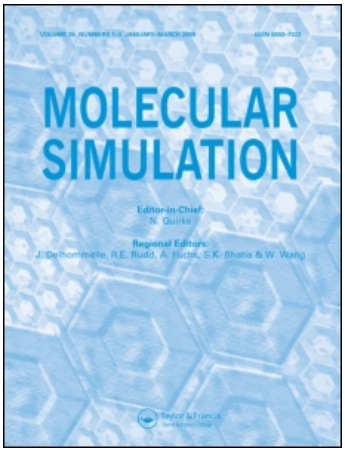

Molecular Simulation
Publication details, including instructions for authors and subscription information:
http://www.informaworld.com/smpp/title~content=t713644482

Parallel Computation of the Matrix of the Chemical Distances on the Connection Machine

Ottorino Ori[a]; Paolo Marenzoni[b]
[a] Thinking Machines Corporation, Cambridge, MA, USA [b] Dipartimento di Fisica, Università di Parma, Parma, Italy

To cite this Article Ori, Ottorino and Marenzoni, Paolo(1993) 'Parallel Computation of the Matrix of the Chemical Distances on the Connection Machine', Molecular Simulation, 11: 6, 365 — 372
To link to this Article: DOI: 10.1080/08927029308022520
URL: http://dx.doi.org/10.1080/08927029308022520

Molecular Simulation, 1993, Vol. 11, pp. 365–372

Reprints available directly from the publisher
Printed in the United States of America
Photocopying permitted by license only

PARALLEL COMPUTATION OF THE MATRIX OF THE CHEMICAL DISTANCES ON THE CONNECTION MACHINE

OTTORINO ORI

Thinking Machines Corporation, Cambridge, MA 02142-1214, USA

PAOLO MARENZONI

Dipartimento di Fisica, Università di Parma, Viale delle Scienze 43100 Parma, Italy

(Received January 1993, accepted January 1993)

We report here the parallel implementation of an original algorithm allowing a fast calculation of the distance matrix D of a graph representing a given chemical structure (molecule, polymer, crystal, etc.). Our algorithm fits perfectly in the SIMD parallel architecture of the Connection Machines CM-200 as we shall show. After discussing the performances of the parallel evaluation of D, we will end with a relevant application concerning C_{60} and C_{70} fullerenes. The present study applies to a generic globally connected graph without any restriction on the local connectivity of each graph's vertex.

KEY WORDS: Connection Machine, distance matrix, topological methods, fullerenes, Wiener index

1 INTRODUCTION

One of the most fascinating and powerful tools handled in modern theoretical chemistry for studying a broad range of physico-chemical properties is formed by a large class of topological methods relying only on the *topology* of a given chemical structure (molecule, polymer, crystal, etc.). Taking into account a pure topological description of, for example, a molecule, implies usually a very strong approximation: the molecule survives in the model only as a *graph* and consequent calculations are based on the sole information regarding the molecular connectivity. Following the excellent paper of Rouvray [1] dealing at an introductory level with topological methods, we can say that the heart of these new techniques is represented by the pattern (*chemical graph*) interconnecting the molecule's atoms, which is responsible for the ultimate architecture of the molecule. Geometric aspects (the actual three-dimensional shape of a molecule, the nature and lengths of the chemical bonds, the angles between the bonds) are seen as products of the molecular topology and thus simply ignored. In spite of that very crude assumptions, topological methods work out quite well in predicting a wide class of properties (see for example the progress in drug design reviewed in [2] and other applications reported in [3]) also for chemical systems showing an high structural and topological complexity (see [4] where the authors discuss energetic aspects of several polymers or [5], [6] for topological studies on zeolitic inorganic bulk lattices). These successes are

the best justification in order to enforce further investigation in that theoretical field.

The present article refers to the calculation of the matrix of the chemical distance D which represents one of the key object carrying out most of the topological information related to a chemical structure depicted as a graph G [3]. In particular, we will show how it is possible to set up an original algorithm working on a SIMD massive parallel computer as the Connection Machine CM-200 [9]. We are assuming G being a connected graph where each pair of vertices are connected by at least one path (the calculation for non-connected graphs can be easily done splitting them in two or more connected sub-graphs). For shake of simplicity the molecular graphs used in the examples are seen as *monochromatic* (all the bond distances have the same length) but the algorithm is applicable with minor variations also to *colored* graphs presenting different bond distances [10]. As possible application of such an algorithm, we shall present some result pointing to a very interesting theoretical finding: topology alone is able to produce valuable data on the molecular symmetry (e.g. the number of symmetry-inequivalent atoms) without knowing the actual point group of the molecule. That prominent result has been confirmed by the analysis of several fullerenes C_n with $n = 60, 70, 76, 78$ [7], [8] and we present here for the first time the calculations for C_{60} and C_{70}.

2 FROM CONNECTIVITY TO D

In what follows we will refer to a generic chemical structure having N atoms connected by a set of bonds. Our treatment aims to remain general and we do not impose any constraint on the number of bonds a particular atom can have. Adopting a topological point of view, our system corresponds to a chemical graph G with N vertices (or nodes) and a certain number of edges representing the chemical bonds. Labeling these N vertices with the first N integers, the connectivity of such a graph G can be given using a sparse $N \times N$ matrix A (the adjacency matrix) with $a_{ij} = 1$ when there is at least one edge connecting vertex i to j; $a_{ij} = 0$ otherwise. Starting from $A(G)$ we can get a second very important symmetric matrix called the matrix of the chemical distances of the graph, $D(G)$. In its elements we store the number of bonds separating two vertices i and j along the *shortest path* in the graph. Assuming the bonds have the same length (monochromatic graph), the D entries are integer numbers with $d_{ij} = 0$ when $i = j$, $d_{ij} = 1$ when i and j are nearest neighbours and $d_{ij} > 1$ otherwise. In spite of an apparent difficulty, we will show in this paragraph that the calculation of those shortest paths can be done in a very simple manner following an elegant algorithm which allows a natural parallel extension.

In the literature of the last decades several papers present different ways for calculating $D(G)$. In particular, reference [11] describes a worthwhile derivation of $D(G)$ based on a particular feature of the adjacency matrix of a *directed* graph G_d (or *digraph*). The particular importance of $A(G_d)$ (which is a non-symmetric matrix made by 0 and 1) derives from the fact that its power $A^k(G_d)$ is a matrix where the existence of an element $[A^k(G_d)]_{ij} = m$ (m integer) means there are m directed paths of length k from vertex i to vertex j. Following this property it is possible to devise a simple algorithm for getting $D(G)$ based in substance on successive multiplications of $A(G_d)$. Whereas that algorithm is very appealing

because of its clarity [12], the presence of several matrix multiplications leads to bad performances when large graphs (large N) are considered.

A very elegant method for the calculation of the $D(G)$'s entries can be found introducing the so-called *adjacency lists* which provide an effective manner for expressing the information stored in $A(G)$. The concept is simple: we assign to each graph's node i a small monodimensional array (say v_i) containing in the first place the number of connected vertices n_i and in the remaining n_i positions the labels $j_1, j_2, \ldots j_{n_i}$ of these vertices. We remember that, in the current approach, the value n_i does not need to be equal for all the lattice nodes. The evaluation of the shortest chemical distance d_{ij} between two points i and j of the graph is now quite simple. We know that the graph's vertices belonging to the first coordination shell (the set of nearest neighbours nodes) of a given node i have distance 1 from i. Analogously, vertices j's belonging to the kth coordination shell centered on i, have $d_{ij} = k (1 \leqslant k \leqslant M_i$, where M_i is the length of the longest chemical distance involving i). Thus, in order to compute the generic element d_{ij}, we have only to check if one among the nearest neighbours of j belongs to the kth shell of i. If this is the case, $d_{ij} = k + 1$.

We can store the N distances from a given vertex i in a monodimensional array w_i in which all the starting entries equal zero. In practice w_i will contain the ith row of $D(G)$: at the end of the calculation each element $w_i(j)$ will store the length d_{ij} of the shortest path connecting the node i to the node j. With the diagonal constrain $w_i(j) = 0$ when $i = j$, the steps for the w_i evaluation are the following four:

- Set to 1 the n_i elements $w_i(j)$ where the node j is the first neighbour of the vertex i: $w_i(j) = 1$ if the vertex j belongs to the set $v_i(l), l = 2, 1 + n_i$ (with $v_i(1) = n_i$);
- Set to 1 the value of the current coordination shell k;
- Set to $k + 1$ the zero elements $w_i(j)$ having at least one among the first neighbours of the vertex j already connected to i by a path of length k: $w_i(j) = k + 1$ if $w_i(j) = 0$ and $w_i(v_j(l)) = k$ for at least one of the nodes labeled by $v_j(l)$, where $l = 2, 1 + n_j$ (with $v_j(1) = n_j$);
- Increase by 1 the value of the current coordination shell k and return to the third step until all the w_i's elements, a part from the diagonal one, are different from zero.

The calculation of $D(G)$ is done when for each vertex i the related w_i array is known.

The above algorithm, jumping along the coordination shells, automatically gives the shortest paths d_{ij} and has a second relevant feature: it admits a natural parallel implementation which will be described in the next paragraph. We found that algorithm performs well also on a serial computer even for big graphs and, because it stores one $D(G)$'s row at a time, it offers also a manner for reducing the memory space required by this kind of calculation. For a generic graph G, the actual computational speed depends from three parameters: the number of nodes N; their mean coordination e (the mean value of the above n_i quantities); the length M of the longest path ($0 \leqslant d_{ij} \leqslant M$ and also $M = \max\{M_i\}$). If, for example, we are treating a set of graphs obtained translating a unit cell along three orthogonal directions (in such a manner we are in practice describing chemical graphs referring to parts of a crystal), the computational time goes as N^2.

We end the description of our method for evaluating $D(G)$ pointing out a subtle aspect of the above 4-steps algorithm. As we said, the crucial point making the algorithm fast consists in having a good representation in memory of the connections among the G's nodes (this is the role played by the v_i arrays). Each w_i is filled percolating among the coordination shells of i. In doing that, we take the point of view of the *unreached* (zero) elements of $w_i(j)$. In a figurative sense, all the unreached j's vertices are trying to pull connectivity from the nodes belonging to the kth shell centered on i. Only the nodes j's really connected to some vertex of that shell will succeed in this task. The dual strategy (when all the nodes belonging to the kth shell propagate their connectivity *pushing* into the $(k + 1)$th shell their nearest neighbours) is also valid, but forces to do a further computational check: the code has to avoid to put in the $(k + 1)$th shell the nodes that actually belong to the $(k - 1)$th one. In order to avoid that unnecessary complication we opted for the proposed *pulling* algorithm.

3 PARALLEL COMPUTATION OF D

The Connection Machine CM-200 series are data parallel computing systems [9]. What is important in the CM system is the way data are allocated into physical memory. Individual arrays of data are spaced out through the whole address space, so that each processor's local memory receives the same amount of data. If the number of elements in the array matches the number of physical processors (up to 64 K processors), then each local memory (usually 128 K bits) receives one element. The CM hardware allows each physical processor to behave as many *virtual processors* (VP's), each with a smaller memory. VP's allow CM programs to deal with large amount of data and to be completely scalable (a parallel code runs on a smaller CM system without reducing the size of the data). In order to get an application running on a CM-200, it is crucial that an algorithm is devised in such a manner that all VP's are doing the same set of parallel operations on local sets of data.

We specialize that approach to the problem of the parallel computation of $D(G)$. The above 4-steps algorithm easily originates a parallel version: figure 1 shows the data layout on the physical memory. On each VP we can store a $D(G)$'s column, in such a way that the computation can be carried out in parallel for a row at a time (the figure highlights the ith row of $D(G)$). In parallel for each unreached element j of a given row i, we test if it has a nearest neighbour m belonging to the kth coordination shell of the vertex i. If that is the case, we set $w_i(j) = k + 1$. The calculation of w_i ends when all its elements are different from zero (a part from $w_i(i)$ which equals identically zero). In order to avoid inter-processors communications, we found very convenient to store the information concerning the connectivity of each node (the vectors v_i's described above) on an array that lies in the memory of the serial computer acting as CM-200's front end (FE) [9]. Figure 2 presents the performances of a code written in CMF, the Fortran for the CM's. $D(G)$ is evaluated for a set of 3-dimensional (3d) cubic Bravais lattices with periodic boundaries conditions (tori) with given edge N (the vertices of their graphs are N^3). Figure 2 shows that the computational time scales with a power of N substantially lower than N^6 (the square of the G's vertices) producing a good improvement with respect to the serial implementation. Whereas an accurate fitting

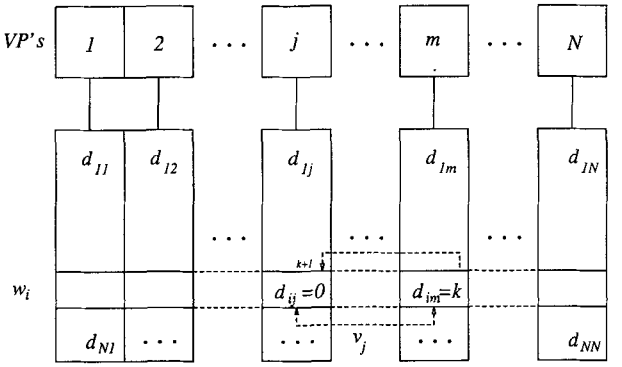

Figure 1 This figure shows how the elements of the distance matrix are stored on the CM's virtual processors (VP's) memory and how the algorithm puts the value of a generic d_{ij} in the $(k + 1)$th shell of i, when one of the nearest neighbours belongs to the previous shell. The nearest neighbours of j are stored in v_j. w_i stores the ith row of $D(G)$.

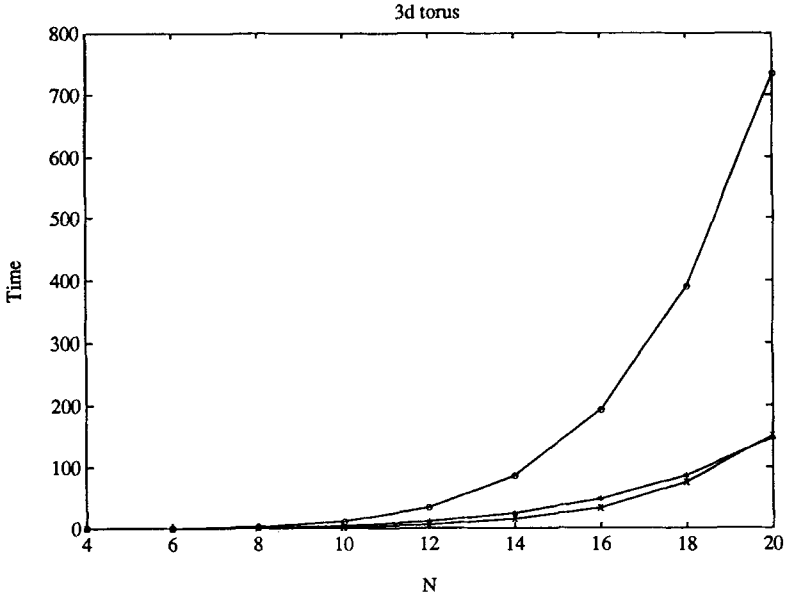

Figure 2 The x's represent the time (in seconds) spent by the parallel algorithm in the computation of the distance matrix on a 3d torus ($e = 6$) of N^3 vertices. The o's(*'s) mark the $N^6(N^5)$scaling low.

is still in progress (we remember that the computational time is a function of N, M, e,), we have to underline that the time spent for example on a 20^3 nodes graph is rather small (less than 3 minutes, on a 8 K processors CM-200) and encourages us to project very intensive applications involving big lattices (for example zeolites).

One among the most useful topological indices that we can get starting from the matrix of the chemical distances, is the celebrated Wiener number W [1]. W is simply defined as the half-sum of the $D(G)$'s elements

$$W = \frac{1}{2} \sum_{i,j=1}^{N} d_{ij} \tag{1}$$

We will show that this topological number contains enough structural information for the calculation of the number of inequivalent atoms of a given chemical graph (representing in that case a fullerene isomer). We report also an interesting property of W: for graphs obtained putting together a certain number of unit cells, W goes as a power of N (the lattice's edge) depending from the dimensionality d of the lattice we are studying. In particular for the above d-dimensional tori we found:

$$W_d(N) = \frac{dN^{2d+1}}{8} \tag{2}$$

$$M_d(N) = \frac{dN}{2} \tag{3}$$

The above results (N supposed to be even) generalize the $d = 1$ case reported in [4].

4 FROM D TO THE FULLERENE'S RESONANCE DATA

The contribution w_i to the sum (1) arising from a single vertex i can be rewritten as:

$$w_i = \sum_{k=1}^{M} k \cdot B_k(i) \tag{4}$$

where k represents the label of the kth-coordination shell centered on the ith vertex, and $B_k(i)$ is the number of graph vertices j's belonging to that shell. The $B_k(i)$ quantities will called the Wiener weights of the ith vertex (atom) of G. Starting from the connectivity lists of the graph representing C_{60} given in Table 1, we can compute in a fast manner $D(G)$ and the related N sets of Wiener weights $B_k(i)$, $k = 1, 2, \ldots, M$, $i = 1, 2, \ldots, N$ ($N = 60$).

The basic idea underlying the topological determination of some of the features (number of peaks and relative intensities) of the ^{13}C NMR spectrum of C_{60} is the following: the Wiener weights are fully representative of the environments of a molecular site i, in the sense that their knowledge implies the knowledge of the connectivity of the vertex i along all the molecular coordination shells. From the structural point of view, two vertices with the *same* $B_k(i)$ sets are symmetry-equivalent sites, and are therefore completely equivalent in respect of their physics-chemical properties. In particular, they will contribute to the *same* ^{13}C NMR line. Grouping the N vertices i in such a manner that each subset s contains sites having the same $B_k(i)$ set, both the number of the ^{13}C NMR lines L (the number of subset s) and the relative intensities $l(s)$ (the cardinality of s) are easily calculated, the normalization condition being $\Sigma_{j=1}^{L} l(s_j) = N$. From the structural point of view, a given set s contains symmetry-equivalent vertices. C_{60} and C_{70} are two natural candidates in order to test the reliability of these rules and a perfect agreement

Table 1 Connectivity lists for the faces of C_{60} fullerene: f is the face's label, n_i is the number of nearest neighbours, $n_i = 5(6)$ for a pentagon (hexagon). The connectivity lists for the C_{60}'s atoms are easily derived thinking each carbon atom sitting at the intersection of 3 faces.

f	n_i	*Connected faces*						f	n_i	*Connected faces*					
1	5	2	3	4	5	6	-	17	6	7	8	16	18	26	27
2	6	1	3	6	7	8	9	18	5	8	17	19	27	28	-
3	6	1	2	4	9	10	11	19	6	8	9	10	18	20	28
4	6	1	3	5	11	12	13	20	5	10	19	21	28	29	-
5	6	1	4	6	13	14	15	21	6	10	11	12	20	22	29
6	6	1	2	5	7	15	16	22	5	12	21	23	29	30	-
7	5	2	6	8	16	17	-	23	6	12	13	14	22	24	30
8	6	2	7	9	17	18	19	24	5	14	23	25	30	31	-
9	5	2	3	8	10	19	-	25	6	14	15	16	24	26	31
10	6	3	9	11	19	20	21	26	5	16	17	25	27	31	-
11	5	3	4	10	12	21	-	27	6	17	18	26	28	31	32
12	6	4	11	13	21	22	23	28	6	18	19	20	27	29	32
13	5	4	5	12	14	23	-	29	6	20	21	22	28	30	32
14	6	5	13	15	23	24	25	30	6	22	23	24	29	31	32
15	5	5	6	14	16	25	-	31	6	24	25	26	27	30	32
16	6	6	7	15	17	25	26	32	5	27	28	29	30	31	-

Table 2 Wiener number W, M, w, B_k, L and l quantities (see equations (1) and (4)) for C_{60} and C_{70} fullerenes: for each set s representative w and B_k values are reported; L (the number of resonance lines) is the number of distinct B_k sets; l is the computed site multiplicity and l_e is the experimental one.

C_{60} ($W = 8340$ $M = 9$ $L = 1$)												
w	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9		l	l_e
278	3	6	8	10	10	10	8	3	1		60	60
C_{70} ($W = 12375$ $M = 10$ $L = 5$)												
w	B_1	B_2	B_3	B_4	B_5	B_6	B_7	B_8	B_9	B_{10}	l	l_e
347	3	6	9	10	11	12	9	6	3	0	10	10
349	3	6	8	11	11	11	10	6	3	0	20	20
353	3	6	8	10	11	11	10	7	3	0	20	20
360	3	6	8	10	10	11	9	6	6	0	10	10
364	3	6	8	10	10	10	9	7	4	2	10	10

was found between the topological calculations and the experimental results (see Table 2). This purely topological approach works also on colored graphs and it has been successfully tested with bigger fullerenes [7], [8], [10]. When only the number of symmetry-inequivalcnt atoms is needed, the present method can conveniently replace the rather complex algorithm proposed by Fowler and Manolopoulos in a recent prominent article [13] for the determination of the fullerene's point group. On a CM-200, the resonance data of the C_{60}'s isomers (1812 molecules [14]) can be obtained in some minutes depending from the actual CM-200's size. Some theoretical investigations are in progress in order to understand the relationship between this interesting property of $D(G)$ and the molecular point group.

5 CONCLUSIONS

We devised an algorithm allowing the calculation of $D(G)$ on a data parallel computer, the CM-200. The CMF version of that algorithm performs very well also on big graphs G. We think our parallel implementation can offer enough speed for dealing with some challenging problems (for example, the topological simulation of diffusion mechanisms in zeolitic lattices) which can get benefited by modeling techniques having an intrinsic computational simplicity and also a substantial physical meaningfulness.

Acknowledgements

The authors wish to thank Thinking Machines Corporation for its support throughout the course of this work and Giulio and Claudio Destri for friendly cooperation.

References

[1] D.H. Rouvray, "Predicting Chemistry from Topology", *Scientific American*, 36–43 (October 1986).
[2] L.B. Kier and L.H. Hall, *Molecular Connectivity in Chemistry and Drug Research*, Academic Press, 1976.
[3] N. Trinajstic, *Chemical Graph Theory*, CRC Press, 1983.
[4] Ov. Mekenyan, S. Dimitrov and D. Bonchev, "Graph-theoretical Approach to the Calculation of Physico-chemical Properties of Polymers", *Eur. Polym. J.* **19**(12), 1185–1193 (1983).
[5] G. Melegari and O. Ori, "Computer-aided Topological Analysis of the Faujasite Lattice *I*: Exact Solution for Zeolite-X", *Mol. Sim.* **3**, 235–250 (1989).
[6] G. Melegari and O. Ori, "Computer-aided Topological Analysis of the Faujasite Lattice *II*: Monte Carlo Solution for Zeolite-Y", *Mol. Sim.* **3**, 325–335 (1989).
[7] O. Ori and M. D'Mello, "A Topological Study of the Structure of the C_{76} Fullerene", *Chem. Phys. Letters* **197**, 49–54 (1992).
[8] O. Ori and M. D'Mello, "Analysis of the Structure of the C_{78} Fullerence: A Topological Approach", *Appl. Phys. A* **55**, 00–00 (1992).
[9] *Connection Machine CM-200 Series Technical Summary*, Thinking Machines Corporation, Cambridge, Massachusetts, 1991.
[10] O. Ori, work in progress on C_{82} isomers.
[11] M.J. Clark and S.F.A. Kettle, "Incidence and Distance Matrices", *Inorganica Chimica Acta* **14**, 201–205 (1975).
[12] O. Ori, "Distance Matrices", Thinking Machines Corporation Internal Report, in preparation.
[13] D.E. Manolopoulos and P.W. Fowler, "Molecular Graphs, Point Groups, and Fullerenes", *J. Chem. Phys.* **96**(10), 7603–7613 (1992).
[14] D.E. Manolopoulos, Comment on "Favourable Structures for Higher Fullerenes", *Chem. Phys. Letters* **192**, 330–330 (1992).