Hybrid Mode Neuro-genetic Networks to Understand Genetic Features

Tanya Titova[1], Veselin Nachev[2], Chavdar Damyanov[3] and Nanko Bozukov[4]
[1]Tanya Titova, Department Automation and Control Systems
University of Food Technologies, 26 Maritza Blvd, Plovdiv
4002,
Bulgaria
t_titova@abv.bg

[2]Veselin Nachev, Department Automation and Control Systems
University of Food Technologies, 26 Maritza Blvd
Plovdiv, 4002, Bulgaria
v_nachevbg@yahoo.com

[3]Chavdar Damyanov, Department Automation and Control
Systems University of Food Technologies, 26 Maritza Blvd
Plovdiv, 4002 Bulgaria
chavdam@yahoo.com

[4]Nanko Bozukov, Department "Informatics and Statistics" at
University of Food Technologies - Plovdiv, 26 Maritsa Blvd
Plovdiv 4002, Bulgaria
bozukovnanko@abv.bg

ABSTRACT: *Neural networks have large applications and have the characteristics of many classes. The hybrid mode neuro-genetic networks are one such division of neural networks. They have adaptive optimization with the character of natural selection and genetical features. The automated classifiers performance can be enhanced to record efficiency in the systems for quality determination using the hybrid structure sorting. We have discussed these issues in this paper.*

Keywords: Artificial Neural Network, Genetic Algorithm, Neuro-Genetic Algorithm, Food Quality

Received: 18 October 2022, Revised 15 December 2022, Accepted 29 December 2022

DOI: 10.6025/jic/2023/14/1/1-9

1. Introduction

Recent years have witnessed an increasing proliferation of information processing technologies based on the combination of methods belonging to different spheres of science and technology. There are many examples of experimental research which successfully integrate neural networks with methods pertaining to wavelet transform theory, genetic programming, fuzzy logic, etc [6,7].

The paper concentrates on the problems related to the improvement of neural network efficiency through hybrid structures featuring genetic algorithms.

2. Neural Networks

Artificial neural networks (ANN) have been thoroughly studied and tested for various applications. Nevertheless, there arise some problems that remain to be solved, which is challenging to ongoing research. One of these problems has to do with the determination of suitable network dimensionality and topology [5,8].

The design of a neural network (NN) is a complex task due to the fact that in most cases it is very difficult to find an optimal dimensionality (Fig. 1).

The methods for determining network architecture are subdivided into statistical and dynamic. Dynamic methods are commonly used since they can readily supply the network with an optimum dimensionality. These methods resort to constructive ("growing") or destructive ("pruning") techniques or combinations.

According to the constructive method, a small neural network is set up and more neurons and connections are subsequently added until a balance is reached. The major drawback here is that the neural network in its final form is more complex than desired or needed.

The destructive method starts with a fully connected neural network (weights and neurons). Afterwards, the network is gradually reduced via a specific algorithm until the proper network structure is obtained. The basic problem lies in the complexity of the initial network and connection weight setup as well as in the assessment of their relative impact on the target function. It is also very difficult to find an evaluation function and network reduction rules on the basis of which the algorithm will be able to generate a sufficiently small network [2,6].

Evolutionary artificial neural networks (EANN) are being increasingly studied and applied. They are a subtype of ANNs

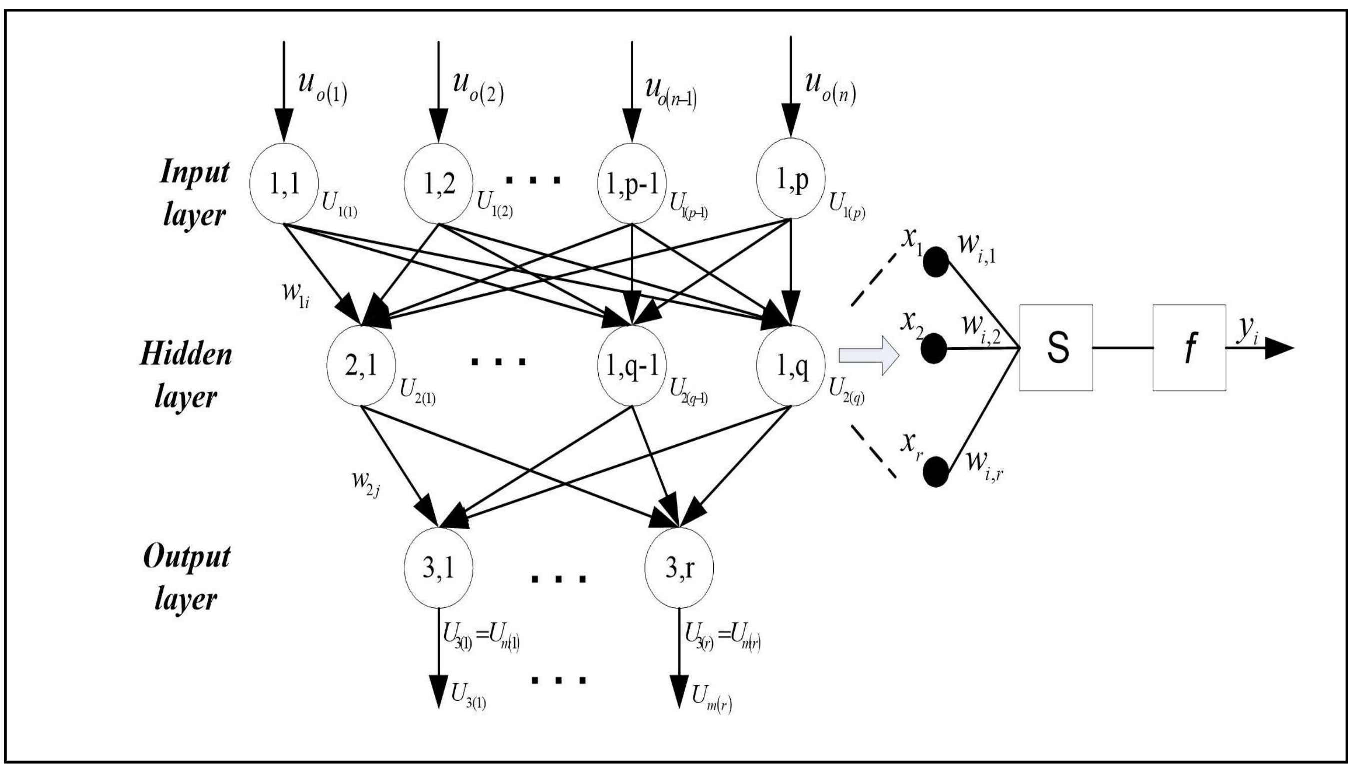


Figure 1. Typical Structure of Neural Networks

constituting hybrid structures which use an evolutionary strategy. Evolutionary algorithms can be applied to the optimization of neural systems so that various problems can be solved, such as connection weight determination, training, architecture synthesis, feature selection, connection with initialization weights, solution that ends the ANN training, etc. EANNs may also be adapted to an environment with recurring changes in the conditions, such as composition, sample size, number of classes, etc.

In essence, the problem is defined thus: if there are Hnumber neurons in the hidden layer, then there are $O\left(2^{H^2}\right)$ network typologies. With a calculation complexity of this kind, it is expedient to look for a suboptimal solution using efficient heuristic algorithms.

Genetic algorithms (GA) are widely applied as such. They are a combination of heuristic rules and stochastic search algorithms, which some authors believe to be the only possible means of finding an optimal solution.

3. Genetic Algorithms

Genetic algorithms (GA) are based on a stochastic method for global search and optimization which imitates the evolution of living organisms delineated by Charles Darwin in his “On the Origin of Species by means of Natural Selection”. For the most part, these algorithms feature operators which select the individuals (solutions) to reproduce themselves and generate new individuals on the basis of those that have already been selected and determine the composition of the population for the next generation. Crossover and mutation operators are the most important ones participating in this procedure. GAs belong to the group of evolutionary algorithms.

Evolutionary algorithms make use of the three basic principles of natural evolution as described by Darwin: natural selection, variety of individuals and reproduction, maintained via the differences between prior and subsequent generations. In the 1960s, these three characteristics of natural evolution inspired a number of researchers to come up with independent stochastic search methods:

• Evolutionary programming

• Evolution strategies

• Genetic algorithms.

All three methods work with a range of individuals. The selection principle is applied by means of a criterion evaluating the proximity between the individual and the solution desired. The most adaptable individuals proceed to the next generation [5].

The huge variety of problems, not only engineering ones but also problems belonging to other spheres of knowledge, necessitates the application of algorithms of a different type, with different characteristics and configuration. This necessity is responsible for the multitude of different evolutionary algorithms and the great number of researchers working on them.

The GA principle holds as follows:

1. An initial set of combinations (populations) is randomly selected $I_0 = \{i_1, i_2, .. i_s\}$ and it is postulated that max $f^* = \max(f(i) \mid i \in I_0), \quad k := 0$

3.1.1. Until the ending criterion (specific time or number of generations) is fulfilled, the following is carried out:

3.1. Selection of parents i_1 and i_2 from the population I_k - selection operator. Proportionate selection (roulette wheel selection) is used. The probability for a k -step solution i being a parent is:

$$P(i - selected) = f(i) \Big/ \sum_{i=1}^{I_k} f(i), \; i \in I_k$$

3.1.2. Constructing $i^{|}$ on the basis of i_1 and i_2 - crossing-over operator. The operator is performed with a P_c probability and for a random point of separation (Figure 2). If the operator is not realized (with a 1- P_c probability), the offspring takes after its parents.

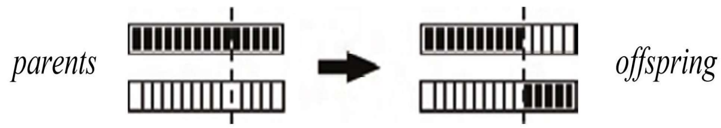


Figure 2. Tentative diagram of a crossing-over operator

3.1.3. Modification of $i^{|}$ - mutation operator.
The operator is realized with a P_m user-defined probability.

3.1.4. If $f^* < f(i^{|})$, then $f^* := f(i^{|})$.

3.1.5. Population renewal ($k := k+1$).

4. Neuro-genetic Algorithms

Neuro-genetic algorithms make use of the combination of an artificial neural network (ANN) and genetic algorithms (GA) to optimize the NN parameters. When the parameter values of a NN are initially entered, they are not usually optimal. That is why GAs are used to determine the weights of the network, suitable training parameters or reduce the size of the training sample by opting for the most significant features and network structure. The network structure to a great extent determines the network efficiency and the problems which it will be able to solve. We know that in order to determine non-linearly separable classes, the network must have at least one layer between inputs and outputs but the determination of the number and size of the hidden layers is commonly defined on the basis of expertise. The number of the input variables accounts for the dimensionality of the neural network. To avoid the problems with its re-training and improve its work, the number of the input variables should be smaller. A GA can optimize the inputs by generating proper network architecture for a specified data range. The preliminary processing of the data is also essential for discovering information and eliminating improper or undesirable features. Many researchers are highly interested in the transformation and selection of a subset of features. Hence, it is vital to optimize the structure of ANNs [1, 3, 4].

The flowchart of the hybrid neuro-genetic approach is shown in Figure 3. It has been applied to potato classification.

The results demonstrate that this approach is likely to enhance precision compared to the results obtained from classification using only a NN and backpropagation.

The hybrid neuro-genetic algorithm holds as follows:

1. The features are determined with the help of experts.

2. Initialization of count=0, fitness=0, number of cycles

3. Generation of an initial population. The individual's chromosome is formulated as a sequence of genes, each of which is encrypted at the input.

4. Design of a proper network (input, hidden and output layers)

5. Determination of the weight for each connection.

6. NN training with backpropagation.

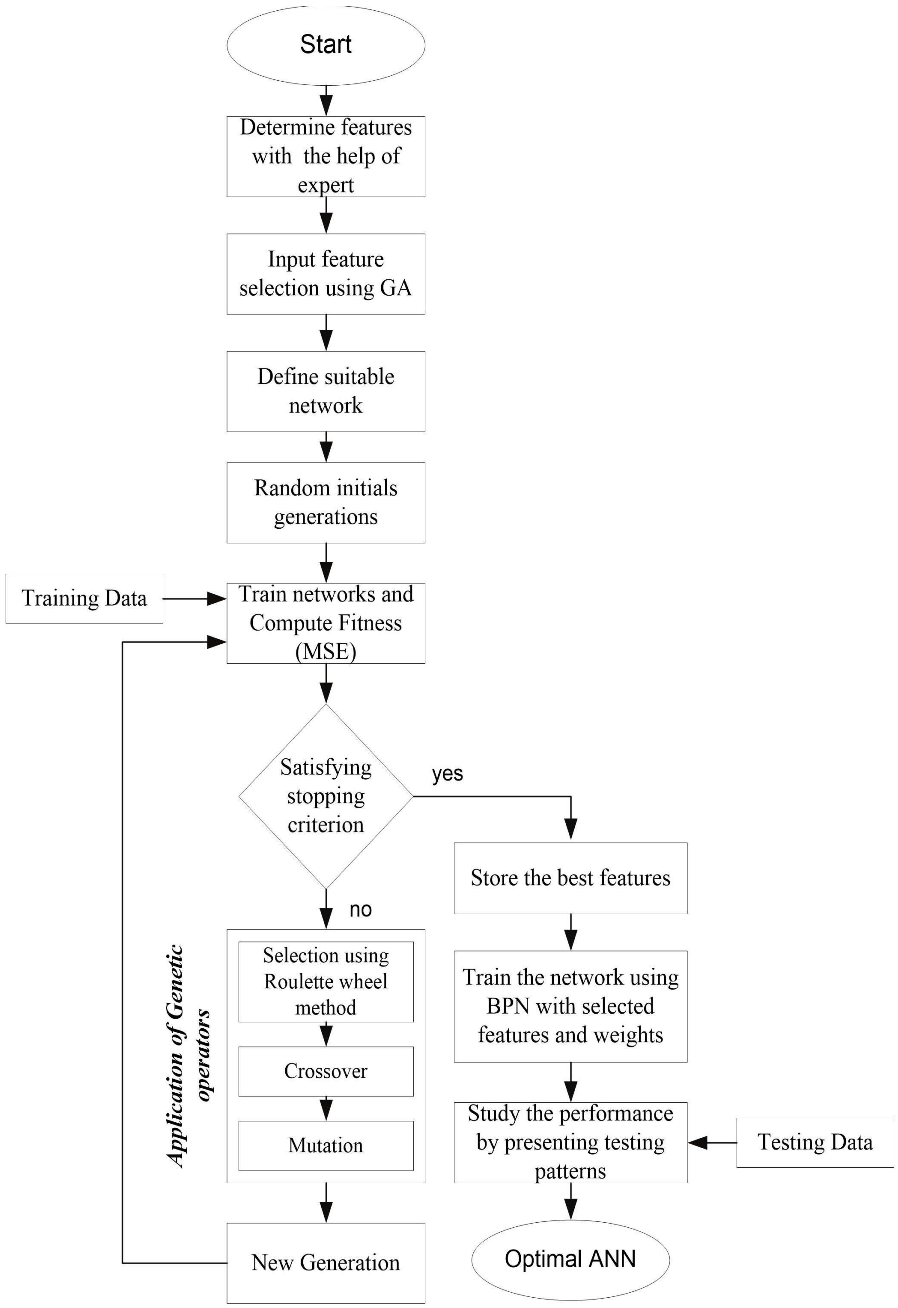


Figure 3. Hybrid Neuro-Genetic Algorithm

7. Finding the accumulated error and the value of the fitness function. Genotypes are evaluated on the basis of a fitness function.

8. If the preceding fitness function is smaller than the current fitness value, the current function is kept.

9. count = count +1

10. The selection: the two parents are selected by means of the roulette wheel principle, i.e. randomly.

11. The genetic operations of crossover, mutation and reproduction generate new characteristics (and new weights are applied for each connection).

12. If (the number of cycles <= count), then point 4.

13. The NN is trained with the features selected. 14. The NN is tested with the testing sample.

5. Experiments And Results

5.1. Materials

Potatoes are classified on the basis of the presence and degree of damage of the tubers (internal and external defects, diseases, genetic anomaly, injuries, etc.). The expert classifies each tuber into one of the following classes - D_1 (1st quality), D_2 (2nd quality) и D_3 (3rd quality). Each realization $U[n]$ of the light transmission coefficient consists of L=25 values, registered by the longitudinal scanning of the tuber. The $U[n]$ function possesses almost complete spectrophotometric information about the product and is invariant to disturbances and product thickness.

The reference evaluation and classification of the potato tubers was carried out with the help of the methodology outlined in [2]. It is worth pointing out that the recognition algorithm does not diagnose the disease of the infected potato but identifies only the presence of a defect in the tissue. As a result of scanning, signals are received which are proportional to the optical densities of transmission OD_T and reflection OD_R. The signals have different shapes depending on the respective wavelength, the shape and size of the tubers, the presence and location of the defects.

5.2. Experimental

In a MATLAB environment, various structures of neural networks were tested experimentally. As shown in Figure 3, for the realization a feature selection algorithm was used.

Therefore, the feature vector is represented by a feature bit mask fig. where N refers to the initial set of features while the possible solutions in this case amount to 2^N -1 .

FEATURES	S1	S2	S3	S4	…	SN
FEATURE BIT MASK (CHROMOSOMES)	0	1	1	0	…	1

Table 1. Feature Vector Encoding

The other basic elements of the algorithm are realized via training modules which are presented in the most generalized fashion by the following pseudocode:

Learning program fragment:

```
% Create a neural network
net = feedforwardnet(n);
net.layers{1}.transferFcn = 'tansig';
net.layers{2}.transferFcn = 'logsig';
% Configure the neural network for this dataset
net = configure(net, inputs, targets); view(net)
% Create handle to the MSE_TEST function, that
calculates MSE
h = @(x) mse_test(x, net, inputs, targets,B,instances)
```

```
% Set of Parameters of Genetic Algorithms
ga_opts = gaoptimset('TolFun', 1e-
20,'display','iter','Generations',1000);
% Running the genetic algorithm with desired options
[x_ga_opt, err_ga] = ga(h,(n*n+n+3*n+3),ga_opts);
Fitness function:
function mse_calc = mse_test(x, net, inputs,
targets,B,instances)
net = setwb(net, x');
y = net(inputs);
% Calculating the mean squared error
length(y);
mse_calc = sum(sum((y-targets).^2))/length(y);
end
```

5.3. Calculation errors

Total (resultant) error is determined by the formula:

$$e_0 = \frac{100}{m}\sum_{i=1}^{N}\left(\sum_{k=1}^{N} m_{ik} - m_{ii}\right), \%;$$

or

$$e_0 = 1 - \left(\sum_{i=1}^{N} m_{ii}\right)/m = 1 - Acc$$

where: $m = \sum_{k=1}^{N} m_k^c$ is the total number of objects in the training sample. The total error can also be calculated by the equivalent formula:

$$e_0 = \sum_{i=1}^{N} e_i P(D_i),$$

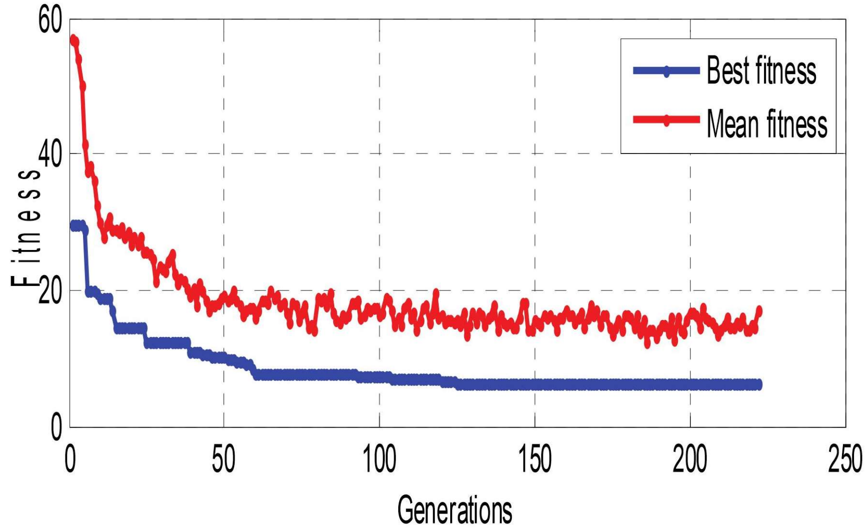
where $P(D_i) = m_i^c / m$, ($i = 1,2,..., N$) are the a priori probabilities of distribution of the objects of the respective classes *i*. According to the rule: $\sum_{i=1}^{N} P(D_i) = 1$, because the features $S \in D_i$ form a complete group of incompatible events.

General Error e,%	Multiplayer Perceptron Networks	NN with GA optimization	NN with GA optimization and feature selection
Train set	6,58	6,73	6,82
Test set	9,58	7,60	6,47

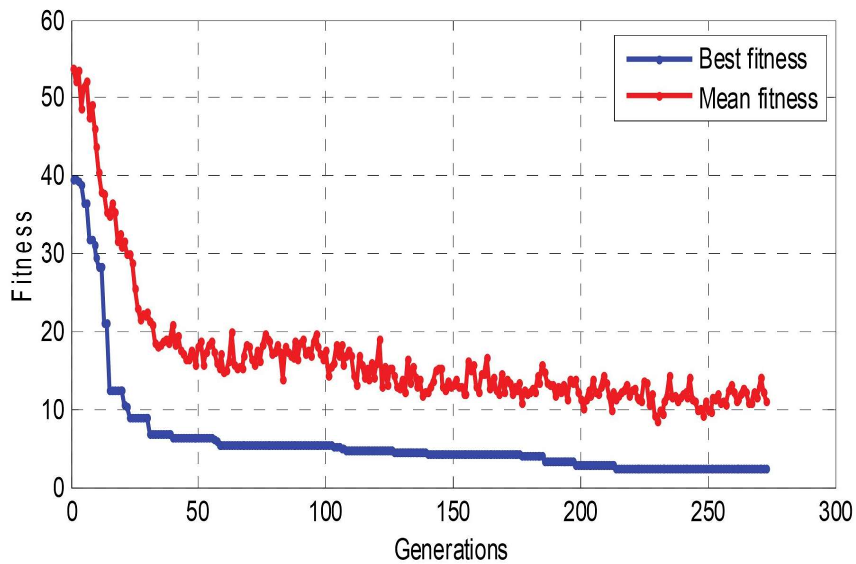
Table 2. Results Of The Classification

After the training the performance of the algorithm, and more precisely its accuracy, is evaluated using a control (test) sample. The resulting error of the test is shown in table of errors, which in the particular case of Table 2 .

These are trends in a NN genetic training by using the primary feature space and feature selection (Figure 4.).



a)



b)

Figure 4. Trends in a NN genetic training by using the primary feature space *a)* and feature selection *b)*

6. Conclusion

Experimentally, it was proven possible to adapt the genetic algorithms and Neural Networks in hybrid structure to spectral image recognition methods related to potato quality evaluation and classification problems.

The results in Table 2 demonstrate the advantages of hybrid neural-genetic algorithms for classification to Multilayer Perceptron Networks and structural genetic algorithm to optimize neural network architecture.

References

[1] Admuthe L. S., S. D. Apte, Neuro-Genetic Cost Optimization Model: Application of Textile Spinning Process, *International Journal of Computer Theory and Engineering*, Vol. 1, No. 4, 441-444, October 2009

[2] Damyanov C., Non-destructive Quality Evaluation in Automatic Sorting Systems, *UFT Academic Publishing House*, Plovdiv, p.360, 2006

[3] Petridis V., Paterakis E., Kehagias "A hybrid neural–genetic multimodel parameter estimation algorithm" *IEEE Trans. on neural networks*, vol.9 no.5 1998

[4] Shanthi D., Dr.G.Sahoo, Dr.N.Saravanan. Input Feature Selection using Hybrid Neuro-Genetic Approach in the Diagnosis of Stroke Disease, *IJCSNS International Journal of Computer Science and Network Security*, Vol.8 No.12, 99-107, December 2008

[5] Siddique, M.N.H.; Tokhi, M.O, Training neural networks: backpropagation vs. genetic algorithms, *International Joint Conference on Neural Networks*, Vol. 4, p. 2673 – 2678, 2001

[6] Titova T., V. Nachev A structural genetic algorithm to optimize neural network architecture, "Food Science Engineering and Technology 2007" Scientific Works of the UFT - Plovdiv, Volume LIV, St. 3, 101-107, 19-20.10. 2007

[7] Titova T., V. Nachev, Ch. Damyanov. Application of Genetic Algorithms for Neural Networks Trainings, International Conference Automatics and Informatics' 07, II-9–II-12, October 3 – 6, 2007, Sofia

[8] Wang W. J., Tang X. C., Li W. C., A variable structure neural networks model and its applications, IEEE TENCON, Vol. 2:5, p. 799-802, 1993.