

An efficient Neural Network algorithm for the p-median problem

E. Dominguez Merino and J. Muñoz Perez

Department of Computer Science, E.T.S.I. Informatica, University of Malaga
29071 Malaga, Spain
{enrique, munozp}@lcc.uma.es

Abstract. In this paper we present a neural network model and new formulation for the p-median problem. The effectiveness and efficiency of our algorithm under varying problem sizes are analyzed in comparison to conventional heuristic methods. The results for small-scale problems (less than 100 points) indicate that our implementation of algorithm is effective. Furthermore, we also have applied our algorithm to solve large-scale problems, demonstrating that a simple recurrent neural network, with an adapted formulation of the problem, can generate good solutions in a few seconds.

1 Introduction

One of the most popular problems in the facility location literature is the p-median problem. This model locates p facilities in some space (such as Euclidean plane or a network) that will attend n demand points. The objective of the model is to minimize the total (weighted) distance between the demand points and the facilities.

Kariv and Hakimi [20] showed that the p-median problem on a general network is NP-hard. This has resulted in the development of heuristic solution techniques in an effort to solve large-scale problems to near-optimality with a reasonable computational effort. Perhaps the most popular p-median heuristic was developed by Teitz and Bart [27]. This is a node-exchange procedure that attempts to improve the objective function value at each iteration. It is a simple to implement and it produces relatively good solutions to smaller problems when it is applied with multiple starting solutions.

There are more sophisticated heuristics for the p-median problem. A number of solution procedure have been developed for general networks. Most of the proposed procedures have been based on mathematical programming relaxation and branch-and-bound techniques. However, recently have been developed new procedure based on tabu search, neural networks, genetic algorithms and tree search. Thus, some proposed procedures include tree search (Christofides and Beasley [3], Bartezzaghi and Colomi [1]), lagrangian relaxation coupled with branch & bound (Narula, Ogbu and Samuelsson [24], Galvao [13]), tabu search (Ohlemüller [23]), heuristic and decision techniques (Hribar and Daskin [20], Hansen, Mladenovic and Taillard, Drezner and Guyse [18]), as well as Kohonen maps (Lozano, Warrior, Onieva and Larrañeta [23]).

Researchers have focused recently on the design of modern heuristics to solve the p-median problem. Modern heuristics can generate better results than the simple node-exchange procedure. But this modern heuristics usually are laborious and difficult to understand and to implement.

We have designed a simple neural network (such as Hopfield neural network) that generates good solutions to the p-median problem, comparable in quality to those of the node-exchange heuristics.

The rest of this paper is organized as follow: Section 2 describes the problem and gives a preliminary analysis. Section 3 shows how to apply a simple recurrent neural network to the problem. Section 4 contains illustrative and comparative simulation results. Finally, section 5 provides a summary and conclusions.

2 Problem Formulation

The p-median problem is well known and it has been studied during years. The p-median problem concerns the location of p facilities (medians) in order to minimize the total weighted distance between the facilities and the demand points. This problem is a multi-facility extension of the Weber problem which is widely accepted as the first formalized facility location problem in the literature. Operational generalizations of the

problem have been investigated by Cooper [1] using iterative approximation methods. ReVelle and Swain [26] provided an integer programming formulation for the discrete P-median problem, which is given below

Minimize

$$\sum_{i=1}^N \sum_{j=1}^N d_{ij} x_{ij} \quad (1)$$

Subject to:

$$\sum_{j=1}^N x_{ij} = 1 \quad i = 1, \dots, N \quad (2)$$

$$\sum_{j=1}^N x_{jj} = P \quad (3)$$

$$x_{ij} \leq x_{jj} \quad i = 1, \dots, N; j = 1, \dots, N \quad (4)$$

where

N is the considered number of points

P is the number of facilities or medians

d_{ij} is the distance (cost) between the demand point i and the facility j

$x_{ij} = \begin{cases} 1 & \text{if the point } i \text{ is associated with the facility } j \\ 0 & \text{otherwise} \end{cases}$

$x_{jj} = \begin{cases} 1 & \text{if the point } j \text{ is a facility} \\ 0 & \text{otherwise} \end{cases}$

The restriction (2) prevents that a demand point i is free, that is to say that does not have any facility associated to him. The restriction (3) establishes the number of facilities or medians. The last condition (4) assures the coherence of the solutions, a demand point i cannot be associated to the facility j ($x_{ij} = 1$) and in the point j not to be established a facility ($x_{jj} = 0$).

To the above formulation they have been applied numerous and different algorithms, but we do not know anyone that has been applied a neural network like the Hopfield network, because the restriction (4) not you ready to their application.

For that reason, in this paper we intends a new more appropriate formulation to the p-median problem:

Minimize

$$\sum_{i=1}^N \sum_{j=1}^N \sum_{q=1}^P d_{ij} S_{iq}^1 S_{jq}^2 \quad (5)$$

Subject to:

$$\sum_{q=1}^P S_{iq}^1 = 1 \quad i = 1, \dots, N \quad (6)$$

$$\sum_{j=1}^N S_{jq}^2 = 1 \quad q = 1, \dots, P \quad (7)$$

where

N is the number of points

P is the number of facilities (medians) to locate

d_{ij} is the distance among the points i and j

$S_{iq}^1 = \begin{cases} 1 & \text{if the point } i \text{ is associated with the group } q \\ 0 & \text{otherwise} \end{cases}$

$S_{jq}^2 = \begin{cases} 1 & \text{if } j \text{ is a facility of the group } q \\ 0 & \text{otherwise} \end{cases}$

Notice the existence of two types of neurons: S_{iq}^1 (allocations neurons) and S_{jq}^2 (facilities location neurons). Even, it is observed that the restrictions are much simpler than in the previous formulations. With the restriction (6) we only allow that a point associates to an only group, and with the condition (7) we make sure that in each group there is only one facility or median.

3 Competitive recurrent neural network model

The proposed neural network consists of a single layer of N interconnected binary neurons or processing elements. Each neuron i has an input h_i and an output $S_i \in \{0,1\}$. In order to design a suitable neural network for this problem, the key step is to construct an appropriate energy function E for which the global minimum is simultaneously a solution of the above formulation. The simplest approach to constructing a desired energy function is the penalty function method. The basic idea in this approach is to transform the constrained problem into an unconstrained one by adding penalty function terms to the objective function (5). These terms cause a high cost if any constraint is violated. More precisely, some or all constraints are eliminated by increasing the objective function by a quantity which depends on the amount by which the constraints are violated. That is, the energy function of the neural network is given by the Liapunov energy function defined as

$$E = \sum_{i=1}^N \sum_{j=1}^N \sum_{q=1}^P d_{ij} S_{iq}^1 S_{jq}^2 + \lambda_1 \sum_{i=1}^N \sum_{q=1}^P (S_{iq}^1 - 1)^2 + \lambda_2 \sum_{q=1}^P \sum_{j=1}^N (S_{jq}^2 - 1)^2 \quad (8)$$

where $\lambda_i \geq 0$ are penalty parameters that they determine the relative weight of the constraints. The penalty parameters tuning is an important problem associated with this approach.

In order to guarantee a valid solution and avoid the parameter tuning problem, we will divide our neural network in disjoint groups according to the two restrictions, that is, for the P-median problem with N points, we will have N groups, according to restriction (6), plus P groups, according to restriction (7). Then, we will reorganize our neurons in two matrices (one matrix per neuron type) where a group is represented by a row or column of the matrix according to neuron type.

S_{11}^1	S_{12}^1	\square	S_{1P}^1	$?$	group1
S_{21}^1	S_{22}^1	\square	S_{2P}^1	$?$	group2
\square	\square	\square	\square	\square	\square
S_{N1}^1	S_{N2}^1	\square	S_{NP}^1	$?$	groupN

$$\begin{array}{cccc}
S_{11}^2 & S_{12}^2 & \square & S_{1N}^2 \\
S_{21}^2 & S_{22}^2 & \square & S_{2N}^2 \\
\square & \square & \square & \square \\
S_{P1}^2 & S_{P2}^2 & \square & S_{PN}^2 \\
? & ? & & ? \\
\text{groupN ? 1} & \text{groupN ? 2} & & \text{groupN ? P}
\end{array}$$

Fig. 1. Neurons organization of the neural network for the P-median problem with N points. This shows two matrices, the first matrix contains the allocation neurons and the second contains the location neurons. The allocation neurons inside same group are in the same row of the matrix, and the location neurons inside same group are in the same column

In this model one and only one neuron per group must have one as its outputs, so the penalty terms are eliminated from the objective function. The neurons inside same group are updated in parallel. Then we can introduce the notion of group update. Observe that the groups are updated sequentially. Then, the energy function of the neural network is reduced to

$$E = \sum_{i=1}^N \sum_{j=1}^N \sum_{q=1}^P d_{ij} S_{iq}^1 S_{jq}^2 \quad (9)$$

Applying a gradient descent method, we obtain the inputs of the neurons of the network

$$h_{iq}^1 = \frac{\partial E}{\partial S_{iq}^1} = \sum_{j=1}^N d_{ij} S_{jq}^2 \quad (10)$$

$$h_{jq}^2 = \frac{\partial E}{\partial S_{jq}^2} = \sum_{i=1}^N d_{ij} S_{iq}^1 \quad (11)$$

The dynamics of the neural network are given by

$$S_{iq}^1(k+1) = \begin{cases} 1 & \text{if } h_{iq}^1(k) = \max_{1 \leq r \leq P} \{h_{ir}^1(k)\} \\ 0 & \text{otherwise} \end{cases} \quad (12)$$

$$S_{jq}^2(k+1) = \begin{cases} 1 & \text{if } h_{jq}^2(k) = \max_{1 \leq i \leq N} \{h_{iq}^2(k)\} \\ 0 & \text{otherwise} \end{cases} \quad (13)$$

then the energy function is guaranteed to decrease (see [12]). Thus, this energy decrease is maximized at every time. The following procedure describes the proposed algorithm based on the above neural network.

1. Set the initial state by randomly setting the output of one neuron in each group to be one and all the others neurons in the group to be zero.
2. Evaluate the initial value of the energy function (9).
3. Select a group g
4. Compute the inputs of the neurons in the group g , by (10) if $1 \leq g \leq N$ or by (11) otherwise.
5. Interchange the activated neuron in the group with the neuron with maximum input.
6. Repeat from step 3 until the neurons with the maximum input in each group are activated.

On step 3 we select a group randomly or sequentially. On step 5, if there are different neurons with the maximum input value, the algorithm must randomly select one of them.

Fig. 2 illustrates the energy decrease for a small-scale problem. In the first iterations, the curve slope is pronounced due to this energy decrease is maximized at every iteration. In fact, a good solution is obtained from iteration 400 with minimal error. In this case, the optimal solution is reached at 1003rd iteration.

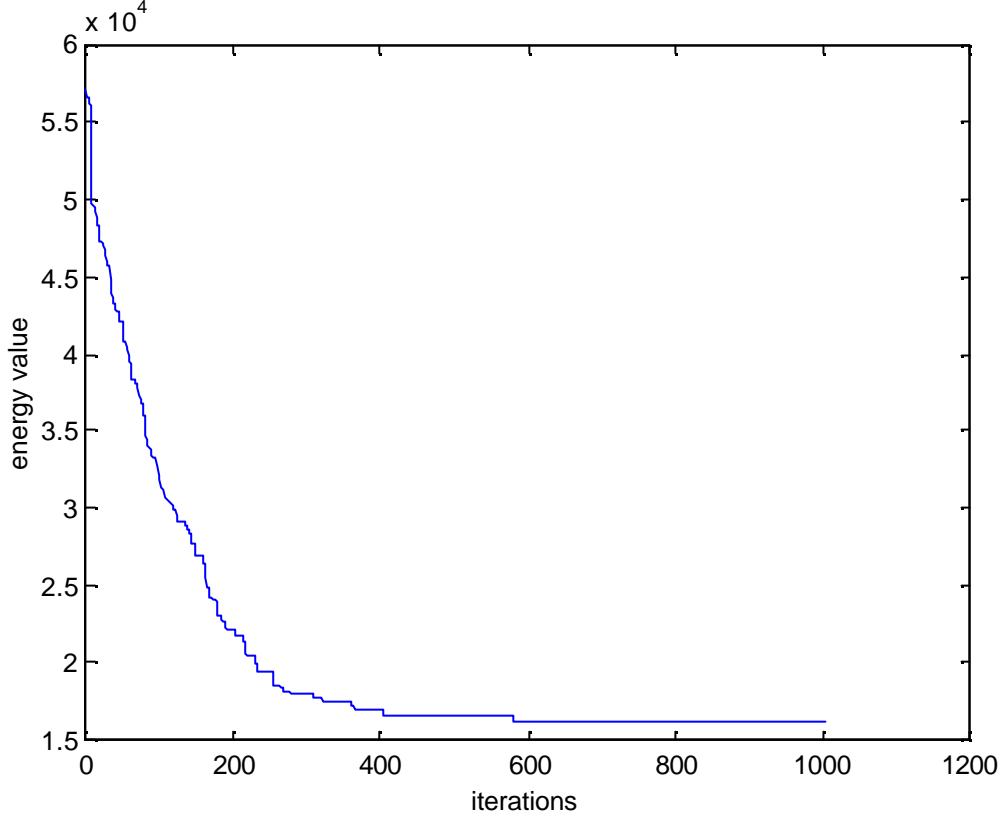


Fig. 2. Graphical representation of the values of the energy function on every iteration step in the 5-medians problem with 100 demand points

4 Simulations results

A random number generator has been used to generate the two-dimensional points. All points are distributed uniformly within the unit square.

For a comparison, we tested all algorithms on an Origin 2000 computer (Silicon Graphics Inc.). The computation of the optimal solutions has been carried out with an exact algorithm [13], using *branch & bound*.

We choose to compare the performance of proposed algorithm (NN) with the performance of the interchange algorithm proposed by Teitz and Bart [27] (T&B), since T&B is very simple to understand and implement, and it produces good solutions with limited computational effort. We recognize, however, that it is possible to generate better solutions for some instances of the p-median problem using other heuristics, such as langrangian relaxation or tabu search.

We first compare our implementation of NN with T&B and random search (RS) on several small-scale problems. **Table 1** lists the comparison of results from different algorithms with optimal solutions. For each instance, which it is represented by a row in the table, 50 randomly problems are generated and tested. T&B and N&N report 100% of optimality for their 50 randomly generated problems with 75 demand points ($N=75$) and 5 medians ($P=5$). The average error figures in the table represent the average percentage deviation from

the best solution calculated by an exact algorithm [13].

Table 1. Results for the three algorithms applied to small scale problems

N (demand points)	P (medians)	NN Avg. Error (%)	T&B Avg. Error (%)	RS Avg. Error (%)
75	5	0.00	0.00	0.10
75	10	0.00	0.09	0.12
75	20	0.10	0.17	0.30
100	5	0.00	0.05	0.10
100	10	0.01	0.21	0.25
100	20	0.09	0.39	0.41

RS is the fastest among these algorithms because there is virtually no processing overhead involved. RS performs so well in these cases because the difference between the number of feasible solutions evaluated and the total number of solutions is similar. However, RS is not likely to perform similarly well for larger problems due to size of the solution space. It is included here only to demonstrate that a simple random search method can generate good solutions for small-scale problems in less time.

In **Table 2** we compare the iterations and time of NN and T&B for the same problems. In this table we demonstrate that our implementation of NN produce better results in less time due to less number of iterations. Parallelism of the neural networks justified the less number of iterations, since the input of each neuron is calculated in parallel. In addition, our neural network is divided in groups and all neurons in the same group are parallel updated in only one iteration. Although an iteration of NN is slower than an iteration of T&B, our implementation of NN needs less time due to the great difference of iterations between NN and T&B. For example, the first row of **Table 2** shows that T&B is faster than our NN, nevertheless our NN needs much less iterations to obtain optimal results. For the rest of problems evaluated, the time of NN is lower than the time of T&B due to the less increase of iterations.

Table 2. Comparison of NN with T&B for small scale problems

N (demand points)	P (medians)	NN		T&B	
		Avg. Iterations	Avg. Time (s)	Avg. Iterations	Avg. Time (s)
75	5	1246	0.74	7051	0.68
75	10	1445	1.00	15253	1.47
75	20	1432	0.93	21422	2.01
100	5	1254	0.81	10965	1.05
100	10	1583	1.20	28199	2.52
100	20	2230	2.15	33241	3.14

The increase of average error or gap is another important characteristic. **Fig. 3** shows the increase of average error according to number of medians for instances of randomly problems with 200 demand points. In this case, we test 100 randomly large-scale problems to evaluate the error. This figure illustrates the good results obtained. In fact, our implementation of NN guarantees a solution with 99.4% of optimality for 50-mediands problems with 200 demand points. Note that, this large-scale problem has more than $4.5 \cdot 10^{47}$ feasible solutions and the solution is calculated in a few seconds, less than 4 seconds.

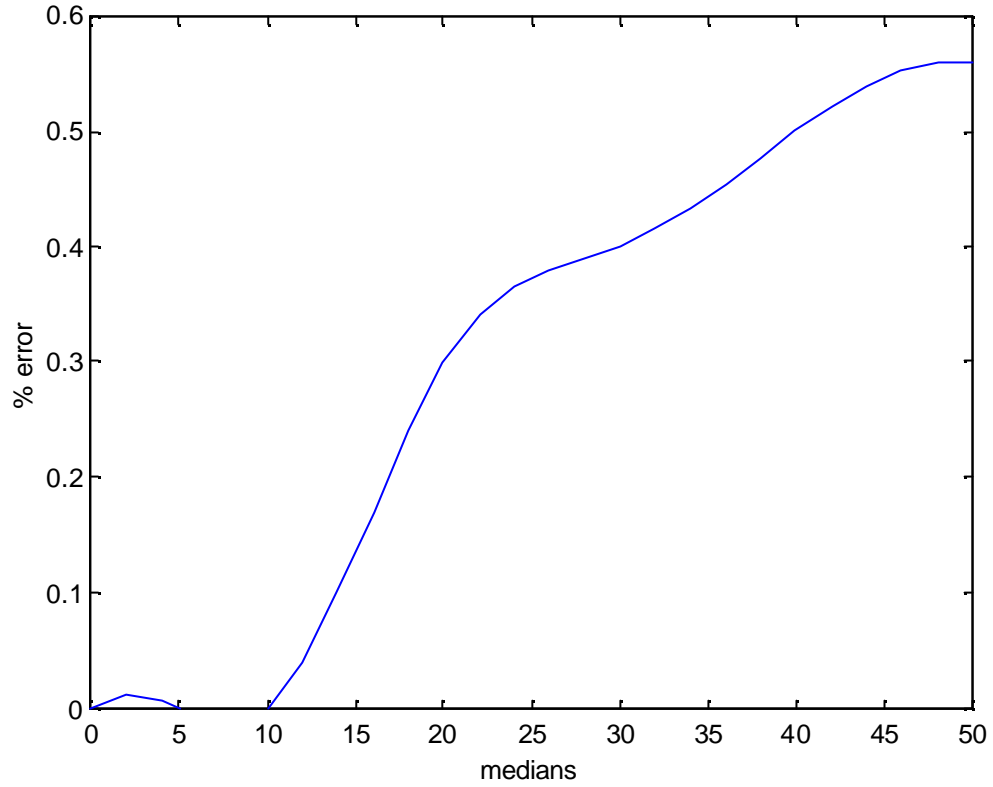


Fig. 3. Graphical representation of the relative error (gap) with 200 demand points

5 Conclusions

In this paper, we applied a neural network to solve the p-median problem. Our objective was to exploit the features of neural networks and demonstrate that a simple recurrent neural network can generate good solutions to location-allocation problems.

With the proposed mathematical model, we have tried to reduce the complexity of the problem that was observed in the resolution of the same one with other formulations. Besides the simplicity of this formulation, a decrease of the number of variables has been gotten, with the rising improvement in the yield of the methods or algorithms to apply. Also, the utility of the neural networks has been shown to treat optimization problems. Although, the proposed model guarantees that every converged state of neural network is equivalent to a feasible solution, and tuning the penalty parameters is not needed. Based on our computational tests, we believe that neural networks have the potential to be useful heuristic for the p-median problem.

In summary, in this paper we have achieved related with success two research areas, Localization and Neural Networks, to solve a classic NP-complete optimization problem, demonstrating this way, the fruitful existent cooperation among these areas.

6 References

- [1] E. Bartezzaghi, A. Colomi (1981), "A search tree algorithm for plant location problems", *European Journal of Operational Research* 7, 371-379
- [2] M.J. Canós, C. Ivorra, V. Liern (1999), "An exact algorithm for the fuzzy problem p-medians", *European Journal of Operational Research* 116, 80-86
- [3] N. Christofides, J.E. Beasley (1982), "A tree search algorithm for the problem p-medians", *European Journal of Operational Research* 10, 196-204
- [4] L. Cooper (1963), "Location-Allocation Problems", *Operations Research* 11, 331-343
- [5] E. Dominguez, J. Muñoz (2001), "Solving the m-median problem with a neural network", *Proc. of IX CAEPIA (Conference of the Spanish Associations for the Artificial Intelligence) Gijón, Spain*
- [6] Zvi Drezner (ed.) (1995), "Facility location: To survey of applications and methods", *Springer*
- [7] Zvi Drezner, Jeffery Guyse (1999), "Application of decision analysis to the Weber facility location problem", *European Journal of Operational Research* 116, 69-79
- [8] G. Galán Marín, J. Muñoz Pérez (1999), "A Net n-parallel Competitive Neuronal for Combinatory Optimization", *Proc. of VIII CAEPIA (Conference of the Spanish Associations for the Artificial Intelligence) Murcia, Spain, vol. 1*, 98-106
- [9] G. Galán Marín, J. Muñoz Pérez (1999), "A Design of a Neuronal Network for the resolution of the problem of the Four Colors", *Have Ibero-American of Artificial Intelligence* 8, 6-17
- [10] G. Galán Marín, J. Muñoz Pérez (2000), "An improved Neural Network Algorithm for the Bipartite Subgraph Problem", *Proc. of International Conference on Computational Intelligence and Neuroscience, Atlantic City, NJ USES*
- [11] G. Galán Marín, J. Muñoz Pérez (2000), "Finding Near-Maximum Cliques with Competitive Hopfield Networks", *Proc. of International ICSC Symposium on Neural Computation, Berlin, Germany*
- [12] G. Galán Marín, J. Muñoz Pérez (2001), "Design and Analysis of Maximum Hopfield Networks", *IEEE Trans. On Neural Networks* 12 (2), 329-339
- [13] Roberto D. Galvao (1980), "A dual-bounded algorithm for the problem p-medians", *Operations Research*
- [14] M.R. Garey, D.S. Johnson (1979), "Computers and intractability: To guide to the theory of NP-completeness", *W.H. Freeman and Co., New York*
- [15] S.L. Hakimi (1964), "Optimum locations of switching centers and absolute centers and medians of to graph", *Operations Research* 12, 450-459
- [16] S.L. Hakimi (1965), "Optimum distribution of switching centers in to communication network and some related graph theoretic problems", *Operations Research* 13, 462-475
- [17] P. Hanjoul, D. Peeters (1985), "A comparison of two dual-based procedures for solving the p-median problem", *European Journal of Operational Research* 20, 386-396
- [18] Pierre Hansen, Nenad Mladenovic, Eric Taillard (1998), "Heuristic solution of the multisource Weber problem ace to problem p-medians", *Operations Research Letters* 22, 55-62
- [19] J.J. Hopfield, D.W. Tank (1985), "Neural computation of decisions in optimization problems", *Biological Cybernetics* 52, 141-152
- [20] Michelle Hribar, Mark S. Daskin (1997), "A dynamic programming heuristic for the problem p-medians", *European Journal of Operational Research* 101, 499-508
- [21] Kariv O. and Hakimi S.L. (1979) "An Algorithmic Approach to Network Location Problem. Part 2: The p-Median". *SIAM J. Appl. Math.*, Vol. 37, pp. 539-560.
- [22] Love, Robert F., Morris, James G. and Wesolowsky, George O. (1998) "Facilities Location Models & Methods", *North-Holland*
- [23] S. Lozano, F. Guerrero, L. Onieva, J. Larrañeta (1998), "Kohonen maps for solving to class of location-allocation problems", *European Journal of Operational Research* 108, 106-117
- [24] Subhash C. Narula, Ugonnaya I. Ogbu, Haakon M. Samuelsson (1977), "An algorithm for the problem p-medians", *Operations Research* 25, 709-713
- [25] M. Ohlemüller (1997), "Taboo search for large location-allocation problems", *Journal of the Operational Research Society* 48, 745-750
- [26] C. ReVelle, R. Swain (1970), "Central facilities location", *Geographical Analysis*, 2, 30-42
- [27] M. B. Teitz, P. Bart (1968), "Heuristic methods for estimating the generalized vertex median of a weighted graph", *Operations Research* 16, 955-961