Conformation of Biomolecules: A search by Genetic Algorithms

V. Sundararajan [1] A. S. Kolaskar[2]

A model has been developed using genetic algorithms to predict the structure of a polypeptide chain. The algorithm is based on the principle of evolution and it improves the solution of the posed problem by genetic operations cross- overs and mutations. Dihedral angles are taken as the basic variables for the structure of the molecules and genetic operations are carried over on a population of binary strings of (ϕ, ψ). First, a sequential code is developed in FORTRAN on a standard workstation. Two parallel models have been implemented on a distributed computing platform PARAM, developed by CDAC. The methodology and the practical aspects are presented with a case study of a homo-octa-peptide. The simple data parallel model performed with more than 80% efficiency on PARAM machines. The usefulness of the migration model, developed for the first time, in achieving efficiency is stressed. The migration model proved to be more efficient and the minimisation for the octa-peptide improved from 2% to 10%. This improvement is expected to be more pronounced for larger molecules. The comparison with the molecular dynamics simulations is also discussed.

1 Introduction

Biomolecules are heterogeneous polymers having complex structures. The structure of these molecules decide their function. Hence, it's important to know the structure of the biomolecules. Experimentally, X-ray and Nuclear Magnetic Resonance (NMR) techniques are used to determine the structure. For that, the molecules are required to be synthesised either in crystal form or in large quantity. If both are not possible, computer simulation is the other alternative yet highly challenging. Such an ability would go a long way in the applications of medicine and bio-technology especially, drug design.

Structure prediction or conformational search is normally treated as an optimisation of energy of the molecule. The calculus based minimisation procedures are good only at locating a local minimum on the energy surface. Dynamical quenching is computationally demanding but, one is not sure of reaching the global minimum energy structure. Recently, randomised techniques like Genetic Algorithms(GA) [1, 2] and simulated annealing(SA) [3] are being widely used in the search of the global minimum. SA is based on the search by doing random process of disturbing the configurations and adopt a slow and step by step *cooling* procedure to arrive at the global minimum energy configuration. On the other hand, GA is based on the search by random selection of configurations and go through the genetic operations viz. cross-over and mutation to arrive at the global minimum. In addition, these techniques could exploit the power of parallel computing. GA is not only better algorithmically, but also quite amenable for parallelisation. The present work exploits both of these aspects.

A number of studies have been made using GA on structure prediction for protein [4, 5, 6, 17] as well as DNA/RNA [9, 10, 11] molecules. A comprehensive review especially, for protein structure prediction, has been made by Pederson and Moult [13].

In the present work a protein has been modelled to be made up of amino acid sequence and restricting the variation only in the torsion angles. The sidechains are also assumed to be fixed. A set of conformations in torsion angle space has been evolved using GA for a homo-octa-peptide. The parallelism in GA as well as the power of the PARAM machines developed at CDAC is exploited. Two models *viz* data parallel and migration models have been implemented. Detailed aspects of implementation, results and performance are presented.

The rest of the paper is organised as follows: The second section describes in brief the implementa-

[1]Centre for Development of Advanced Computing
Pune University Campus, Pune 411 007 INDIA
[2]Bioinformatics Centre, University of Pune 411 007 INDIA

tion details especially, the parallel models. The third section describes a sample application on an octa-peptide, the results and a comparison with molecular dynamics simulations. The last section draws the conclusion.

2 Implementation

The basic needs of a GA implementation are the string representation and the fitness function. The internal coordinates are used to represent the molecule. In that, we fix the bond-lengths and bond-angles and vary only the dihedral anlges (ϕ, ψ). Binary strings are arranged in the order $(\phi_1, \psi_1), (\phi_2, \psi_2), \ldots, (\phi_n, \psi_n)$ for an n-peptide (each angle represented by a 9-bit binary string). Energy of the molecule is used as the objective function which is given by a sum of non-bond, electrostatic and torsion contributions. More details of simple GA implementation has been presented in [7, 8].

A simulation requires to go through the iterations of selection, cross-over and mutation on a population of conformations represented by the binary strings. A size of 100 is used in the case of an octa-peptide. The probability of cross-over is varied in the range of 0.5 to 0.7 and that of mutation in the range of 0.001 to 0.01. About 200 iterations are found to be enough for the octa-peptide. The simulations are performed a number of times and ensured statistically that what is achieved at the end is an optimal solution and not just a local minimum. The validation of the simulation has been done considering a di-peptide of *Ala* which converged to $\phi - \psi$ angles that are well within the allowed regions [8] starting from random values.

In the iterative process of GA the most time consuming part of computation is the calculation of fitness. So, as a first step towards parallelisation we have implemented a data parallel model that distributes the calculation of fitnesses to different processors. Further, we have also implemented a parallel GA model called *migration* model. In this section we describe these models and give their performance on PARAM machines.

PARAM is a series of supercomputers developed by Centre for Development of Advanced Computing (C-DAC) implementing the openframe architecture. The first in the series is PARAM 9000/SS based on Sun super sparc (SS) processors. The second is PARAM 9000/AA based on DEC Alpha Processors. The recent one is called PARAM Openframe based on SUN Ultra Sparc-2 dual CPUs. The architecture is built around a multistage interconnect network which uses a packet switching wormhole router as a basic element. Each switch is capable of establishing 32 simultaneous non-blocking connections to provide a sustainable bandwidth of 320 Mb/sec. The communication links has built-in support for standard environments like MPI and PVM. The architecture allows the parallel machine to be viewed as an ensemble of independent workstations or as an MPP connected by a high bandwidth network. The compiler used in the present work is Fortran 77 with Message passing interface (MPI) standard. The code is portable to any other parallel machine.

2.1 Data Parallel Model

The Data parallel implementation of GAs is essentially parallelising the loop over the population size to calculate the fitnesses. The master-slave concept is used. The required communication calls are dumped on the subroutines called *broadcast* and *collect*. The former is used for broadcasting the required data like $\phi - \psi$ angles, chromosome length, population size, *etc.* to all the slave processes by the master. The latter is used to calculate the fitnesses on slaves as well as master and collect them back at the master process. The master process proceeds to the next generation of selection, cross-over and mutation. The speedup over the number of processors for PARAM machines are given in the next section.

2.2 Migration Model

In the migration model, several islands of subpopulations evolve simultaneously with one island mapped on each processor. That is, each processor simultaneously initiates an evolution process using GA constituting an island. Population on each processor could be treated as subpopulation. Periodically, say after every n iterations, each processor

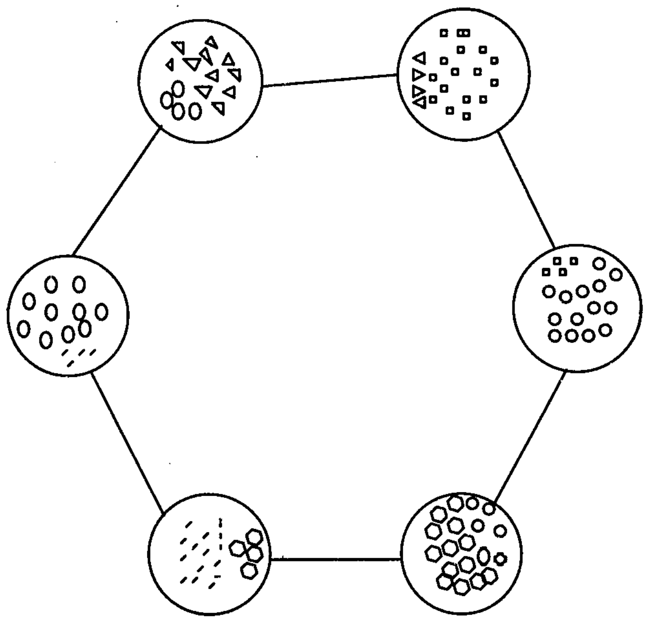

Figure 1: Migration model showing 6 processors as circles. Each of them start from different random seeds indicated by the shapes inside the circles. A part of them are different and indicates the migrated ones from its neighbouring island.

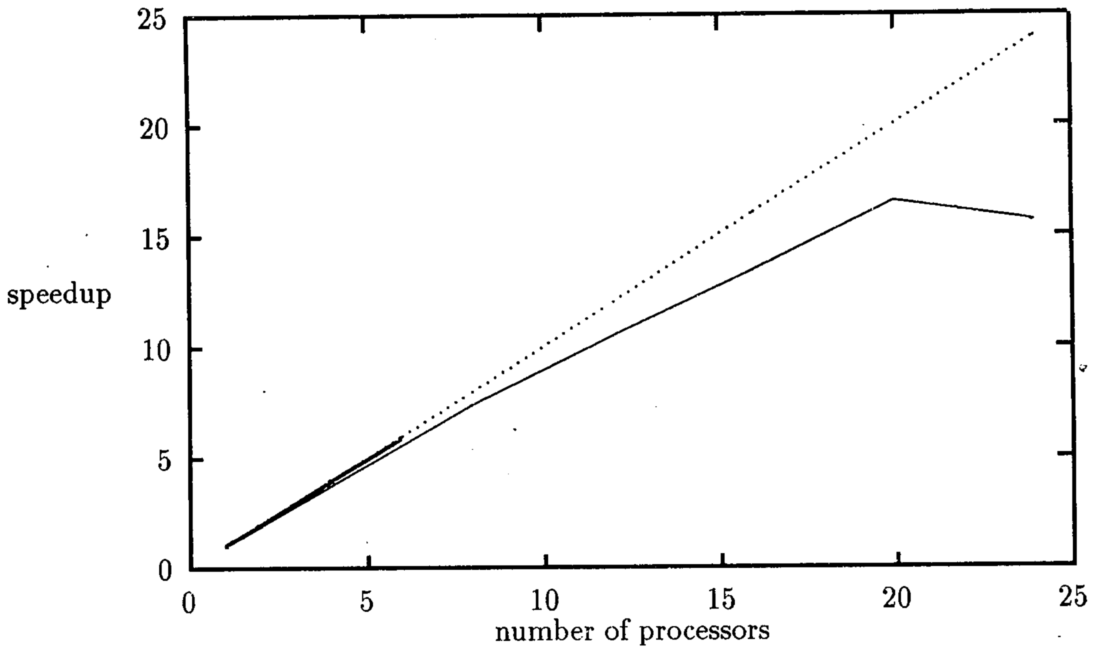

Figure 2: Performance (speedup) on PARAM 9000/SS (dashed line) and AA (continuous line) for the data parallel model. The dotted line indicates the ideal speedup expected.

would send its $x\%$ of conformations to its predecessor. Then, each processor would replace their individuals by the ones received from the neighbour.

Figure 1 shows a pictorial representation of a set of processors by circles. The pattern in each circle is distinct, indicating that they have populations starting from different random seeds. The migration process is indicated by a smaller number of symbols similar to that of its neighbour. The value of x and the method of choosing the individuals for migration as well as replacement could be treated in a variety of ways. This kind of distributed GA has several advantages.

1. A single run in parallel is equivalent to a set of runs on a sequential processor. Normally, GA is run with different starting points and solution, whichever is the best out of all these runs, is picked up. So, there is a definite reduction in computation time.

2. It brings in more diversity because of the additional operation of migration which may help in realising the optimal result faster than sequential GA. Thus the efficiency is enhanced as well as the search space is explored more efficiently by avoiding possible stagnation.

3. There are a number of ways by which one can exploit the distributed GA by the variation of % of migration, the number of migrants as well as the pattern of migration.

4. Communication overheads is minimal hence, an ideal candidate for parallel processing.

This kind of model has been used by Muhlenbein *et al* [14] and also discussed by Dorigo and Maniezzo [15]. Mansour and Fox [16] have compared their implementations of parallel simulated annealing, parallel neural network and parallel GA on an optimisation problem of data allocation. Though slow, parallel GA produced the best results. They used the shifting balance theory for distributed GA in which, weakest individuals are replaced with the fittest migrants from the neighbourhood. Instead, in the current implementation weakest few are replaced by the fittest few in the next node irrespective of their relative fitnesses. This in our opinion would maintain the diversity depending on how frequently the migration is performed as well as how many are migrated. More details on such studies are presented in the next section. In the migration model, one cannot compare the performances the way it is done for data parallel model. However, it is possible to highlight the benefits of such algorithmic parallelisation.

3 Results

This section is devoted to the results obtained using the GA for an homo-octa-peptide of Alanine. Ala_8 is a test for GA to check whether it can evolve a helical turn from a random set of torsion angles. It is well known that a chain of *Ala* forms a helical structure. About 5 simulations each of about 200 iterations were performed. Each time the random seed is changed to have a different starting populations for the GA. The final population of torsion angles are found to have most of them lying within the allowed region. In addition, a check is made so that there are no short-contacts among the atoms in the molecule [7].

The relative speedups on two of the PARAM machines (SS & AA) have been shown in Figure 2. Linear speedup has been observed on the PARAM 9000/AA (continuous line) in which a gigabit network has been used. On PARAM 9000/SS (dashed line) the speedup degrades beyond 16 processors. The ratio in speedup of single SS and AA node is 1:2.5 for this application. For a similar model maximum speedup of 430 has been reported by Hermann and Suhai [17] on MASPAR having 16384 processors for bigger problem sizes in the range of 36 to 56 amino acids. From the figure it is evident that, linear speedup is realised for about 8 processors beyond which the communication overheads start increasing. For the case of 24 processors, the efficiency drops to about 65%. There are two aspects responsible for this. All the processes have to wait at the end of each iteration, firstly in order to communicate the fitnesses and strings to the master process and secondly for the master process to complete the selection and genetic operations. Overlapping of communication and computation by allowing the master to do only selection and genetic

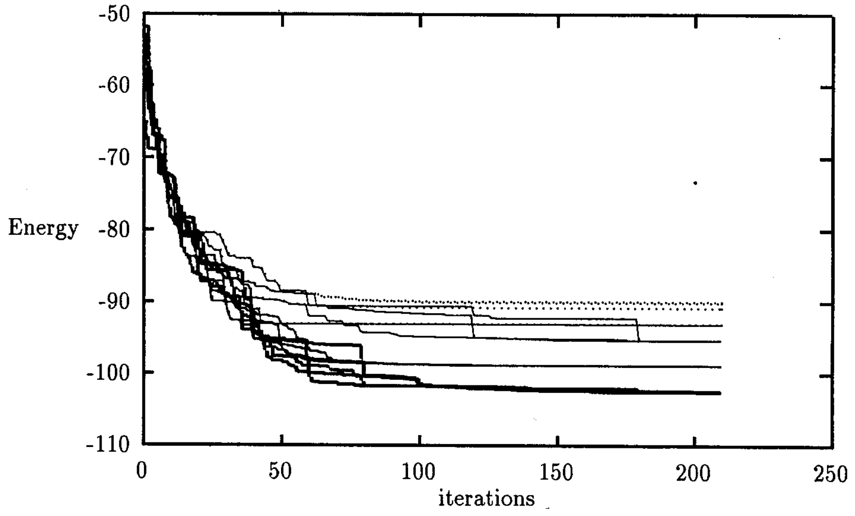

Figure 3: Performance plot in the migration model with 4 processors. The dotted curves are for the case of GA-islands working concurrently without migration. The continuous curves are for the case where there is a 10% migration after every 20 iterations. Small dashed curves for the case of migration after every 10 iterations and long dashed curves for the case of migration after every 60 iterations.

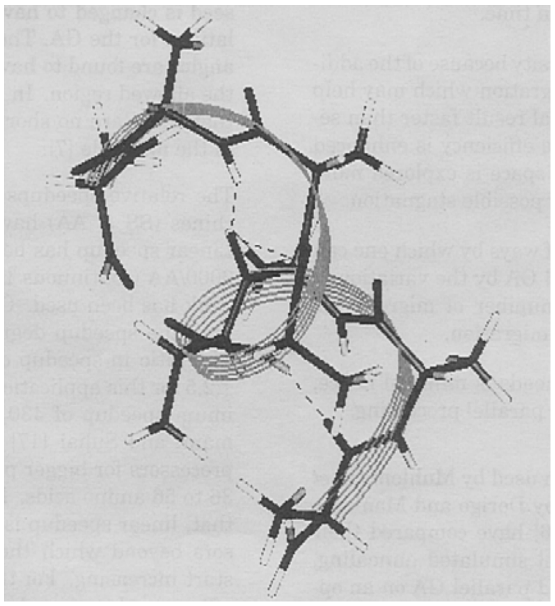

Figure 4: Three dimensional structure of the homo-octa-peptide. The best obtained by the migration model on 6 processors along with ribbon as a guide to the eye for helical turn. Dashed lines indicate the establishment of hydrogen bonds. This graphical display was printed out from the InsightII molecular modeling system of Biosym Technologies Inc.

operations and the slaves to do fitness evaluations may reduce the overheads. Moreover, right now the MPI implementation is on top of a layer of TCP/IP which makes the communication more time consuming. MPI implementation under light weight protocols is in progress in order to reduce to the communication overheads.

3.1 Migration Model

First, experiments were conducted on 4 processors for the homo-octa-peptide. In the migration model one has to decide on two quantities *viz.* n and x. Number of migrants is chosen as 5% and frequency of migration is once in every tenth iteration. In this, minimum obtained in ths distributed GA is not lower than the sequential one. This could be because of biasing due to frequent migration. Then the number of migrants has been increased to 10% and simulations were carried out for migrations at every 10, 20, 40 and 60 iterations, as well as without migration. Out of these, the migration at every 20 iterations yielded the best minimum. The performance plots for each of these are given in Figure 3 except for the case of 40.

When, the frequency of migration is increased to every 10 iterations, all the islands tend to converge to the same sub-optimal result. This amounts to an elitist-like model. On the other hand, when we decrease the frequency, *i.e.* migration at every 40 or 60 iterations, the diversity seems to persist and the individual islands converge to different sub-optimal solutions. Thus, increase in number of processors may also help in maintaining the diversity which in turn depends on the problem size.

In order to check this we performed the simulations on 6 processors with migration frequency of every 50th iteration. This yielded better results (Figure 1 in [7]). Here, most of the islands at least 3 out of 6 performed better than that in a concurrent run without migration having same starting points. The improvement observed in energy upto 10% is crucial for large molecules because of the magnitude involved. A further study on the effect of variation of n, x and the number of processors would provide more insight into the performance of the migration model.

The three-dimensional structure of the best conformation for the homo-octa-peptide is shown in Figure 4. The hydrogen bonds formation are shown by dashed lines. For poly-peptides of *Ala* every 3.2 residues would have a helical turn. This is clearly seen through the stick model guided by ribbon exhibiting two turns in the figure. Hermann and Suhai [6] have also obtained a single turn for a tetra-alanine. These results demonstrate one of the important expectations that a homo-octa-alanine would show helical structure.

In order to compare the results with other techniques, molecular dynamics (MD) simulations on the octa-peptide have been performed with different starting structures. Out of the six simulations, 3 of them could get only single helical turn and have RMS deviations from a perfect helical conformation by at least 3.8 *Angstroms* compared to the 2.05 *Angstroms* obtained using GA for the structure shown in Figure 4. The MD simulations were done on IRIS workstation using the DISCOVER module, a commercial package of Biosym Technologies Inc. The MD simulations had a schedule of minimisation using conjugate gradients, 5 to 10 *ps* of equilibration followed by 10 to 150 *ps* of annealing from 600 to 300 *K* and then minimisation. The consumption of time varied from about 300 to 3200 *secs*. Considering the fact that MD simulations have more physical appeal, a better approach would be is to use GA at the initial stages, when the possible conformations are totally unknown, and switch over to MD and calculus based minimisers to arrive at the final structure. A study of such a switching process would help in exploiting the power of each technique.

4 Conclusion

A GA based method has been presented to optimise the structure of an octa-peptide. The algorithm gives encouraging results in the sense that, we are able to start from a population of random conformations and evolve the global minimum through the processes of selection and genetic operations viz. cross-over and mutation. We used a binary string to represent the (ϕ, ψ) angles of the individuals in the population. The code was parallelised first to distribute the calculation of fitness values to different processors of a parallel machine PARAM. This gives a near linear speedup. Algorithmic parallelisation has been implemented using the *migration* model and found to give performance better than the sequential one by about 10% which may be crucial for larger molecules. Further work on the analysis of the data parallel and migration models is expected to give more insight into the GA as well as its exploitation on parallel processing platforms. The comparison with MD simulations seems to be favourable to GA but, more study is required to establish any conclusion. Work on one more method of parallelisation, fine-grain model, is in progress. In this model, a single population evolves; each individual placed in a cell of a planar grid; selection and cross-over are applied only between neighbouring individuals. The work on the structure prediction for a bigger molecule, yet one of the smallest proteins brovine pancreatic trypsine inhibitor (BPTI) turned out to be a tough testbed for GA. Results of that are to be published soon.

Acknowledgments The authors would like to thank Dr. Kalyanmoy Deb and Dr. Devapriya Choudhury for many fruitful and exciting discussions.

References

[1] David E. Goldberg.,"Genetic algorithms in search, optimisation, and machine learning" *NY Addison-Wesley*, 1989.

[2] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi.,"Optimisation by Simulated Annealing" *Science* **220**, p671, 1983.

[3] Shaojian Sun., "Reduced representation model of protein structure prediction: Statistical potential and genetic algorithms" *Protein Science*, 32:762, 1993.

[4] Schulz-Kremer., "Parallel Genetic Algorithms" Ed. by J. Stender, p129, *IOS Press*, 1993.

[5] Frank Hermann and Sandor Suhai., "Energy minimisation of peptide analogues using genetic algorithms", *Journal of Computational Chemistry*, 16:1434, 1995.

[6] V. Sundararajan and A.S. Kolaskar., "Parallel Genetic Algorithms on PARAM for Conformational Search" *Third International conference on High Performance Computing*, IEEE Computer Society, p22, 1996;

[7] V. Sundararajan and A.S. Kolaskar., "Distributed Genetic Algorithms on PARAM for Conformational Search", *A Computational Paradigm for Synthesizing Models and Simulations of Complex Biological Systems* Edited by S.S. Iyengar, CRC Press (in print).

[8] C.D. Lucasius and G. Kateman., "Application of Gentic Algorithms in Chemometrics", ICGA'89, p170, 1989.

[9] C. B. Lucasius and M. J. J. Blommers and L. M. C. Buydens and G. Kateman, "A Genetic Algorithm for Conformational Analysis of DNA", *Handbook of Genetic Algorithms*, edited by Lawrence Davis, Van Nostrand Reinhold, New York, p251, 1991.

[10] Bruce A. Shapiro and Joseph Navetta, "A Massively Parallel Genetic Algorithm for RNA Secondary Structure Prediction, ", *The Journal of Supercomputing*, 8:p195, (1994), 9:p29, 1995.

[11] Bruce A. Shapiro and Jin Chu Wu, "An annealing mutation operator in the genetic algorithms for RNA folding", *CABIOS 12 (3)* p171, 1996.

[12] Jan T. Pedersen and John Moult "Genetic Algorithms for Protein Structure Prediction", *Current Opinion in Structural Biology*, 6:227, 1996.

[13] Gorges-Schleuter M Muhlenbein H and Kramer O. "Evolution algorithms in combinatorial optimisation", *Parallel Computing*, 7:65, 1988.

[14] Dorigo M and Maniezzo V. "Parallel Genetic algorithms", Ed. by J. Stender, p 5. *IOS Press*, 1993.

[15] Mansour N and Fox G. C. "Parallel physical optimisation algorithms for allocating data to multicomputer nodes", *The Journal of Supercomputing*, 8:53, 1994.

[16] Frank Hermann and Sandor Suhai "Genetic Algorithms in Protein Structure Prediction", *Computational Methods in Genome Research* Edited by S. Suhai, *Plenum Press*, New York, p173, 1994.