of 149912.56 only once. The best values obtained by EP-Gauss and EP-Cauchy are found to be 149525.90 and 149532.73 respectively, while that for the SA based method is 149409.25. The best value obtained by ANEV-clustering algorithm is 149333.64 (which is obtained in 40% of the total runs). Notably, the latter always obtains values of ICS that are better than the best obtained by the former methods. For *Iris* (Table 2) and *Crude Oil* (Table 3) data sets, the proposed clustering algorithm attains the best values of 97.1007 and 278.9650 respectively. The K-means algorithm, on the other hand, fails to attain these value in any of its runs. The best that K-means algorithm achieved are 97.2248 and 279.7432 respectively. In case of EPs, we see that the best value obtained for *Iris* is 97.1108. The SA based method performs poorly for this data, attaining a best value of 97.3527. For *Crude_Oil*, once again the K-Means and SA based method perform poorly, attaining best values of 279.74232 and 279.4581 respectively. For the EP based methods, these values are 278.9651 for both the cases.

Table 1: ICS obtained by K-means, SA, EP and ANEV-clustering algorithm for five different runs (R) for *Vowel* when K=6

R	K-means	SA	EP-Gauss	EP-Cauchy	ANEV
1	149912.56	149409.25	149892.82	150485.17	149356.54
2	150469.73	149409.31	149525.90	149532.73	149333.64
3	149931.56	149429.39	150320.34	149887.21	149369.39
4	149912.64	149443.48	151538.95	150530.25	149386.35
5	149942.68	149409.31	154008.32	154653.34	149347.39

Table 2: ICS obtained by K-means, SA, EP and ANEV-clustering algorithm for five different runs (R) for *Iris* when K=3

R	K-means	SA	EP-Gauss	EP-Cauchy	ANEV
1	97.2248	97.4537	97.1108	97.2045	97.1007
2	97.2248	97.5717	97.2045	97.1108	97.1007
3	97.2248	97.5717	97.1108	97.2045	97.1007
4	97.2248	97.4537	97.2232	97.1108	97.1007
5	97.2248	97.3527	97.1108	97.2045	97.1007

Table 3: ICS obtained by K-means, SA, EP and ANEV-clustering algorithm for five different runs (R) for *Crude Oil* when K=3

R	K-means	SA	EP-Gauss	EP-Cauchy	ANEV
1	279.7432	279.4581	279.2709	278.9651	278.9650
2	279.7432	279.4581	279.2709	278.9651	278.9650
3	279.7432	279.4581	278.9651	278.9651	278.9650
4	279.7432	279.4581	279.7432	278.9651	278.9650
5	279.7432	279.4581	279.2709	279.1068	278.9650

3.2. Pixel Classification of *IRS* Image of Mumbai

In this section, the utility of the ANEV-clustering technique for partitioning different landcover regions of a 512× 512 satellite image of a part of the city of Mumbai is demonstrated. Two bands of the image are considered: green band of wavelength 0.52 - 0.59 μm, and near infra red band of wavelength 0.77 - 0.86 μm. Note that satellite images usually have a large number of overlapping classes and hence the clustering problem in such space becomes quite difficult. This data has been used earlier for classifying the pixels into different categories under the supervised framework [4, 12].

Some important landcovers of Mumbai, as seen more prominently from near infra red band (Fig. 2 shows the image with histogram equalization to make it more prominent), are as follows : The elongated city area is surrounded by the Arabian sea. There is a concrete structure (on the right side top corner) connecting Mumbai to New Mumbai. On the southern part of the city, there are several islands, including the well known *Elephanta islands*. The dockyard is situated on the south eastern part of Mumbai, which can be seen as a set of three finger like structure. On the upper part of the image, towards left, there is a distinct criscrossed structure. This is the *Santa Cruz airport*.

Fig. 3 provides the classified Mumbai image obtained using ANEV-clustering algorithm. K was taken to be 6 for this image data. As can be seen from the results, most of the landcover categories have been correctly distinguished. For example, the *Santa Cruz* airport, dockyard, bridge connecting Mumbai to New Mumbai, and many other road structures have come out distinctly. Moreover, the predominance of one category of pixels in the southern part of the image conforms to the ground truth; this part is known to be heavily industrialized, and hence the majority of the pixels in this region should belong to the same class of concrete. The *Arabian sea* has come out as a combination of pixels of two different classes. This is again in conformity with earlier experiments with this data [12], where the sea water was found to be decomposed into two classes, turbid water 1 and turbid water 2, based on the difference of their reflectance properties. (Note that the result presented here does not include any postprocessing operations. One such step that is usually carried out in image processing is the merging of very small segments to one of the neighboring regions. This helps in cleaning up the image by filling up small holes embedded with large regions.) Therefore, from the ground truth available and the performance of ANEV-clustering, it can be concluded that it is a useful technique for differentiating the various landcover types present in an image.

4. Conclusions and Discussion

An integration of simulated annealing and evolutionary programming, thereby combining the merits of both these approaches, has been used in this article for developing an efficient clustering technique. The performance of the proposed technique is compared with the K-means algorithm, as well as schemes based on simulated annealing and conventional evolutionary programming, with both Cauchy and Gaussian mutation, for three real-life data sets. It is found that the proposed method outperforms the other four techniques for all the data sets. Another real-life application of the said clustering method is demonstrated for classifying different land cover regions in remote sensing imagery. As a scope of further work, a detailed study regarding the convergence of K-means, SA, EP and ANEV-clustering algorithms needs to be done. Moreover, the proposed clustering algorithm may be extended to the case of fuzzy partitioning of a data.

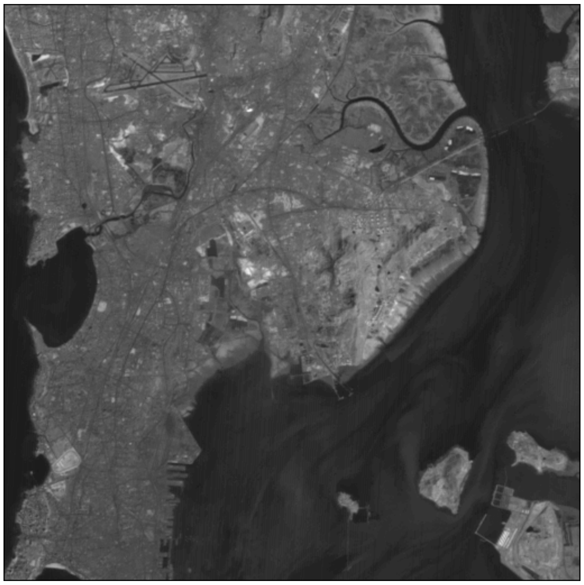

Figure 2: Near infra red band of the Mumbai image with histogram equalization

Acknowledgement : The research is partially supported by a project titled *Region Classification in Remote Sensing and Satellite Images: Soft Computing Approach*, No. 8019/RDII/TAP/COM(239)/2000-01, funded by All India Council for Technical Education (AICTE), Govt. of India, 2001-2003.

References

[1] M. R. Anderberg. *Cluster Analysis for Application.* Academic Press, 1973.

[2] S. Bandyopadhyay and U. Maulik. Non-parametric genetic clustering : Comparison of validity indices. *IEEE Transactions on Systems, Man and Cybernetics Part-C*, 31(1):120–125, 2001.

[3] S. Bandyopadhyay, U. Maulik, and M. K. Pakhira. Clustering using simulated annealing with probabilistic redistribution. *Intl. J. Pattern Recognition and Artificial Intelligence*, 15(2):269–285, 2001.

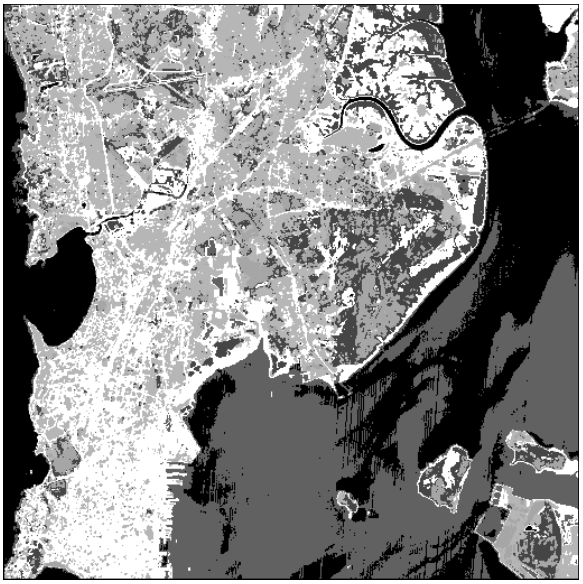

Figure 3: Mumbai image partitioned using the ANEV-clustering method

[4] S. Bandyopadhyay, C. A. Murthy, and S. K. Pal. Supervised pattern classification by surface fitting using genetic algorithms. *Proc. INSA*, 67,A(2):295–314, 2001.

[5] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 3:179–188, 1936.

[6] L. Fogel. Autonomous automata. *Industrial Research*, 4:14–19, 1962.

[7] L. Fogel, A. Owens, and M. Walsh. *Artificial Intelligence through Simulated Evolution.* John Wiley & Sons, New York, 1966.

[8] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, Englewood Cliffs, NJ, 1988.

[9] R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis.* Prentice-Hall, 1982.

[10] S. Kirkpatrik, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.

[11] U. Maulik and S. Bandyopadhyay. Genetic algorithm based clustering technique. *Pattern Recog.*, 33:1455–1465, 2000.

[12] S. K. Pal, S. Bandyopadhyay, and C. A. Murthy. Genetic classifiers for remotely sensed images : Comparison with standard methods. *Intl. J. Remote Sensing*, 22(13):2545–2569, 2001.

[13] S. K. Pal and D. D. Majumder. Fuzzy sets and decision making approaches in vowel and speaker recognition. *IEEE Trans. Syst., Man, Cybern.*, SMC-7:625–629, 1977.

[14] S. Z. Selim and M. A. Ismail. K-means type algorithms : A generalized convergence theorem and characterization of local optimality. *IEEE Trans. Pattern Anal. Mach. Intell.*, 6:81–87, 1984.

[15] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles.* Addison-Wesley, Reading, 1974.

[16] X. Yao, Y. Liu, and G. Lin. Evolutionary programming made faster. *IEEE Transaction on Evolutionary Computation*, 3(2):82–102, 1999.