VSP International
Science Publishers
P.O. Box 346, 3700 AH Zeist
The Netherlands

Lecture Series on Computer and Computational Sciences
Volume 1, 2004, pp. 324-327

Matrix Bandwidth Minimization: A Neural Approach

D. López-Rodríguez[1] **and E. Mérida-Casermeiro**[2]

Department of Applied Mathematics,
E. T. S. I. Informática,
University of Málaga
29071 Málaga, Spain

Received 30 July, 2004; accepted in revised form 28 August, 2004

Abstract: The aim of this paper is to present a neural technique to tackle the bandwidth minimization problem. This neural model, successfully applied to other problems, presents better solutions than those of other classical approaches to this problem, as shown by the simulations performed.

Keywords: Bandwidth Minimization problem. Multivalued neural network. Combinatorial Optimization.

Mathematics Subject Classification: 05C78 (Graph labelling), 05A05 (Combinatorial choice problems).

PACS: 120304

1 Introduction

Bandwidth Minimization Problem (BMP) is one of the most recurrent themes in literature, since it has many applications in science and engineering, not only in linear algebra. The aim of BMP is to find a reordering scheme of rows and columns of a matrix, i.e. a permutation of its rows and columns, such that nonzero elements get as close as possible to the main diagonal.

This problem can be derived from graph theory in terms of minimizing the bandwidth of the adjacency matrix associated to a given graph.

So, BMP has indeed many optimization applications [6], from both linear algebra and graph theory, including circuit design, large hypertext media storage, VLSI design, finite element method for partial differential equations and resolution of large linear systems. In this last case, if matrix size is $N \times N$, Gaussian elimination can be carried out in $O(Nb^2)$ if matrix bandwidth is $b < N$, instead of a much slower $O(N^3)$.

BMP is also interesting due to its NP-completeness, proved in the general case in the middle 70's [10]. In the case of minimizing a graph bandwidth, Garey et al. [4] showed that the problem is NP-complete even if the maximum vertex degree in the graph is 3. So, the intractability of this problem is assured and thus there is need for algorithms to achieve near optimal solutions.

In the late 60's, E. Cuthill and J. McKee [2] introduced the first algorithm, CM, to approach BMP by constructing a level structure of the associated graph. Subsequently, Liu and Sherman [7] showed that reversing Cuthill-McKee's ordering can never increase the bandwidth of the matrix.

[1] Corresponding author. E-mail: dlopez@ctima.uma.es
[2] E-mail: merida@ctima.uma.es

Another approach based on the level structure of the associated graph is Gibbs-Poole-Stockmeyer (GPS) algorithm [5], which achieved results comparable to those of CM, with the advantage of being less time-consuming.

A Simulated Annealing procedure was presented by Dueck and Jeffs (1995). Although it finds better results than GPS or RCM in some particular cases, it should be noted that it takes up to 2000 times longer.

Some another approaches consist in a spectral decomposition of the matrix [1], although they are focused on minimizing the 'work-bound' of the matrix, i.e. the cost of a Cholesky factorization, and bandwidth of the permuted matrix can be high. In 2001, Corso and Romani modified this spectral technique to reduce both bandwidth and 'work-bound', but in general their algorithm performed better than RCM only when RCM was used at a preprocessing stage.

Since neural networks have been successfully applied to many other combinatorial optimization problems [8, 9, 11], improving the solutions given by classical algorithms, we expect that the model proposed in this paper achieves better results than other techniques.

2 Definition of the Problem

Bandwidth Minimization Problem (BMP) can be stated from two closely related points of view, one being matrix theory and the other being graph theory.

Let $M = (m_{i,j})$ be a $N \times N$ real symmetric matrix, with zero-diagonal. The bandwidth of M is defined as $w(M) = \max_{i,j=1,\dots,N}\{|i-j| : m_{i,j} \neq 0\}$.

BMP consists in searching for a permutation of the rows and the columns in M such that its nonzero elements lie as close as possible to the main diagonal, making the bandwidth of the new matrix minimum. Denoting by $\mathcal{S}_N$ the set of permutations of $\{1,\dots,N\}$ and by $M_\alpha = (m_{\alpha(i),\alpha(j)})$ $(\alpha \in \mathcal{S}_N)$ the permuted matrix, we look for a permutation $\sigma \in \mathcal{S}_N$ such that $w(M_\sigma) = \min_{\alpha\in\mathcal{S}_N} w(M_\alpha)$.

¿From the viewpoint of graph theory, BMP arises when all nodes in a given graph must be optimally labelled. So, let $G = (V, E)$ be an undirected graph with no self-connections, where $V = \{v_i\}$ is the set of vertices and E is the edge set. A *labelling* (or *numbering*) of G is a bijective function $\phi : V \to \{1, 2, \dots, N\}$ (where N is the cardinality of V) that assigns an index to each vertex. Then, the adjacency matrix of G with respect to ϕ is $A_\phi = (a_{\phi(i),\phi(j)})$, $a_{\phi(i),\phi(j)} = 1$ if and only if the edge connecting $v_{\phi(i)}$ and $v_{\phi(j)}$ exists. In this context, BMP consists in looking for a labelling that minimizes the bandwidth of the associated adjacency matrix.

So, this formulation is a particular case of the former one. Then, we can restrict ourselves to the matricial case.

3 The Neural Model

The proposed neural model (MREM) consists in a group of N neurons whose outputs, denoted by $\{s_1,\dots,s_N\}$, can take values in a finite set F, usually $F \subset \mathbb{Z}^+$, although F needs not be numerical. Then, the state of the i-th neuron is characterized by s_i. So, the state of the net can be expressed as a vector, called state vector, $\vec{S} = (s_1,\dots,s_N) \in F^N$. In our case, $F = \{1, 2, \dots, N\}$ and the only feasible states for the net are given by state vectors $\vec{S}$ associated to permutations in $\mathcal{S}_N \subset F^N$, such that $s_i = \sigma(i)$ for all $i \in F$ and a fixed $\sigma \in \mathcal{S}_N$. Thus, we can identify $\vec{S} \in \mathcal{S}_N$.

In order to solve BMP, an energy function, based entirely on the problem, is associated to state vectors. Given a matrix $M = (m_{i,j})$, this function can be defined in terms of the bandwidth w' of the permuted matrix $M_{\vec{S}} = (m'_{i,j}) = (m_{s_i,s_j})$, and the number $n_{w'}$ of nonzero elements that lie on the diagonals w'-th and $(-w')$-th of $M_{\vec{S}}$.

Table 1: Performance Results.

N	ρ	Best	Av.	RCM	N	ρ	Best	Av.	RCM
10	0.1	1	1.45	1	50	0.1	19	21.76	28
10	0.3	3	4.15	4	50	0.3	33	35.50	40
10	0.5	5	5.44	6	50	0.5	39	41.13	42
10	0.7	6	6.68	8	50	0.7	42	43.97	46
10	0.9	7	7.69	7	50	0.9	45	46.43	46
25	0.1	5	6.92	10	100	0.1	54	58.69	71
25	0.3	12	13.59	17	100	0.3	78	81.46	85
25	0.5	15	17.64	19	100	0.5	86	88.88	92
25	0.7	18	20.00	22	100	0.7	90	92.87	95
25	0.9	22	22.20	23	100	0.9	95	96.03	97

So, the energy function is $\boldsymbol{E(\vec{S}) = w' + \frac{n_{w'}}{A}}$ where A is the number of nonzero elements in M and $n_{w'} = \sum_{k=1}^{N-w'} \left[(1-\delta_{m'_{k,k+w'},0}) + (1-\delta_{m'_{k+w',k},0})\right]$.

The dynamics proposed in this paper attempts to minimize E by updating two neuron outputs, interchanging their values. Only changes that produce a state vector with less energy are allowed, and so, in each step, the energy function will never be increased.

First, the network computes the energy of a randomly generated permutation. At any time, the scheduling selects cyclically two neurons (p and q), and the network computes the increase of energy, resulting of the interchange of their outputs, with respect to the previous saved energy value. If this increase is negative, the next state vector is given by:

$$s_i(t+1) = \begin{cases} s_p(t), & \text{if } i = q \\ s_q(t), & \text{if } i = p \\ s_i(t), & \text{if } i \notin \{p, q\} \end{cases}$$

and the new value of the energy function is saved. This process is repeated until there is no change in the state vector that could decrease the energy. Therefore, the network always achieves a local minimum state in which the bandwidth is minimized.

4 Simulation Results

We have compared the neural algorithm proposed in this paper to the classical RCM algorithm due to its efficiency, as mentioned in the introduction. As RCM is implemented for MATLAB (`symrcm` command), our algorithm was also implemented and tested under such environment, on a Pentium III processor (451 Mhz).

The test set was formed by random binary symmetric matrices. These matrices were build depending on two parameters, N (size) and ρ (density, meaning that the number of 1's in a matrix is the closest integer to $\rho N(N-1)$). In order to cover a wide range of cases, N was taken from $\{10, 25, 50, 100\}$ and $\rho \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$. Thus, for each N and ρ as above, a total of 100 runs were performed on the test matrix.

Table 1 shows the average and best results of our simulations. So, we can confirm that our neural model improves RCM solutions, giving, in every case, a value for the best bandwidth indeed lower than that of RCM, and achieving, in most cases, better solutions on average, specially when performed over low-density matrices. Note that RCM always produces the same solution, however this neural model achieves better optimal solution by initializing with different state vectors.

In order to improve the knowledge about the performance of the proposed model, a problem obtained from a real situation has been analyzed. The studied matrix is the adjacency matrix of a graph, whose nodes represent some cities and edges representing that it exists a road between them.

For this problem, the number of cities is 13, and the adjacency matrix has its non zero elements $\{(1,2), (1,6), (1,9)\ (1,12), (2,3), (2,13), (3,4), (3,13), (4,5)\ (4,11), (6,7), (6,8)\ (8,9), (9,10), (11,12)\}$ and their symmetric ones. For this matrix the RCM algorithm obtains a bandwidth of 4, while the proposed model achieves an average of 4.08 and a best bandwidth of 3 (after 100 iterations). Note that, in general, RCM algorithm always obtains the same solution for a problem, while the network achieves different solutions that can improve the best previous result.

References

[1] S. T. Barnard, A. Pothen and H. D. Simon. *A spectral algorithm for envelope reduction of sparse matrices.* Journal Num. Lin. Alg. with Appl. **2**, 317-334, 1995.

[2] E. Cuthill and J. McKee, *Reducing the bandwidth of sparse symmetric matrices.* Proc. ACM National Conference, Association for Computing Machinery, New York, 157-172, 1969.

[3] G.M. Del Corso and F.Romani, *Heuristic Spectral Techniques for the reduction of Bandwidth and Work-bound of Sparse Matrices*, Numerical Algorithms, **28(1-4)**, 117-136, 2001.

[4] M.R. Garey and D.S. Johnson, *Computers and Intractability. A guide to the theory of NP-Completeness.* W. H. Freeman and Company, New York, 1979.

[5] N. E. Gibbs, W. G. Poole and P. K. Stockmeyer. *An algorithm for reducing the bandwidth and profile of a sparse matrix.* SIAM J. Numer. Anal. **13**, 236-250, 1976.

[6] Y. Lai and K. Williams, *A survey of solved problems and applications on bandwidth, edgesum, and profile of graphs*, J. Graph Theory, **31**, 75-94, 1999.

[7] J.W.H. Liu and A.H. Sherman. *Comparative analysis of the Cuthill-McKee and the reverse Cuthill-McKee ordering algorithms for sparse matrices.* SIAM J. Num. Anal. **12**,198-213, 1975.

[8] E. Mérida-Casermeiro, G. Galán-Marín and J. Muñoz-Pérez. *An Efficient Multivalued Hopfield Network for the Traveling Salesman Problem.* Neural Processing Letters, **14**, 203-216, 2001.

[9] E. Mérida-Casermeiro, R. Benítez-Rochel and J. Muñoz-Pérez. *Neural Implementation of Dijkstra's Algorithm.* Lecture Notes in Computer Science, **2686**, 342-349, 2003.

[10] C. Papadimitriou, *The NP-completeness of the Bandwidth Minimization Problem*, Computing, **16**,263-270, 1976.

[11] K. A. Smith, *Neural Networks for Combinatorial Optimization: A review of more than a decade of research.* INFORMS Journal on Computing **11-(1)**, 15-34, 1999.