# Algorithms for the Computation of Molecular Distance Matrix and Distance Polynomial of Chemical Graphs on Parallel Computers

P. Thangavel[†] and P. Venuvanalingam[*,‡]

Department of Mathematics and Department of Chemistry, Bharathidasan University,
Tiruchirapalli 620 024, India

Received July 17, 1992

A parallel algorithm adaptable to supercomputers is developed, based on the *modified matrix multiplication scheme*, to construct the molecular distance matrix from the simple connectivity matrix of chemical graphs. This algorithm is combined with our earlier algorithm[5] for the computation of distance polynomials of chemical graphs. The present algorithm can also handle weighted and directed graphs. Our algorithm for the construction of distance matrix takes only $O(\log_2 n)$ time in the CRCW PRAM model with $O(n^3)$ processors. The algorithm for the computation of distance polynomial will take only linear time in the same configuration.

## INTRODUCTION

Molecules are coded as graphs and matrices in chemical graph theory, and the underlying connection between molecular properties and the graph theoretic indices and matrix polynomials has been well recognized.[1-3] This has led to widespread applications of these mathematical constructs, especially in chemistry and chemical information theory. Computational aspects of solving such problems gained much importance in the recent past because of the large-scale availability of personal and other computers. Once computers entered the scene, there has been a spate of publications carrying newer computational techniques solving the problems computationally. Now, with the dawn of the era of *vector processors and parallel and supercomputers*, parallel algorithms are becoming popular,[4,5] and these algorithms, being faster and efficient, have started replacing the sequential ones. The present work involves developing a parallel algorithm that is adaptable to supercomputers for computing the molecular distance matrix from the simple connectivity matrix and the distance polynomial from the distance matrix. The molecular distance matrix is selected in this work as it has vast application potential as found in an extensive recent review by Rouvray.[6]

The distance matrix is one of the important matrices associated with a graph. There are two kinds of distance matrices; one based on topological distance and the other on geometrical distance. Molecular descriptors derived from topological and geometric distance matrices are used for characterizing the constitution and configuration of molecules.[7,8] In this work, we are interested only on the topological (graph-theoretical) distance matrix, and whenever we mention here the distance matrix, we mean only the topological distance matrix. The distance matrix of a graph $G$ will hereafter be referred as $D(G)$ whose entries $d_{ij}$ are the shortest paths between vertices $i$ and $j$. The distance matrix will hence be more complete (dense) than the adjacency matrix of a graph $A(G)$ and was described as containing more structural details than the latter. The structural details of connected graphs are more apparent in the distance matrix, while it is hidden in the adjacency matrix. It has also been observed[9] that distance polynomials are not unique in characterizing graphs, as characteristic polynomials are. This is borne out by the following observation that in some cases distance polynomials can differentiate isospectral graphs, although there are nonisomorphic graphs which have the same distance polynomial. Balasubramanian in his recent article has conjectured some problems relating to 'distance eigenvalues' based on his computed results.[9]

The distance matrix has been used in explicit and implicit forms in various branches of science,[10-20] and its widespread application has been well-documented in a recent article by Rouvray.[6] In chemistry, Wiener[18,19] and Platt[20] used the distance matrix in implicit forms to define path number, polarity number, and the first neighbor sum, correlating them with the physicochemical properties of hydrocarbons. The first explicit use of the distance matrix was reported by Clark and Kettle[14] in connection with stereochemical interconversion mechanisms. Dugundji and Ugi[16] used it in studying reaction graphs. Mihalić and co-workers have recently reported a comparative study of 10 molecular descriptors derived from topological and geometric distance matrices.[8]

Distance matrices of tree graphs have been extensively studied, and the properties of distance polynomials have also been investigated by several authors.[21-24] Křivka and Trinajstić[25] have investigated extensively distance polynomials of graphs. The distance matrix of weighted and directed graphs have been explored by Graham and co-workers.[24] Various sequential algorithms have been proposed so far[26-33] for computing distance polynomials from a simple connectivity matrix. Traditionally, the distance matrix was computed[26] from the powers of $A(G)$, and Bersohn[27] used a circularly linked list algorithm and showed that his method was efficient and fast. Müller and co-workers[33] developed an algorithm in which they used a "modified matrix multiplication scheme". They observed that their method was faster when a straight comparison was made with Bersohn's method. They also noted that when a simple modification was made in the Bersohn's algorithm it was faster than their own. Further, they claimed that their method would be superior, however, than the former due to the reason that their method would allow vectorization and, hence, will be adaptable for supercomputers while Bersohn's method did not. Recently Balasubramanian[9] has developed a program for computing distance polynomials of graphs based on his earlier method.[34] Balasubramanian[35] has also developed a program to compute the characteristic polynomial of the geometric distance matrix, which was defined as the geometry-dependent characteristic polynomial of

[*] Author to whom correspondence should be addressed.
[†] Department of Mathematics.
[‡] Department of Chemistry.

chemical graphs. Recently Nikolić and co-workers[36] have used the geometric distance polynomial for differentiating conformational isomers.

In this paper we developed parallel algorithms for constructing the distance matrix from the connectivity matrix and for computing distance polynomials of chemical graphs from the distance matrix on CRCW PRAM (Concurrent Read and Concurrent Write Programmable Random Access Machine) shared memory model.[37] The second section comprises definitions and the sequential algorithm for the construction of distance matrix. The third section contains the parallel algorithm for the construction of the distance matrix. In the fourth section we describe the computation of the distance polynomial from the distance matrix using the algorithm proposed in our earlier work.[5] Finally, we concluded that our proposed parallel algorithm could be extended to other parallel models like EREW PRAM (Exclusive Read and Exclusive Write Programmable Random Acess Machine) shared memory model, cube connected computers, etc.

## SEQUENTIAL ALGORITHM FOR COMPUTATION OF DISTANCE MATRIX

The adjacency matrix of a graph $G$ with $n$ vertices $A(G) = (a_{ij})$ is defined as follows:

$$a_{ij} = \begin{cases} 1 \text{ if } i \neq j \text{ and } i \text{ and } j \text{ are neighbors} \\ 0 \text{ otherwise} \end{cases}$$

The distance matrix of a graph $D(G) = (d_{ij})$ is defined as

$$d_{ij} = \begin{cases} 0 \text{ if } i = j \\ \text{the length of the shortest path between} \\ \text{vertices } i \text{ and } j \text{ if } i \neq j \end{cases}$$

If the graph is edge weighted then

$$a_{ij} = \begin{cases} 0 \text{ if } i = j \\ w_{ij} \text{ if } i \neq j \text{ and } i \text{ and } j \text{ are neighbors} \\ 0 \text{ otherwise} \end{cases}$$

For the purpose of the computation of the distance matrix $D(G)$ from the adjacency matrix, we redefine $A(G)$ as follows:

$$a_{ij} = \begin{cases} 0 \text{ if } i = j \\ 1 \text{ if } i \neq j \text{ and } i \text{ and } j \text{ are neighbors} \\ \infty \text{ otherwise} \end{cases}$$

and for edge weighted graphs

$$a_{ij} = \begin{cases} 0 \text{ if } i = j \\ w_{ij} \text{ if } i \neq j\text{, } i \text{ and } j \text{ are neighbors, and } w_{ij} \text{ is the weight} \\ \infty \text{ otherwise} \end{cases}$$

Here wer have used $\infty$ to represent some large number as edge weight where there is no edge between vertices $i$ and $j$. The algorithm for constructing the distance matrix works as follows. We construct the sequence of matrices $D_1$ (=$A$), $D_2$, $D_4$, $D_8$, ..., $D_n$ (assume that $n$ is power of 2). $D_k$ can be computed from $D_{k/2}$ by using the following rule:

$$d_{ij}^{(k)} = \min\ [d_{im}^{(k/2)} + d_{mj}^{(k/2)}]$$

In the first iteration the matrix $D_2$ is constructed, whose entries $d_{ij}^{(2)}$ give the shortest path between vertices $i$ and $j$ containing only one intermediate vertex or simply one edge from $i$ to $j$. That is, the entries of $D_2$ give the shortest distances with two or less edges. From $D_2$, $D_4$ can be constructed; the entries of $D_4$ would be the shortest paths or distances between vertices with three or less intermediate vertices and so on. Hence $D_n$ can be computed from $D_1 = A$, by computing $D_2$, $D_4$ etc. The entries of $D_n$ give us the shortest paths or distances between the corresponding vertices with $(n - 1)$ or less intermediate vertices. It can be easily seen that it requires $k = \log_2 n$ steps ($n = 2^k$) to compute $D_n = D$, the molecular distance matrix. The most crucial step is to compute $D_k$ from the matrix $D_{k/2}$, in which the computation of $D_k$ is just like matrix multiplication of $D_{k/2}$ by itself, if we replace multiplication by addition and addition by minimum operation in matrix multiplication. Hence, the time required for computing $D_k$ from $D_{k/2}$ is same as that of multiplication of $D_{k/2}$ by itself. Since the sequential algorithm for multiplication requires $O(n^3)$ time, the sequential algorithm for distance matrix computation requires $O(n^3 \log n)$ time. This algorithm has been reported by Müller and co-workers.[33]

## PARALLEL ALGORITHM FOR COMPUTATION OF DISTANCE MATRIX

We have reported[5] a constant time matrix multiplication algorithm using a CRCW PRAM model with $O(n^3)$ processors. In that we have also introduced CRCW PRAM with some write conflict resolution rules. In this paper, we shall use the MINIMUM CRCW PRAM. With the MINIMUM CRCW PRAM, if $n$ processors try to write into the same memory location, then the processor which holds the minimum value succeeds.

We first present a parallel algorithm which calculates $D_k$ from $D_{k/2}$. Initially assume that the matrix $D_{k/2}$ is stored in the shared memory of CRCW PRAM. This algorithm gives the output matrix $D_k$. A parallel algorithm to construct matrix $D_k$ using CRCW PRAM with $O(n^3)$ processors is given in Algorithm 1.

```
  Algorithm 1: $(DK^{(k/2)}, DK^{(k)}$
for $i$ = 1 to $n$ do in parallel
  for $j$ = 1 to $n$ do in parallel
    for $m$ = 1 to $n$ do in parallel
      MINIMUM WRITE $(d_{im}^{(k/2)} + d_{mj}^{(k/2)})$ into $d_{ij}^{(k)}$
    end for
  end for
end for
```

Now we present a parallel algorithm which constructs the distance matrix $D = D_n$ using CRCW PRAM with $O(n^3)$ processors. Assume that initially the adjacency or weight matrix $A = D_1$ is stored in the shared memory of CRCW PRAM. We use the matrix variable $DK[1..n,1..n]$ to store the matrix $D_k$ and the matrix variable $A[1..n,1..n]$ for storing $D_{k/2}$. The parallel algorithm is presented in Algorithm 2.

```
  Algorithm 2: Parallel-Distance-Matrix
for $l$:= 1 to $\log_2 n$ do
  perform algorithm 1[$A$,$DK$]
    for $i$ = 1 to $n$ do in parallel
      for $j$ := 1 to $n$ do in parallel
        ARBITRARY WRITE $DK[i,j]$ into $A[i,j]$
      end for
    end for
end for
```

The outermost for loop of the above algorithm requires $O(\log_2 n)$ time. Algorithm 1 requires constant time as MINIMUM WRITE operation takes only constant time.[37] Also ARBITRARY WRITE operation takes only constant

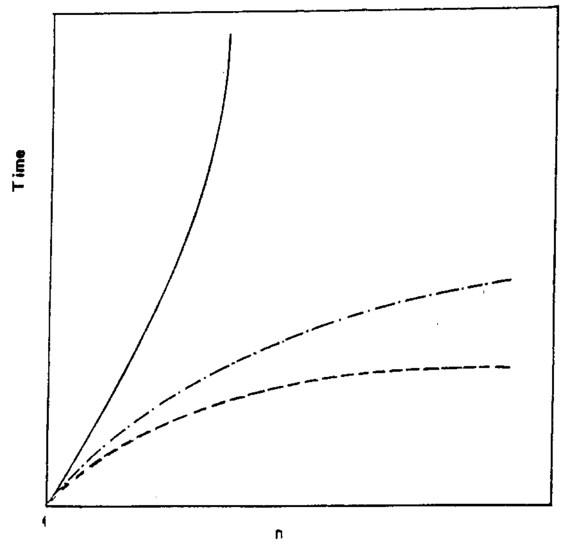

**Figure 1.** Schematic diagram showing variation of computation time on a serial machine and various parallel computation models with $O(n^3)$ processors. (—) Sequential machine; (· - ·) EREW PRAM and cube connected computers; (- -) CRCW PRAM.

time. Hence Algorithm 2 takes $O(\log_2 n)$ time in CRCW PRAM with $O(n^3)$ processors to compute molecular distance matrix.

## PARALLEL ALGORITHM FOR DISTANCE POLYNOMIAL

Once the distance matrix is available, the distance polynomial can be computed using our earlier parallel algorithm.[5] It was shown that the parallel algorithm for the computation of characteristic polynomials using the Le Verrier–Faddeev–Frame method takes $O(n)$ time using CRCW PRAM with $O(n^3)$ processors. Therefore the distance polynomials can be computed in $O(n)$ time using CRCW PRAM with $O(n^3)$ processors from the distance matrix. In the computation of distance polynomials from the adjacency matrix, the polynomial computation time dominates the computation time of the distance matrix, as the later takes less time than the former using CRCW PRAM with $O(n^3)$ processors. The matrix multiplication algorithm is also available for weaker parallel computation models like EREW PRAM, cube connected computers, etc.[38] The parallel algorithm for matrix multiplication takes $O((\log_2 n)^2)$ time using EREW PRAM or cube connected computers with $O(n^3)$ prcoessors. Based on this, parallel algorithms can be designed for the above weaker computational models to compute the molecular distance matrix and the distance polynomials. The computation time for the sequential and parallel algorithms are shown in Figure 1.

## ACKNOWLEDGMENT

We thank the referees for their helpful comments on the manuscript.

## REFERENCES AND NOTES

(1) Trinajstić, N. *Chemical Graph Theory*; CRC Press: Boca Raton, FL, 1983; Vols. I and II.

(2) Balasubramanian, K. Applications of Combinatorics and Graph Theory to Spectroscopy and Quantum Chemistry. *Chem. Rev.* **1985**, *85*, 599–618.

(3) Balaban, A. T., Ed. *Chemical Applications of Graph Theory*: Academic Press: London, 1976.

(4) Balasubramanian, K. Comments on the Characteristic Polynomial of a Graph. *J. Comput. Chem.* **1991**, *12*, 248–254.

(5) Venuvanalingam, P.; Thangavel, P. Parallel Algorithm for the Computation of Characteristic Polynomials of Chemical Graphs. *J. Comput. Chem.* **1991**, *12*, 779–783.

(6) Rouvray, D. H. The Role of Topological Distance Matrix in Chemistry. In *Mathematics and Computational Concepts in Chemistry*: Trinajstić, N., Ed.; Ellis Horwood: Chichester, 1986; p 295.

(7) Hosaya, H. Topological Index. Newly Proposed Quantity Characterizing the Topological Nature of Structural Isomers of Saturated Hydrocarbons. *Bull. Chem. Soc. Jpn.* **1971**, *44*, 2332–2339.

(8) Mihalić, Z.; Nikolić, S.; Trinajstić, N. Comparative Study of Molecular Descriptors Derived from the Distance Matrix. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 28–37.

(9) Balasubramanian, K. Computer Generation of Distance Polynomials of Graphs, *J. Comput. Chem.* **1990**, *11*, 829–836.

(10) Hoare, M. R. Structure and Dynamics of Simple Microclusters. *Adv. Chem. Phys.* **1979**, *40*, 49–135.

(11) Hogeweg, P.; Hesper, B. The Alignment of Sets of Sequences and the Construction of Phyletic Trees—An Integrated Method. *J. Mol. Evol.* **1984**, *20*, 175–186.

(12) Goad, W. B.; Kanchisa, M. I. Pattern Recognition in Nucleic Acid Sequences I: A General Method for Finding Local Homologies and Symmetrics. *Nucleic Acids Res.* **1982**, *10*, 247–263.

(13) Bonchev, D. *Information-Theoretic indices for characterization of chemical structures*; Research Studies Press: Chichester, U. K., 1983; p 127.

(14) Clark, M. J.; Kettle, S. F. A. Incidences and Distance Matrices. *Inorg. Chim. Acta* **1975**, *14*, 201–205.

(15) Muetterties, E. L. Topological Representation of Stereoisomerism II: The Five Atom Family. *J. Am. Chem. Soc.* **1969**, *91*, 4115–4122.

(16) Dugundji, J.; Ugi, I. An Algebraic Model of Constitutional Chemistry as a Basis for Chemical Computer Programs. *Topics Curr. Chem.* **1973**, *39*, 19–64.

(17) Mekenyan, O.; Dimitriv, S.; Bonchev, D. Graph-Theoretical Approach to the Calculation of Physico-Chemical Properties of Polymers. *Eur. Polym. J.* **1983**, *19*, 1185–1193.

(18) Wiener, H. The Structural Determination of Paraffin Boiling Points. *J. Am. Chem. Soc.* **1947**, *69*, 17–20.

(19) Wiener, H. Vapor Pressure–Temperature Relationships among the branched Paraffin Hydrocarbons. *J. Phys. Chem.* **1948**, *52*, 425–430.

(20) Platt, J. R. Influence of Neighbor Bonds on Additive Bond Properties in Paraffins. *J. Chem. Phys.* **1947**, *15*, 419–420.

(21) Graham, R. L.; Pollak, H. O. On the Addressing Problem of Loop Switching. *Bell Syst. Tech. J.* **1971**, *50*, 2495–2549.

(22) Graham, R. L.; Lovasz, L. Distance Matrix Polynomials of Trees. *Lect. Notes Math.* **1978**, *642*, 186–190.

(23) Simóes-Pereira, J. M. S.; Zamfirescu, C. M. Submatrices of Non-Tree-Realizable Distance Matrices. *Lin. Alg. Appl.* **1982**, *44*, 1–17.

(24) Graham, R. L.; Hoffman, A. J.; Hosaya, H. On The Distance Matrix of a Directed Graph. *J. Graph Theory* **1977**, *1*, 85–88.

(25) Křivka, P.; Trinajstić, N. On the Distance Polynomial of a Graph. *Apl. Matemat.* **1983**, *28*, 357–363.

(26) Biggs, N. L. *Algebraic Graph Theory*: Cambridge University Press: London, 1974; p 11.

(27) Bersohn, M. J. A Fast Algorithm for Calculation of the Distance Matrix of a Molecule. *J. Comput. Chem.* **1983**, *4*, 110–113.

(28) Peredunova, I. V.; Kuzmin, V. E.; Konovorotskii, V. E. Program Complex for the Calculation of Topological Characteristics of Molecules. *Zh. Struct. Chem.* **1983**, *24*, 145–147.

(29) Herndon, W. C., unpublished work as referred to in ref 6.

(30) Even, S. *Graph Algorithms*: Computer Science Press: Rockville, MD, 1979; p 11.

(31) Deo, N.; Pang, C. Shortest-Path Algorithm: Taxonomy and Annotations. *Networks* **1984**, *14*, 275–323.

(32) Chachra, V.; Ghare, P. M.; Moore, J. M. *Applications of Graph Theory Algorithms*: North Holland: New York, 1979; pp 67–77.

(33) Müller, W. R.; Szymonski, K.; Knop, J. V.; Trinajstić, N. An Algorithm for Construction of the Molecular Distance Matrix. *J. Comput. Chem.* **1987**, *8*, 170–173.

(34) Balasubramanian, K. Computer Generation of the Characteristic Polynomials of Chemical Graphs. *J. Comput. Chem.* **1984**, *5*, 387–394.

(35) Balasubramanian, K. Geometry-dependent Characteristic Polynomials of Molecular Structure. *Chem. Phys. Lett.* **1990**, *169*, 224–228.

(36) Nikolić, S.; Trinajstić, N.; Mihalić, Z.; Carter, S. On the Geometric Distance Matrix and the Corresponding Structural Invariants of Molecular Systems. *Chem. Phys. Lett.* **1991**, *179*, 21–28.

(37) Akl, S. G. On the power of concurrent memory access, Technical Report No. 88-231, Department of Computing and Information Science, Queen's University, Kingston, Ontario, Canada, July 1988.

(38) Dekel, E.; Nassimi, D.; Sahni, S. Parallel Matrix and Graph Algorithms. *SIAM J. Comput.* **1981**, *10*, 657–675.