International Journal of Computer Applications (0975 – 8887)
Volume 81 – No.11, November 2013

Prediction of Protein Structure using Parallel Genetic Algorithm

Jasdeep Singh Bhalla
Department of Computer Science,
Bharati Vidyapeeth's College of Engineering,
New Delhi, INDIA

Anmol Aggarwal
Department of Information Technology,
Bharati Vidyapeeth's College of Engineering,
New Delhi, INDIA

ABSTRACT
The Genetic Algorithms are generally used to draw a similarity between the Genetic mutation and Cross Over within populations from the field of biology. Genetic algorithms are highly and significantly parallel in nature and performance. These types of algorithms can be used to solve many other important problems such as the Graph Partitioning problem that deals with partitioning of graph, the famous Travelling salesman problems etc. Implementation of these algorithm shows a trade-off between Genetic search capable qualities and execution performance qualities. In this paper we worked in order to improvise the execution performance rate of algorithms, those particular implementations with lesser communications between populations are considered best and highly efficient. In this same direction, we tried to present an algorithm using discrete small subpopulation groups. Therefore, this particular implementation tries to reduce the quality of search of the algorithm. Thus, we tried to improve the quality of this type of search by having a centralized population system. Here, we analyzed some of the other alternatives for the implementation of these algorithms on distributed memory architectures in which centralized data can be significantly implemented. Prediction of tertiary protein structure is also presented in the paper as an example in which we tried to implement these alternatives of parallel algorithms on it. In the last section, we tried to summarize the performance analysis of the various proposed architectures.

Keywords
Genetic Algorithms, Protein Structure Prediction, Parallel Genetic Algorithms, Distributed Memory Architecture

1. INTRODUCTION
Genetic Algorithms is a search technique which is heuristic in nature and execution. This concept draws its inspiration from the anciently popular scheme "the survival of the fittest" of natural evolution in science and technological field. This logic was first pioneered by John Holland in the age of 1960s. Genetic Algorithms is the theory that has been widely studied and it is used in most of the fields. In genetic algorithms, a collection of possible solutions, known as population are maintained, monitored and executed. This type of search algorithm proceeds in steps generally referred as generations similar as in nature. Therefore, each generation mainly involves transformation of many individual solutions based on their individual fitness functions and results in a new population of solutions with different parameters associated with them. There are majorly two operators of transformation that are considered in genetic algorithms: crossover and mutation.

In cross over transformation, the individual pieces of solution are completely swapped to give new solutions and their fitness is tested upon for results. After finishing crossover, the concept of mutation is applied which involves random selection of new characteristics for the solution obtained. The termination is completely based on pre- defined number of steps or a pre-specified level of optimality for future reference as well. Genetic Algorithms significantly provides alternative methods for solving the problem and consistently performs other traditional methods efficiently without any errors. Most of the real world problems that involves finding of optimal parameter may seem to be difficult for traditional methods but are ideal for Genetic Algorithms in many cases. Although these are highly skilful in solving hard problems, they might also take longer execution times as compared to others. To counter this execution delay problem, scientists resorted to exploit the parallel nature of Genetic Algorithms. Thus, these came to be known as parallel genetic algorithms. Their implementation generally involves very similar issues as other parallel algorithms do which are explained in terms of granularity, synchronization and locality of reference etc. In addition to all these parameters, considerations about replication strategy, mutation, distribution of population and quality of search also must be addressed for better results. In this paper, we tend to address the issue of population distribution on distributed memory architectures. Thus, the distribution of population, improves the execution performance considerably. The main key point is to design independent processes which handle subpopulations of solutions individually. Implementations with least communications are always considered best, unique and significant. Other than this, there is a compromise with the quality of search in such implementations and results obtained. Here, we try to present certain alternatives for the parallel implementations that can be beneficial. These are characterized as centralized, partially distributed or fully distributed. Also to prove its application, we explain the quality of search and the execution performance of our strategies on the problem of predicting protein structure.

2. PARALLEL GENETIC ALGORITHM
Some strategies for distribution of population for parallel Genetic Algorithms are discussed in this section. Here, we considered and analysed the synchronized processes in which the processors run for a certain number of generations and then communicate significantly. In this case, Parallel Genetic Algorithms can be easily implemented and are efficient in situations where the processors perform sufficient work between communications. Various memory machines are discussed below:

2.1 Distributed memory machine

The Distributed memory here means a multiple processor (more than one processor) system in which each processor has its own private (distinct) memory on which operations (processes) can be performed. Most of the computational procedures can be processed only on local data, and if the need of remote data is initiated, the computational task must converse with single or additional remotely connected processors and process the operation respectively. For such type of communication some form of interconnection is allowed and preferred to some extent. There are some links between processors and these links can follow standard protocol.

2.2 Shared memory machines

In these, a number of multiprocessors have an access to a large memory which is different from private memory. Therefore, the processors need not be aware of where the data resides and where it should be located. But somehow, there are performance penalties and race conditions that need to be avoided in these types of implementations.

3. Use of Parallel Genetic Algorithm on Distributed Memory Architectures

Genetic Algorithms are highly parallel in nature as compared to other algorithms. Therefore, it is simple to implement these on distributed memory machines for efficient results. But, the distribution results in destruction of the quality of search results obtained in contrast to those obtained through centralized implementations of algorithms. This main trade-off between search quality and distribution can be resolved through semi-distributed implementations. All the processors have their own private memories in these cases. In addition to these, the processors are combined together in order to form clusters of data. Out Of this, each cluster has its own large shared memory. This type of strategy exhibits least contention and communication overhead problems. Although, it may suffer from execution overhead problem. It still draws the benefits of centralized implementation and result in obtaining better search results in fewer generations as compared to others. In the subsequent sections we described the different implementation strategies we studied and analysed.

3.1 Covenant Implementation

The Covenant Implementation is used to achieve the most challenging scenario distribution of population is a single large population. Covenant technique solves the issue by creating an agreement between the kernel processor and slave processor. The arrangement is done in form of star topology where only master gets power to access the memory. The role of master is to gather information from the memory and run replication algorithm. After this, the master sends the couple solutions to the slaves. The slaves need to transform them using the genetic operators and then test their validity using the fitness functions. The transformed form is sent back to master.[1] The challenges faced by this mechanism are the large amount of communication, a fraction is sequential computation (replication) and the granularity is too fine. Such complications can be reduced by making the slaves work on several generations. Another issue is the time when master remains idle, while slaves are working. This problem can be solved by making the master work on different subset of population. After master is done with his work, the slaves can send back subpopulations to master. And the process continues like a handshake.

3.2 Semi-Disparate implementations

This strategy aims at settling for an intermediate level of distribution of population. It benefits from the global knowledge of current population and its distribution. In this strategy, we have clusters of processors, each having its own kernel which is arranged in a torus topology. After certain intervals of time, all the kernels communicate with each other to exchange some of their best individuals. The communication done is synchronized, to prevent contention in the network. This implementation succeeds in reducing the overhead of maintaining a single master but suffers from the problem of large amount of inter-processor communication and semi distributed replication.

3.3 Disparate implementations

In this scheme all nodes work independently of each other. Every node has the total population of solutions and is free from shared population. Every node runs the complete sequential Genetic Algorithm on it independently. All the nodes are connected in torus topology. After certain intervals of time, all the kernels communicate with each other to exchange some of their best individuals. The communication happens during initialization (to get a description of [2] problem to be solved) and termination (in which all processors send their best individuals to the processor responsible for displaying the results). This implementation is beneficial as there is no contention among processors and all have their own population set also the amount of communication can be made as minimum as desired. However a lot of time is wasted because of computations done on locally fit solutions.

3.4 Totally disparate implementations

This is the other extreme followed in the distribution of population. In this strategy, the processors don't exchange their individuals. Communication takes place only at initialization and termination of the algorithm. Thus, there is absolutely no contention and the communication is even lesser.

4. Prediction Of Protein Structure Using Genetic Algorithms

Proteins are classified as complex molecules that can be broken down into sequences of amino acids bounded by a peptide bond, which plays the main role in nearly all biological processes, for instance, immune response mechanisms, signal transduction etc. Tertiary structure is a 3 dimensional structure, [20] native structure of a single polypeptide or protein, set in such a way that the hydrophobic side chains are held internally and the hydrophilic groups are held externally, providing steadiness to the molecule. This tertiary structure assembles the different secondary structure components by describing the folding of the polypeptide chain in a protein structure arrangement. Thus, the domain field is the entity of tertiary structure. The Multi-domain based proteins, which can be built from several other domains, [17][3] may represent their arrangement within each chain and relative to each other. However, these can also be separated into numerous classes on the basis of their size, their physical and other chemical characteristics. The chief categorization is done into hydrophobic residues, referring to its fragile (or weak) interaction to solvent water molecules, and hydrophilic based residues, due to its ability in construction of hydrogen bonds along with water (H_2O). [16] Therefore, every amino acid constitutes a common main-chain part, including the atoms N, C, O, Ca, two hydrogen atoms and a definite side chain. These amino acids are united by the peptide bond, the planar CO– NH cluster. The foremost (or main) degrees of freedom in forming the three Dimensional (3D) trace of the polypeptide chain are the two dihedral angles, x and y on each side of the Ca atom. [15] The side chains branch out of the main chain from the Ca atom and have supplementary degrees of freedom enabling them to adjust their local conformation to their surroundings. Protein structure prediction constitutes its importance which generally lies in its wide presence in living organisms, usage of the computational protein structure prediction directed to computer assisted drug design and computer assisted molecular design. Use of a near-optimal meta-heuristic in nature, such as a Genetic Algorithm, which is one of the most significant and promising optimization methods as it explores minimal figure of potential structures in this field. [4] A more prevailing exploration of the conformational space can be accomplished by using these population-based meta-heuristics. Thus, given the primary structure of a specified protein, we call for determining its ground state conformation. However, they have limited search intensification capacities.

5. Existing Work

Traditionally, two search methods have been employed to estimate the tertiary structure of a protein from its primary (main) structure. These two are referred to as: Molecular Dynamics (MD) and Monte Carlo (MC).

Molecular Dynamics takes into account, the system's reaction to the forces the atoms on each other, thus assuming that atoms move in a Newtonian manner on the basis of theory of molecular dynamics [7]. As Molecular Dynamics methodologies are based on the direct (straight) simulation of the natural folding processes, the Monte Carlo algorithms are based on minimization of an energy function, through a path that does not essentially follow the natural folding pathway set primarily.

Monte Carlo method calculates the free energy of successive small conformational step which is then accepted if the free energy is reduced compared to the previous conformation. Therefore, the Molecular Dynamics methods almost by definition require a full atomic model [5] of the protein and a detailed energy function, but Monte Carlo methods can be used both on detailed models as well as on simplified models of proteins.

6. Proposed Model

The Monte Carlo algorithms, although efficient, are more likely to get caught in a local minimum. Therefore, use of Genetic Algorithms to tackle the protein folding trouble can be more effectual and significant. While folding a chain with a Monte Carlo algorithm (MC algo) based typically on altering a single amino acid, it may be general to get into a state where every single amendment is rejected because of a significant increase in free energy, and thus, only a concurrent change of several angles might enable further energy minimization. This type of concurrent change is provided naturally by the crossover operator of the Genetic Algorithm.

To employ a genetic algorithm, it is obligatory to encode the variables of the optimization problem (complexity) into the genes. These genes of the parents are then operated on through recombination and mutation to construct the genes of the children.

6.1 Selection Operator

The selection operator tries to use a fitness function to identify the test individuals of the existing population that will supply as parents for the next upcoming generations. Fitness value of every individual is specified by a problem-specific function. Thus, the selection mechanism guarantees that the finest individuals have a higher probability to be selected to reproduce to form a novel generation.

6.2 Fitness Function used in selection

Various numerous energy functions have been used as a part of the various Genetic Algorithm based protein structure prediction protocols, which ranges from the hydrophobic potential in the simple HP lattice model to energy models such as CHARMM (Chemistry at Harvard Molecular Mechanics), based on complete and detailed molecular mechanics. Thus, the fitness function here can be easily modified to include terms that are not used in the traditional methods of protein structure prediction.

6.3 Crossover

Crossover tends to randomly select pairs of crossing points and exchanges substrings between them to produce new off-springs. Thus, this primary exploration mechanism for Genetic Algorithms is crossover. An example of this operation (crossover) is that a branch is selected for crossover in each of the parent trees. In this particular example, the primary left branch from the primary parent is replaced with the primary right branch of the secondary parent. Therefore, this forms two new trees.

6.4 Mutation

The mutation operator is typically taken as a secondary operator. It is used mostly for restoring the diversity that may be lost from the recurring application of selection and crossover. Nevertheless, this operator merely takes one string from the population and randomly alters some value within it.

6.5 Algorithm: Protein Structure Prediction

```
1. Select Initial Population and consider it as G(0)
2. Put, t ←0;
3. Repeat loop while n(G(t))>1(Number of offspring's >1)
4. Evaluate the value (P(t));
5. Apply the fitness function on G(t) as G(t) ← Selection(G(t));
6. Alter the various reproduction operators and apply the chosen operator to obtain the new population G(t+1): G (t+1) ← Apply Reproduction Operator(G(t));
7. G(t + 1) ← G(t)
8. Change, t = t + 1;
9. End loop in started in Step 3
```

In this above algorithm, reproduction operators are applied at the initial population to yield the first set of offspring's rather than other steps. Then, every new set of offspring's is iteratively tested by the fitness function (see Step 5) and if the offspring's are closer to the solution, they try to undergo further reproduction operation (see Step 6). Thus, note that any of the reproduction operators may be applied to the offspring's. These offspring's, G (t+1) then serves as our new population G(t) (Steps 7 and step 8). A problem specific fitness function must be applied for evaluation.

7. Experimental Results

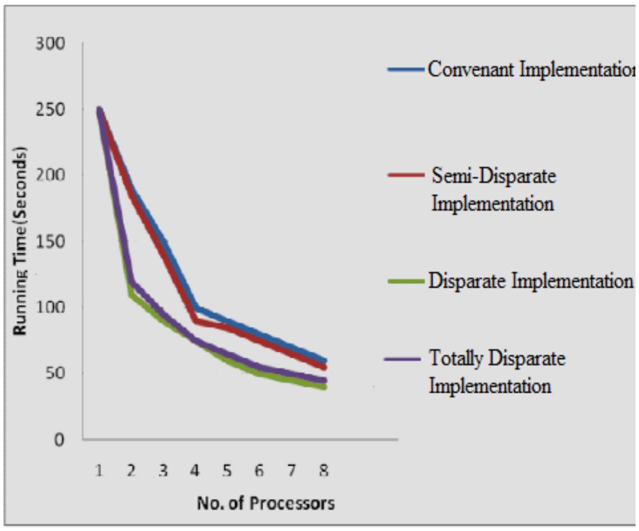

Fig. 1. **Experimental Results**

In Fig 1, we assessed the performance of the above mentioned algorithm on every of the planned architectures and illustrated the results (outcome) as a graph amongst the running time and the count of processors used in the system model. It can be inferred from the above graph model that the running time in an incongruent implementation is significantly reduced, and therefore, the performance (presentation) of this implementation is enhanced than a covenant performance. Thus, there exists a trade-off amongst the performance and quality of search. Selection (choice) of the architecture is reliant in the application model. But if Genetic Algorithm is being used to forecast a serious problem where search quality (value) is a critical aspect, it would be better to prefer a covenant implementation despite its slow running time.

8. CONCLUSION

The protein structure prediction based problem is in particular appropriate for the Genetic Algorithms, as these algorithms have produced significantly better results as contrasted to the Monte Carlo (MC) method (technique). Therefore, the ability of the parallel genetic algorithms in describing numerous biological processes arrives from its exceptional ability to model cooperative pathways, which makes it better matched to predict protein structure in the respective field of study. The implementations in the form of disparate and semi disparate implementations have been proved to be more effective and of better time complexity. Thus, making the protein structure prediction even better suited concept for distributed memory architecture. Parallel Genetic Algorithms model the methods that retrieve an optimum value in an iterative manner. The prediction of Protein structure serves as an excellent example in this domain.

9. FUTURE SCOPE

We used genetic algorithms to predict the protein structure. In future some other problems can be analysed using genetic algorithm. As there is no end for generation building thus, there is no end for research on genetic algorithms. More new methods for dealing with these algorithms could be developed and analysed. We considered parallel genetic algorithm over distributed memory architectures. Thus, our proposed methodology can be improvised in terms of algorithm, execution time and uniqueness.

10. REFERENCES

[1] R. Tanese. Parallel Genetic Algorithm for a hypercube.in proc of 2nd int. conference on Genetic Algorithms and its applications,1987 Volkan Kurt, Protein Structure Prediction Using Decision Lists, 2005.

[2] ALFRED A. RABOW AND HAROLD A. SCHERAGA, Improved genetic algorithm for the protein folding problem by use of a Cartesian combination operator.

[3] Akshay Gupta, Sunil Kr. Singh, Khushboo Aggarwal, “Performance of Parallel Genetic Algorithms on Distributed Memory Architectures”, Journal of Engineering Research and Studies, E-ISSN 0976-7916, JERS/Vol.II/ Issue I/January-March 2011/10-17.

[4] B. Manderick, P. Speissens, Fine Grained Parallel Genetic Algorithms, In proc of 3rd int. conference on Genetiv Algorithms and its applications pages 428-433,in 1989.

[5] Jane S Richardson, The Anatomy & Taxonomy of Protein Structure Joseph D. Szustakowski and Zhiping Weng, Protein Structure Alignment Using a Genetic Algorithm.

[6] Ricardo Bianchini, Christopher Brown, Parallel Genetic Algorithms on Distributed Memory Architecture.

[7] Richard Dallaway, Genetic programming and cognitive models.

[8] Ron Unger, The Genetic Algorithm Approach to Protein Structure Prediction Ron Unger, The Genetic Algorithm Approach to Protein Structure Prediction.

[9] J.Y Suh and D.V. Gutch.Distributed Genetic algorithms.Technical Report 225,Indiana University,Computer Science Department,1987 SHAOJIAN SUN, Reduced representation model ofprotein structure prediction Statistical potential and genetic algorithms.

[10] Sunil Kr. Singh, Khushboo Aggarwal, Akshay Gupta, In proc of Int. conference on Innovative Practices in Management & Information technology for Excellence at MAIMT, India, page 133-140.

[11] Steffen Schulze-Kremer, Genetic Algorithms for protein tertiary structure.

[12] S. Cahon, N. Melab, E.-G. Talbi, An enabling framework for parallel optimization on the computational grid, in: Proc. 5th IEEE/ACM Intl. Symposium onClusterComputing and the Grid,CCGRID’2005,Cardiff, UK, 9–12 May, 2005.

[13] E.-G. Talbi, A taxonomy of hybrid metaheuristics,Journal of Heuristics 8 (2002) 541–564.

[14] E. Alba, G. Luque, E.-G. Talbi, N. Melab, in: E. Alba (Ed.), Metaheuristics and Parallelism,John Wiley and Sons, 2005.

[15] S.Cahon,N.Melab,E.-G.Talbi,ParadisEO:Aframework forthe reusable design of parallel and distributed metaheuristics,Journal of Heuristics 10 (2004) 357–380.

[16] B. Parent, A. Kok¨ osy, ¨ D. Horvath, Optimized evolutionary strategies in conformationalsampling,Journal of SoftComputing (2006).

[17] C. Levinthal, How to fold graciously, in: J.T.P. DeBrunner, E. Munck (Eds.), Mossbauer Spectroscopy in Biological Systems (Proceedings of a Meeting Held at Allerton House, Monticello, Illinois), University of Illinois Press, 1969, pp. 22–24.

[18] J.D. Knowles, D.W. Corne, Reducing local optima in single-objective problems by multi-objectivization, in: E. Zitzler, et al. (Eds.), Proc. First International Conference on Evolutionary Multi-criterion Optimization, EMO'01, Springer,Berlin, 2001, pp. 269–283.

[19] B. Ma, S. Kumar, C.-J. Tsai, R. Nussinov, Folding funnels and binding mechanisms, Protein Engineering 12, 713–720.

[20] J.J. Grefensttete, Parallel Adaptive Algorithms for function Optimization, Technical reportCS-81- 19, Nashville, Vanderbilt University.