ELSEVIER

Parallel pattern search energy minimization

David N.J. White

Department of Chemistry, University of Glasgow, Glasgow, Scotland

This article describes the adaptation of a pattern search algorithm, for the computational optimization of model molecular structures, to run on a parallel computer. The parallel efficiency (speedup) of the algorithm is discussed, and the rate of convergence of the parallel procedure is compared to that of the sequential implementation. More generally, the article also illustrates how inherently nonparallelizable stochastic procedures may be successfully parallelized by rearrangement of the algorithm. © 1997 by Elsevier Science Inc.

INTRODUCTION

In contrast to gradient methods of energy minimization, search methods will always guarantee to find the location of a local energy minimum, have large radii of convergence, and possess the ability to tunnel through small energy maxima in order to reach an adjacent local minimum. These desirable properties of search methods mean that there will frequently be circumstances in which they are useful; indeed, one popular molecular mechanics program uses a search procedure as its sole energy minimizer.[1]

Our interest in search procedures arose out of a desire to offer users of the VULCAN molecular mechanics program[2] a choice of energy minimizers for use as circumstances dictate. VULCAN currently offers sequential and parallel versions of the block diagonal Newton–Raphson (BDNR) procedure, together with sequential and parallel versions of a pattern search minimizer, the subject of this article, whereas sequential and parallel versions of a new variant of the truncated full matrix Newton–Raphson (TFMNR) optimizer will be added shortly. VULCAN is written entirely in Fortran and uses the COMFORT COMmunicating FORTran) programming environment/libraries[3] for interprocessor communication. We use VULCAN sequentially on 486/Pentium PCs and on our PARALLEL PC,[4] although VULCAN is highly portable and easy to implement on any PC, workstation, or parallel computer.

Address reprint requests to: David N.J. White, Department of Chemistry, University of Glasgow, Glasgow G12 8 QQ, Scotland.

Received 5 December 1996; revised 6 May1997; accepted 9 May 1997.

SEARCH METHODS

There are a variety of search methods that can be used for function minimization, including grid search, pattern search, regular simplex, and irregular simplex, listed in increasing order of effectiveness.[5] For historical reasons we use a modified pattern search procedure, the sequential pseudocode for which is given in Listing 1, omitting details such as data input, calculation of interaction lists, loading code onto the nodes, and the distribution of initial coordinates and interaction lists to the nodes. The basic idea is to establish a successful (i.e., energy-lowering) pattern of coordinate moves, and then to reapply this pattern exhaustively, with ever decreasing shifts in the successful directions, until no further energy reduction can be obtained. Successive iterations repeat this whole process after establishing a new pattern with a reduced size of exploratory shift. There are a number of hidden refinements in the practical implementation of this algorithm that serve to speed up the computation. For example, when an exploratory coordinate shift is made followed by a subsequent test for a reduction in energy, the energy of the whole molecule is not calculated; rather, only the energy accumulating from those interactions to which the current atom contributes is calculated. This is then compared with a similar evaluation of the energy, over the same interaction list, for the unshifted atom. The overhead incurred in generating lists of interactions (bond lengths, valency angles, torsion angles, nonbonded distances, Coulombic, and out-of-plane bending) for each atom is more than repaid by avoiding the necessity of calculating large numbers of invariant contributions to the difference in steric energy between the initial and shifted molecular coordinates.

PARALLEL IMPLEMENTATION

It is not intuitively obvious that search methods of energy minimization can be successfully parallelized because the atomic coordinates are updated one at a time, an action that presupposes that the most recent atomic coordinates are known to all processors at all times. This condition could be met only when the atomic coordinates are distributed across the nodes of a parallel computer, with every node broadcasting each new atomic coordinate, as it was determined, to every other node. This latter action would almost certainly lead to a parallel

Journal of Molecular Graphics and Modelling 15, 154–157, 1997

655 Avenue of the Americas, New York, NY 10010

1093-3263/97/$17.00
PII S1093-3263(97)00099-5

```
Do until all iterations completed or no energy reduction
    Reduce size of exploratory shift
    Do while no pattern found
      Reduce size of exploratory shift
      Do for each atom
        Do for each coordinate ( x,y,z )
          Calculate steric energy at coordinate ± exploratory shift
          If energy reduction
            move coordinate to new position
            note successful move
            pattern found
          End if
        End do
      End do
    End do
    Do until steric energy rises
      Do for each atom
        Do for each coordinate ( x,y,z )
          Try same pattern move again
          Accept new position if steric energy drops
        End do
      End do
    End do
    Do until exploratory shift too small
      Halve size of exploratory shift
      Do for each atom
        Do for each coordinate ( x,y,z )
          If no previous successful pattern move
            Calculate steric energy at coordinate ± exploratory shift
            If energy reduction
              move coordinate to new position
              note successful move
            End if
          Else
            Try same pattern move again
            Accept new position if steric energy drops
          End if
        End do
      End do
    End do
End do
```

Listing 1. The basic, sequential pattern search algorithm.

program that would run considerably slower than the sequential original! This is because communication is a much more expensive process than computation on a parallel computer, and is to be avoided wherever possible.

Molecular dynamics[6] programs and some molecular mechanics (energy minimization)[7] programs do not suffer from this difficulty because all of the atomic coordinates are updated *simultaneously* at the end of every time step or iteration. Parallelization involves, among other things, a global redistribution of the new coordinates calculated by the host and nodes at the end of every time step or iteration. Unfortunately, however, a sequential "one atom at a time" BDNR minimizer converges faster than an "all atoms at once" procedure because, within any one iteration, each atomic position is updated in the knowledge, rather than ignorance, of the positions of those atoms that have already been updated. The benefits of "one atom at a time" molecular mechanics and parallel computers would therefore appear to be mutually exclusive.

Nevertheless, we have successfully parallelized our "one atom at a time" BDNR energy minimization algorithm[2] by allowing each processor, host or node, to do the best it can with the slice of atoms assigned to it, knowing only the coordinates of the other atoms from the previous iteration. After each node has updated the positions of its slice of atoms it sends them to the host, which then broadcasts a complete set of updated coordinates to each node ready for the next iteration. The sequential equivalent of this procedure would be a "one atom at a time" minimizer, with the atoms divided up into an arbitrary number of slices, where each atom "sees" only the updated positions of atoms in the same slice as itself until all of the atomic positions, in every slice, have been updated. One might expect that convergence on the energy minimum might be slower in this case, but in fact the parallel algorithm takes the same number of iterations to reach convergence as the sequential one, although the initial rate of convergence is slightly slower, and the differences between the two sets of final coordinates and steric energies are so small as to be inconsequential.

The experience with the BDNR minimizer encouraged us to try the same approach with the pattern search minimizer, although we were not sure exactly what kind of behavior to expect. The pseudocode for the core of the host and node programs is given in Listing 2. The initial data distribution and the coordinate redistribution at the end of every iteration closely mirror that of the BDNR algorithm; all that has changed is the energy minimization code.

RESULTS

The results given in this section were obtained on a PARALLEL PC[4] comprising a Cyrix 6×86 P166+ host processor with 32 MB of memory and seven AMD 5×86-166 P90 node processors, each with 8 MB of memory. The initial test molecule was crambin,[8] a small protein with 327 atoms; the force field employed was the WBFF[9] plus polypeptide functionality[10]; and the iteration histories of a sequential plus a parallel minimization are shown in Figure 1. These, and all subsequent minimizations, were performed with no nonbonded distance cutoff and included no water molecules.

As one might expect, in the early stages of minimization the energy calculated by the parallel program is greater than that calculated sequentially, but by a declining margin as minimization proceeds, until at around 250 iterations the energies

```
Host:-
    Do until all iterations completed or no energy reduction
      Broadcast complete set of atomic coordinates to nodes
      Pattern search optimization of host slice of xyz
      coordinates using the boxed pseudocode from
                    Listing 1.
      Get slice of corrected coordinates from each node
    End do

Node:-
    Do until all iterations completed or no energy reduction
      Get complete set of atomic coordinates from host
      Pattern search optimization of node slice of xyz
      coordinates using the boxed pseudocode from
                    Listing 1.
      Send slice of corrected coordinates to host
    End do
```

Listing 2. The basic, parallel pattern search algorithm.

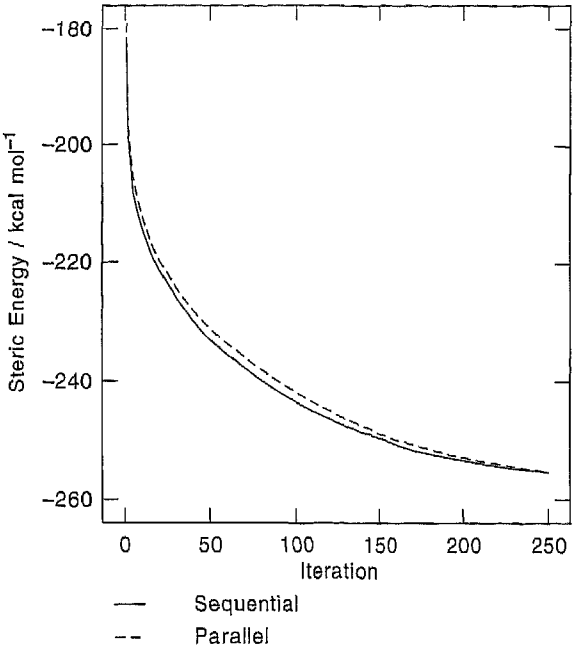

Figure 1. The variation in steric energy as a function of iteration number for sequential and parallel pattern search energy minimizations of crambin (327 atoms).

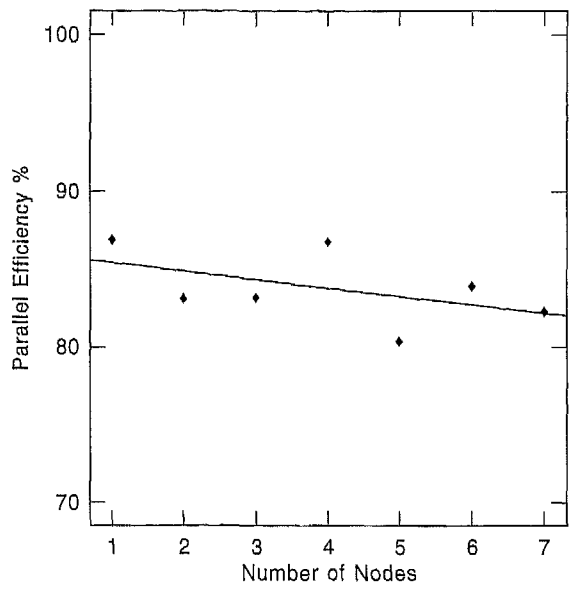

Figure 2. The variation parallel efficiency as a function of the number of node processors for the crambin pattern search minimization.

become essentially equal, and remain so until convergence on the energy minimum, at about −260 kcal · mol^{-1}, after roughly 400 iterations. Convergence generally takes between 50 and 500 iterations depending on the curvature of the energy minimum for the molecule concerned.

Because the ratio of computation time to communication time is greater for parallel pattern search than for parallel BDNR the parallel efficiency of pattern search is correspondingly higher at 83% as opposed to 74% for BDNR, using seven nodes (this is somewhat less than previously reported[2] because the processing power of our PARALLEL PC has been upgraded to a much greater extent than the interprocessor communication speed). Figure 2 shows a plot of the parallel efficiency versus the number of nodes in use for the crambin minimization; it can be seen that the efficiency declines gently as more nodes are added. The slope of the graph indicates that up to roughly 64 node processors could be used before the parallel efficiency drops below 50%. It would be possible to make effective use of more than 64 processors if either the interprocessor communications were overlapped with computation, or faster interprocessor links were used.

The rate of decline in parallel efficiency as a function of number of nodes for the pattern search algorithm is about half that for the BDNR,[2] which implies an effective maximum of 32 processors using the latter procedure. However, a 32-processor BDNR minimization will still be more cost effective in purely speed terms than a 64-processor pattern search calculation, because the former converges several times faster.

Parallel pattern search algorithms with their high compute-to-communicate ratios are ideal for running on parallel computers consisting of networks of workstations, most of which have very slow interprocessor communications links implemented with a single 10-Mbits/s ethernet. The parallel efficiencies so obtained would be somewhat lower than those recorded here, but workstations networked with a 100-Mbits/s erthernet or ATM would probably be as efficient as our PARALLEL PC.

The volume of computation performed by the host or node processor during a parallel pattern search minimization is potentially much less deterministic than that performed by the equivalent processor during a parallel BDNR calculation. The number of successful coordinate shifts in a pattern, and the number of times that a particular pattern can be reapplied, are both potential sources of variation in the computational load of a processor. Significant imbalances in computational load will lead to depressed parallel efficiencies. Our experience so far is that variation in parallel efficiency from molecule to molecule of similar size is rarely greater than 10–15%. Table 1[8,11–14] shows the variation in parallel efficiency with molecular size, using a variety of proteins as test molecules. The parallel efficiency is essentially independent of molecular size and is not expected to decline for molecules even bigger than those shown in Table 1, simply because the compute-to-communicate ratio will increase marginally as molecular size increases. The parallel efficiencies of greater than 100% arise

Table 1. The parallel efficiencies for host plus seven node energy minimizations on a number of proteins[a]

Molecule	Number of atoms	Parallel efficiency
Avian pancreatic protein[11]	115	87.0
Crambin[8]	327	82.3
Ubiquitin[12]	602	106.1
Phospholipase A_2[13]	1007	84.7
Tosyl elastase[14]	1822	108.7

[a] The polypeptide chain of the avian pancreatic protein model was cut after residue Asp-16 in order to obtain a small enough number of atoms.

when the sequential code becomes bogged down in making large numbers of small energy reductions; the parallel code avoids this because of the different path taken to the minimum. Unfortunately the same process can also happen in reverse, but is observed less frequently.

We have restricted discussion to large molecules because the overall run times for, say, drug-sized molecules are so short on a single processor (on the order of fractions of a second to seconds) that it is not worth using the parallel machine; particularly as parallel efficiencies are low (~50%) for small molecules, for which the compute-to-communicate ratio is also low.

CONCLUSION

We have demonstrated that it is possible to implement parallel pattern search energy minimization procedures that have essentially the same rate of convergence as their sequential counterparts. Furthermore, because search procedures perform more calculations per iteration than gradient procedures they can be parallelized with greater efficiency; although for general purpose use gradient procedures are usually faster, despite their inferior parallel efficiencies, because of their superior rate of convergence. The methodology described here should be equally effective with simplex energy minimization procedures.[1]

This does, however, suggest that the parallel efficiencies of gradient energy minimization procedures could be improved if some way could be found to increase the amount of productive calculation between the communication-bound global coordinate updates at the end of each iteration. One obvious way in which this could be accomplished is for the host and nodes to perform more than one cycle of energy minimization on their local slice of atoms before the global coordinate update. The effectiveness of this approach is currently being investigated.

REFERENCES

1 Vinter, J.G., Davis, A., and Saunders, M.R. The COSMIC molecular mechanics force field. *J. Comput.-Aided Mol. Design* 1987, **1,** 31

2 White, D.N.J. Sequential and parallel molecular mechanics calculations. *J. Mol. Graphics* 1996, **14,** 119–129

3 Bissland, L. and White, D.N.J. Parallel molecular mechanics calculations. In: *Transputer Applications and Systems '95* (Cook, B.M., et al., eds.). IOS Press, Oxford, 1995, pp. 473–487

4 White, D.N.J. A hardware and software environment for parallel processing with PCs. *Comput. Chem.* 1996, **20,** 381–384

5 Walsh, G.R. *Methods of Optimization.* John Wiley & Sons, London, 1975, pp. 74–104

6 Vincent, J.J. and Merz, K.M., Jr. A highly portable parallel implementation of AMBER4 using the message passing interface standard. *J. Comput. Chem.* 1995, **16,** 1420

7 Watowich, S.J., Meyer, E.S., Hagstrom, R., and Josephs, R. A stable and rapidly converging conjugate gradient method for energy minimization. *J. Comput. Chem.* 1988, **9,** 650–661

8 Hendrickson, W.A. and Teeter, M.M. Structure of the hydrophobic protein crambin determined directly from the anomalous scattering of sulphur. *Nature (London)* 1981, **290,** 107

9 White, D.N.J. and Bovill, M.J. Molecular mechanics calculations on alkanes and non-conjugated alkenes. *J.C.S. Perkin II* 1977, 1610

10 White, D.N.J., Ruddock, J.N., and Edgington, P.R. Molecular mechanics. In: *Computer-Aided Molecular Design* (Richards, W.G., ed.) IBC Technical Services, London, 1989

11 Blundell, T.L., Pitts, J.E., Tickle, I.J., Wood, S.P., and Wu, C.W. X-ray analysis (1.4Å resolution) of avian pancreatic polypeptide: A small globular protein. *Proc. Natl. Acad. Sci. U.S.A.* 1981, **78,** 417

12 Vijay-Kmar, S., Bugg, C.E., and Cook, W.J. The structure of ubiquitin refined at 1.8Å. *J. Mol. Biol.* 1987, **194,** 531

13 Dijkstra, B.W., Van Nes, G.J.H., Kalk, K.H., Brandenburg, N.P., Hol, W.G.J., and Drenth, J. The structure of bovine pancreatic prophospholipase A2 at 3.0Å resolution. *Acta Crystallogr.* 1982, **38, 793**

14 Sawyer, L., Shotton, D.M., Campbell, J.W., Wendell, P.L., Muirhead, H., Watson, H.C., Diamond, R., and Ladner, R.C. The atomic structure of crystalline porcine pancreatic elastase at 2.5Å resolution: Comparisons with the structure of α-chymotrypsin. *J. Mol. Biol.* 1978, **118,** 137