ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 385-391

Neural Network Algorithms for the p-Median Problem

E. Domínguez Merino, J. Muñoz Pérez and J. Jerez Aragonés
Department of Computer Science, E.T.S.I.Informatica

Campus Teatinos s/n, 29071
University of Malaga – Malaga, Spain

Abstract.
In this paper three recurrent neural network algorithms are proposed for the p-median problem according to different techniques. The competitive recurrent neural network, based on two types of decision variables (location variables and allocation variables), consists of a single layer of $2Np$ process units (neurons), where N is the number of demand points or customers and p is the number of facilities (medians). The process units form $N + p$ groups, where one neuron per group is active at the same time and neurons in the same group are updated in parallel. Moreover, the energy function (objective function) always decreases as the system evolves according to the dynamical rule proposed. The effectiveness and efficiency of the three algorithms under varying problem sizes are analyzed. The results indicate that the best technique depend on the scale of the problem and the number of medians.

1 Introduction

The traditional location problem is concerned with the location of one or more facilities, in some solution space, so as to optimize some specified criteria.

The p-median problem concerns the location of p facilities (medians) in order to minimize a weighted sum of the distance from each node (population center or customer) to its nearest facility. Kariv and Hakimi [7] showed that the p-median problem on a general network is NP-hard.

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, tree search and heuristic techniques. Thus, some proposed procedures include tree search (Christofides and Beasley [1]), lagrangian relaxation with branch & bound (Galvao [4], Erlenkotter [3]), tabu search (Ohlemüller [8])as well as

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 385-391

heuristic and decision techniques (Hribar and Daskin [6], Hansen, Mladenovic and Taillard [9]).

In this paper we present three recurrent neural network algorithms developed according to different techniques. The effectiveness and efficiency of these algorithms under varying problem sizes are analyzed. The true advantage of using recurrent neural networks for solving difficult optimization problems relates to speed considerations. Due to their inherently parallel structure and simple computational requirements, neural networks techniques are specially suitable for direct hardware implementation, using analog or digital integrated circuits, or parallel simulations. Moreover, the recurrent neural networks works have very natural implementations in optics.

The rest of this paper is organized as follow: Section 2 describes the problem and gives a preliminary analysis, also a new representation is proposed. In section 3 the competitive Hopfield model is applied to the problem and in section 4 the three different algorithms are proposed. Section 5 contains illustrative and comparative simulation results. Finally, section 6 provides a summary and conclusions.

2 Problem Formulation

The p-median problem is a well known problem that has been studied during years. The p-median problem is concerned the location of p facilities (medians) in order to minimize the total weighted distance between the facilities and the demand points. Domínguez and Muñoz [2] provided a new integer programming formulation for the discrete p-median problem, which is given below

Minimize

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{q=1}^{p} d_{ij} S_{iq} T_{jq} \tag{1}$$

Subject to:

$$\sum_{q=1}^{p} S_{iq} = 1 \quad i = 1, ..N \tag{2}$$

$$\sum_{j=1}^{N} T_{jq} = 1 \quad q = 1, ..p \tag{3}$$

where

N is the number of points, customers or population centers

p is the number of facilities (medians) to locate

d_{ij} is the distance between the point i and the facility j

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

$$T_{jq} = \begin{cases} 1 & \text{if } j \text{ is a facility in the cluster } q \\ 0 & \text{otherwise} \end{cases}$$

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 385-391

We have two types of variables (neurons): S_{iq} (allocation neurons) and T_{jq} (location neurons). Notice that the restrictions are much simpler that in the traditional formulation, and by the restriction (2) we only allow that each point is associated to an only cluster, and by the condition (3) we guarantee that there is only one facility or median in each cluster.

3 Competitive Recurrent Neural Network Model

The proposed neural network consists of two layers (allocation layer and location layer) of binary interconnected neurons or processing elements. In order to avoid the parameter tuning problem, our neural network is divided in disjoint groups according to the two restrictions, that is, for the p-median problem with N points, we have N groups according to restriction (2) in the allocation layer, and p groups according to restriction (3) in the location layer.

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 should ought introduce the notion of group update. The energy function of the neural network is defined as

$$E = \sum_{i=1}^{N} \sum_{j=1}^{N} \sum_{q=1}^{p} d_{ij} S_{iq} T_{jq} \tag{4}$$

The inputs of each neuron of the network are

$$h_{S_{iq}} = -\sum_{j=1}^{N} d_{ij} T_{jq} \tag{5}$$

$$h_{T_{jq}} = -\sum_{i=1}^{N} d_{ij} S_{iq} \tag{6}$$

where $h_{S_{iq}}$ is the activation potential of allocation neuron iq and $h_{T_{jq}}$ is the activation potential of the location neuron jq.

The central property of the proposed network is that the computational energy function always decrease (or remains constant) as the system evolve according to its dynamical rule

$$S_{iq}(k+1) = \begin{cases} 1 & \text{if } h_{S_{iq}}(k) = \max\limits_{1 \le j \le N} \{h_{s_{jq}}(k)\} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

$$T_{jq}(k+1) = \begin{cases} 1 & \text{if } h_{T_{jq}}(k) = \max\limits_{1 \le i \le N} \{h_{T_{iq}}(k)\} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

then the energy function is guaranteed to decrease (see [2] for a proof). Thus, this energy decrease is maximized at every time. A detailed procedure and comments of the neural network algorithm (NNA) is described in [2].

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 385-391

4 Neural Network Algorithms

The effectiveness and efficiency of neural networks algorithms (NNAs) vary with various coding techniques. This study proposes three different techniques for NNAs to solve p-median problems. They are the iterative median method (IM-NNA), the agglomerative median method (AM-NNA) and the stepwise median method (SM-NNA). The details of these three models are described as follows.

4.1 Iterative Median method (IM-NNA)

IM-NNA simply uses the neural network algorithm (NNA) to solve the p-median problem several times. The algorithm selects the best of the different simulations. The following is the algorithm.

1. Initialize $Cost_{opt} \leftarrow \infty$
2. Calculate a solution using NNA and get the $Cost = E$ using expression (4)
3. If $Cost < Cost_{opt}$ then save the new calculated solution in the previous step and $Cost_{opt} \leftarrow Cost$
4. Repeat the process (steps 2 and 3) several times

4.2 Agglomerative Median method (AM-NNA)

AM-NNA involves a series of successive merges. Initially, each point is a median. These initial clusters are merged according to their degree of distortion and the distances of their respective medians. The distortion of a cluster k is given by

$$D_k = \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} S_{ik} T_{jk} \tag{9}$$

The following are the steps in AM-NNA for grouping N points in p clusters.

1. Start with N clusters ($c = N$ where c is the current number of clusters) containing a single point.
2. Merge the two clusters which medians are the nearest, removing the median which distortion is lower.
3. Calculate a new solution with $c = c - 1$ using NNA
4. Repeat the two previous steps until the number of clusters is p ($c = p$)

ESANN'2003 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 23-25 April 2003, d-side publi., ISBN 2-930307-03-X, pp. 385-391

Table 1: Results for the three algorithms applied to small problems

N	P	Avg. Error (%)			
(demand points)	(medians)	IM-NNA	AM-NNA	SM-NNA	T&B
75	5	0,0	1,6	2,2	0,0
75	10	0,0	2,9	3,7	0,3
75	20	0,5	4,3	3,0	0,7
75	50	7,4	7,4	0,0	1,5
100	5	0,0	0,5	0,6	0,1
100	10	0,0	5,8	3,9	0,3
100	20	0,0	0,4	4,5	0,9
100	50	8,5	2,8	0,0	2,1

4.3 Stepwise Median method (SM-NNA)

An initial single cluster containing all points is divided into two clusters such that the objective function is optimized at this stage. Through each binary cluster process, a cluster is divided into two clusters. The selected cluster is divided according to its degree of distortion.

1. Initially, there is only one cluster ($c = 1$) containing all points.
2. Find the most distant point to the median of the most distortion cluster, establishing the new founded point as a new median
3. Calculate a new solution with $c = c + 1$ using NNA
4. Go to step 2 and repeat until the number of medians ($c = p$)

5 Simulations Results

A random number generator is using to generate two-dimensional points. All points are distributed uniformly within the unit square.

So much for the simulations, like for the calculation of the optimal solutions an Origin 2000 (Silicon Graphics) with MatLab has been used. The computation of the optimal solutions has been carried out with an exact algorithm [5], using branch & bound and technical heuristics.

We choose to compare the performance of proposed algorithms with the performance of the interchange algorithm proposed by Teizt and Bart [10] (T&B), since T&B is very simple to understand and implement, and it produces good solutions with limited computational effort.

We first compare the implementations of the three algorithms with T&B on several small-scale problems (less than 100 demand points). Table 1 lists the comparison of results from the different algorithms with optimal solutions. For each instance, which it is represented by a row in the table, 50 randomly

Table 2: Results for problems of 50 medians

N (demand points)	P (medians)	Avg. Error (%)		
		IM-NNA	AM-NNA	SM-NNA
75	50	7,4	7,4	0,0
100	50	8,5	2,8	0,0
125	50	9,3	2,3	0,0
150	50	4,2	0,0	1,5
175	50	4,7	0,0	1,5
200	50	5,6	0,0	1,9

problems are generated and tested. The average error figures in the table represent the average percentage deviation from the optimal solution.

IM-NNA is the most sensitive algorithm to the number of medians (P). However, we have obtained the best results with IM-NNA for small number of medians (less than 23 medians). That is, the performance of the IM-NNA is better when the number of medians is small, whereas SM-NNA is the best algorithm when the number of medians is greater for small scale problems. T&B is the fastest algorithm and we obtained good results for small scale problems, but T&B is less effective for large scale problems.

In Table 2 we show the results for the three different algorithms applied to problems with a greater number of medians. The best results for small scale problems are obtained with SM-NNA, since we had said above. However, the performance of AM-NNA is better than SM-NNA for large scale problems.

6 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-allocations problems. Also, we studied the effectiveness and efficiency of three different techniques applied to neural network algorithm. The results demonstrate that the best technique depend on the number of demand points (scale of the problem) and the number of medians.

With the proposed mathematical model, we have reduced 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 got. Also, the utility of the neural networks has been shown to treat optimization problems.

References

[1] N. Christofides and J.E. Beasley. A tree search algorithm for the problem p-mediates. *European Journal of Operational Research*, 10:196–204, 1982.

[2] E. Domínguez and J.Muñoz. An efficient neural network for the p-median problem. *Lecture Notes in Aritifitial Intelligence*, 2527:460–469, 2002.

[3] Donald Erlenkotter. A dual-bounded procedure for uncapacitated facility location. *Operations Research*, 26(6):992–1009, 1978.

[4] Roberto D. Galvao. A dual-bounded algorithm for the problem p-mediates. *Operations Research*, 28:1112–1121, 1980.

[5] P. Hanjoul and D. Peeters. A comparison of two dual-based procedures for solving the p-median problem. *European Journal of Operational Research*, 20:386–396, 1985.

[6] Michelle Hribara and Mark S. Daskin. A dynamic programming heuristic for the problem p-mediates. *European Journal of Operational Research*, 101:499–508, 1997.

[7] O. Kariv and S.L. Hakimi. An algorithmic approach to network location problem. part 2: The p-median. *SIAM J. Appl. Math.*, 37:539–560, 1979.

[8] M. Ohlemüller. Taboo search for large location-allocation problems. *Journal of the Operational Research Society*, 48:745–750, 1997.

[9] Nenad Mladenovic Pierre Hansen and Eric Taillard. Heuristic solution of the multisource weber problem ace to problem p-mediates. *Operations Research Letters*, 22:55–62, 1998.

[10] M. B. Teitz and P. Bart. Heuristic methods for estimating the generalized vertex median of a weighted graph. *Operations Research*, 16:955–961, 1968.