Competitive Template Analysis of the Fast Messy Genetic Algorithm When Applied to the Protein Structure Prediction Problem

R. Day, J. Zydallis, and G. Lamont

Department of Electrical and Computer Engineering
Graduate School of Engineering and Management
Air Force Institute of Technology, Wright-Patterson AFB, OH 45433, USA
(Richard.Day, Jesse.Zydallis, Gary.Lamont)@afit.edu

ABSTRACT

[1]The Protein Structure Prediction (PSP) problem is a Grand Challenge problem among biochemists, computer scientists and engineers alike. Solving this problem involves correctly predicting the geometrical conformation of a fully folded protein. This paper focuses on CHARMm energy minimization and the use of a genetic algorithm, the fast messy genetic algorithm (fmGA), to obtain solutions to this optimization problem. The fmGA is a novel algorithm that explicitly manipulates building blocks (BBs) in order to obtain "good" solutions to an optimization problem. In order to obtain these "good" solutions, fully specified competitive templates are used within the fmGA to evaluate the BBs found. This paper presents "good" results of an analysis of various competitive template schemes for the application of the fmGA to the PSP of [Met]-Enkephelin and the much larger Polyalanine peptide.

Keywords: Energy Minimization, Protein Structure Prediction Problem, Genetic Algorithm, Fast Messy Genetic Algorithm.

1 INTRODUCTION

Scientists and engineers have been working on the Protein Structure Prediction (PSP) problem for a number of years with no "optimal" method having been found to solve this NPC problem [4]. The objective of developing a generalized technique for predicting a polypeptide's molecular structure given its amino acid sequence is commonly referred to as the PSP problem. The problem requires the minimization of an energy function in conformational space [1]. This paper focuses on an energy minimization technique and an associated stochastic search technique, a GA approach.

PSP energy minimization techniques use quantum mechanical calculations to find a protein's energy level, also referred to as a protein's "fitness". Finding the lowest global energy level is thought to lead to discovering the protein's ideal structure or its fully folded geometric conformation. Unfortunately, the search space for a total enumeration of even a small protein containing a few amino acids is extremely large and of exponential nature. Thus, a reasonable stochastic search technique using a Genetic Algorithm (GA) that achieves a "good" solution is suggested and employed here.

Our contemporary research developed a generalized technique, the fast messy GA (fmGA), to predict a polypeptide's molecular structure given its amino acid sequence. Our previous research with the fmGA in finding the minimum energy using the empirical CHARMm energy model and generation of the associated protein structure is now extended to include various innovative techniques to improve the effectiveness of the fmGA [5]–[7]. The two proteins analyzed here are [Met]-Enkephelin, which consists of 5 residues and 24 dihedral angles (Tyr-Gly-Gly-Phe-Met amino acids), and Polyalanine which consists of 14 residues and 56 dihedral angles ($ALA_1, \ldots,$ ALA_{14} amino acids). Each of the dihedral angles is represented by a binary string of 10 bits yielding a landscape size of 24^{1024} and 56^{1024} respectively.

The work presented focuses on building "better" CTs through the use of multiple CT approaches to include a memetic approach and seeding the CTs with known protein secondary structures. The CT is a key to finding "good" solutions in the fmGA. By focusing on modifying the process that the fmGA uses to create and update the CT during the execution of the algorithm the algorithm's effectiveness maybe increased. Previous work has shown the fmGA to obtain favorable results when applied to [Met]-Enkephelin and Polyalanine [6], [7]. Here we focus on improving the effectiveness of the fmGA through modifications to the CT generation process. Four new dynamic approaches are suggested and the results are analyzed in an attempt to find the geometric conformation of polypeptides. Statistical results are presented. In the next section a description of the fmGA is presented, followed by the four new CT approaches, testing and results and the conclusions.

2 FAST MESSY GENETIC ALGORITHM

The fmGA is a binary, stochastic, variable string length, population based approach to solving optimization problems. The fmGA was developed by Goldberg,

[1]The views expressed in this article are those of the authors and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

Deb and Kargupta [2] and later applied to the PSP problem by Merkle, Gates, Lamont and Pachter [3]. The main difference between the fmGA and other genetic approaches is the ability of the fmGA to explicitly manipulate building blocks (BBs) of genetic material in order to obtain "good" solutions and potentially the global optimum. The fmGA contains three phases of operation: *the initialization phase, the building block filtering (BBF) phase, and the juxtapositional phase*, which includes various parameters.

In the initialization phase of the fmGA, a population sizing equation is used to derive a population large enough to overcome the noise present in the BBF process. Once the population size is determined, initial population members are randomly generated and their corresponding fitness values are calculated through the use of the CHARMm energy model. These population members are referred to as fully specified since all of the associated genes of the population member contain allelic values.

The fully specified population members from the initialization phase are then systematically reduced in length to the user specified BB size through the use of a BBF schedule. The BBF process randomly deletes a certain number of bits from each population member over a number of generations specified in the schedule. This deletion of bits is alternated with tournament selection so that only the "best" partial strings are kept for processing in the subsequent generations. A CT is used at this stage to evaluate the underspecified population members. At the end of the BBF process, the entire population consists of underspecified strings of the user specified BB length.

The juxtapositional phase takes the "good" BBs found from the BBF process and combines them together through a cut-and-splice operator. This operator randomly chooses two strings and based on the probabilities of cut and splice, cuts the strings and splices them together accomplishing the goal of crossing over information between the strings. This process is also alternated with tournament selection so that only the best strings are kept from generation to generation. At the conclusion of this phase, fully specified strings exist in the population and the next BB size is evaluated via an outer loop over these three phases.

3 COMPETITIVE TEMPLATE GENERATION

Competitive templates are an extremely important part of the fmGA. Population members containing very few specified bits (underspecified members) with respect to the overall string length, as is the case at the end of the BBF process, are highly dependent on the CT. The reverse holds for strings that have the majority of their bits specified, as they only need to take a few bits from the CT. This illustrates the importance of the CT in the overall execution of the fmGA, especially at the start of the juxtapositional phase in generation fitness values.

To evaluate an underspecified population member, the CT is copied into a temporary location and the bits that are specified in the population member replace the bits of the CT within this temporary location. Once this is accomplished, the temporary string is evaluated and the resulting fitness is associated with the underspecified population member. In the case of an overspecified population member, which may occur when the cut-and-splice procedure causes a member to have multiple occurrences of a particular loci, a left-to-right method is employed. In this method, the first allelic value encountered for a particular loci is recorded as the value present for evaluation purposes.

Previous research is based on the concept of generating a random CT and periodically updating this template with the best found population member over the course of execution of the algorithm [5]–[7]. We propose using a more intelligent choice for the CT in order to increase the effectiveness of the fmGA.

A random CT is a natural starting point since the goal of the fmGA work is to generate a robust algorithm that obtains solutions for various optimization problems. In order to increase the effectiveness of the algorithm over this approach, the next step is to incorporate problem domain knowledge into the fmGA or increase the number of CTs utilized. The four CT methods suggested are:

1. Randomly generate a CT, then conduct a localized search on this CT. This memetic approach involves conducting a local search of the competitive template before each template update at the end of the juxtapositional phase.

2. The use of a fully specified population members containing specific structures as the CTs. Each seeded CT is hard coded into the fmGA using known alpha-helix and beta-sheet dihedral angles. The algorithm is expected to achieve better fitness values at a faster rate for proteins having either of these secondary structures through this method.

3. Utilizing a panmetic CT. The process generates an odd number of CTs and merges these into one template called a panmetic CT in this method. The merge is anticipated to take the best components of each of the CTs and combine them together. The CTs can be all random or a combination of random and those containing a structure.

4. Using more than a single CT developed via the aforementioned methods. This approach allows for more exploration since each population member is evaluated using multiple templates and there-

fore has the potential to find a better solution by searching different areas of the landscape.

4 TESTING, RESULTS AND ANALYSIS

Testing of the various CT approaches in the fmGA algorithm was accomplished on a 1.7 GHz Intel P-4 machine with 256 MB of RAM, using the Red Hat 7.1 distribution of the Linux operating system. The code was written in ANSI C. The fmGA algorithm was executed 10 times for each of the experiments in order to provide statistical results. All of the results presented here are averaged over 10 runs. Over all runs the following fmGA parameters were kept constant; cut probability = 0.02, splice probability = 1.00, primordial generations = 200, juxtapositional generations = 100, total generations = 300. An input schedule was used to specify sizes of the building blocks the algorithm uses and during which generations BBF occurs. Tests were conducted using both [Met]-Enkephelin, with 240 bit length strings and BB sizes 6-10, and Polyalanine, with 560 bit length strings and BB sizes 16-20.

The results for the four CT generation methods are plotted in figures 1 and 3, which presents the average for the [Met]-Enkephelin and Polyalanine data runs. The Alpha and Beta CT methods involve seeding the starting CT with a population member containing an α-helix and a β-sheet structure. The motivation for doing this is to provide a good starting point for the fmGA when applied to proteins containing this type of structure. Since [Met]-Enkephelin does not contain either of these types of protein structures, one would not expect to see any marked improvement with these CT choices over any other CT choice. In the case of Polyalanine one would expect the Alpha method to achieve better results as Polyalanine only contains an α-helix structure.

The Random CT randomly creates a population member and then conducts a local search on that population member to start with a locally optimized CT. This method has been used in the past and has provided good results [5]–[7]. The Alpha, Random & Beta Panmetic CT utilizes all three of the aforementioned CT generation methods to generate three CTs and then uses an exclusive OR operator to combine these three CTs into one panmetic CT. The motivation for this method is to combine portions of the three CTs for proteins containing a variety of structures within the one protein.

The last method of the Alpha, Random & Beta CT actually uses three CTs during execution of the algorithm. Each population member is evaluated with respect to each of the CTs and the "best" value found for that population member is thereby associated with that member. The motivation here is to combine portions of the three CTs for proteins containing a variety of structures within the one protein. In each of these three methods, the CT is updated at the conclusion of each BB size. At this point the best found population member becomes the new CT and the algorithm continues to execute with the next BB size and the new CT.

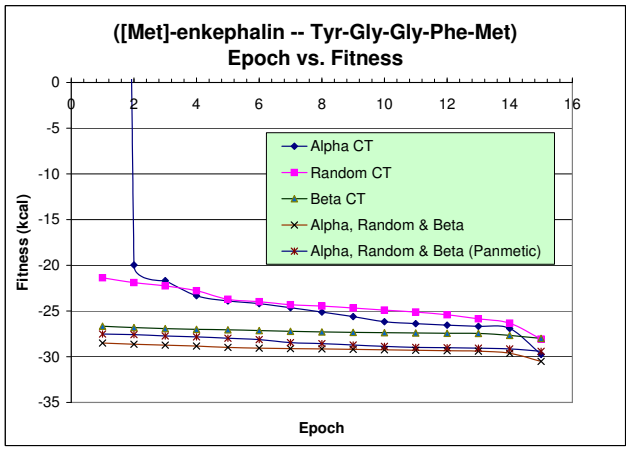

Figure 1: [Met]-Enkephelin Fitness Values

Figure 1 presents the average results of each CT method as applied to [Met]-Enkephelin. Standard deviations (not shown) for each of the CT choices overlap across all of the methods. Thus, one can conclude that there is no statistical difference between the 4 methods when applied to [Met]-Enkephelin. Figure 2 adds an important piece of information to this conclusion, that the multiple CT method, Alpha, Random & Beta CT, takes significantly more time than any of the other methods since three fitness evaluations must take place for each population member. The panmetic method also takes addition time over the random and seeded structure methods. The best choice for an efficiency versus effectiveness standpoint would be to utilize the random or the seeded structure methods as they obtain statistically the same results.

Figure 3 presents the results for the CT methods as applied to Polyalanine. The standard deviation of the various CT methods illustrates a statistical difference between the Random method as compared to the Alpha, Panmetic, and Alpha, Random & Beta CT methods. The last three methods achieve statistically similar results. These results are what one would expect since each of these last three methods contain an Alpha CT embedded in them and Polyalanine contains an α-helix structure. The best choice here would be the Alpha or Panmetic CT method since the same timing results are obtained with Polyalanine as are presented for [Met]-Enkephelin in figure 2.

In terms of the best fitness found for each method, this is presented in Table 1 and these results are very favorable when compared with results using other techniques. For [Met]-Enkephelin, our conclusion of using

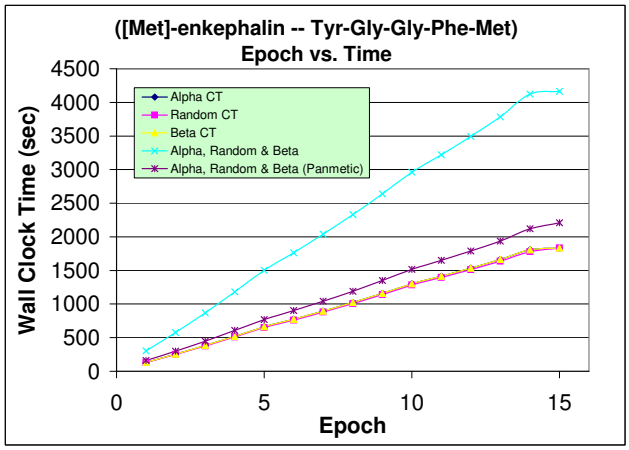

Figure 2: [Met]-Enkephelin Timing Values

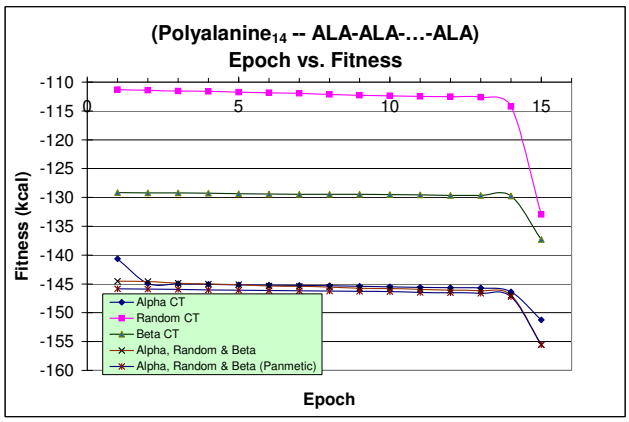

Figure 3: Polyalanine Fitness Values

the random or seeded methods is supported here considering that the Random and Beta methods obtained the two best results for this protein. For Polyalanine, our conclusion of using the Alpha or Panmetic CT methods again is supported here considering that the Alpha and Panmetic found two of the three best values with the Alpha, Random & Beta CT method slightly edging out the Alpha CT method.

5 CONCLUSIONS

We have presented the results for four different innovative CT generation schemes as applied to [Met]-Enkephelin and Polyalanine. The application of the fmGA to these proteins suggests the following generalized conclusions: The application of the fmGA to a protein containing no structure produced the best results either through the use of the Random or Seeded CT. The application of the fmGA to a protein containing a structure produced the best results with the Panmetic or the CT reflecting the structure of the protein. This is expected as those CTs contain some of the "good" BBs necessary to obtain "good" fitness values. Since the goal is to apply the fmGA to proteins of unknown structure, the panmetic or random CT methods show the most promise. Future work will look at larger proteins and enhancing the work described here with a multiobjective approach.

Table 1: Best Fitness Found

	Alpha	Beta	Random	Panmetic	A,R&B
Met	-31.716	-33.191	**-34.114**	-31.546	-31.834
Poly	-163.393	-157.203	-159.105	**-172.096**	-171.760

REFERENCES

[1] George H. Gates, Jr., Ruth Pachter, Laurence D. Merkle, and Gary B. Lamont. Parallel simple gas vs parallel fast messy gas for protein structure prediction. *Proceedings of the Intel Supercomputer Users' Group Users Conference*, 1995.

[2] David E. Goldberg, Kalyanmoy Deb, Hillol Kargupta, and Georges Harik. Rapid, accurate optimization of difficult problems using fast messy genetic algorithms. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 56–64, San Mateo, CA, July 1993. Morgan Kaufmann Publishers.

[3] Laurence D. Merkle, George H. Gates, Jr., Gary B. Lamont, and Ruth Pachter. Application of the parallel fast messy genetic algorithm to the protein structure prediction problem. *Proceedings of the Intel Supercomputer Users' Group Users Conference*, pages 189–195, 1994.

[4] Kenneth M. Merz and Scott M. Le Grand, editors. *The Protein Folding Problem and Tertiary Structure Prediction*. Springer, New York, 1994.

[5] Steven R. Michaud. Solving the Protein Structure Prediction Problem with Parallel Messy Genetic Algorithms. Master's thesis, Air Force Institute of Technology, Wright Patterson AFB, March 2001. AFIT/GCS/ENG/01M.

[6] Steven R. Michaud, Jesse B. Zydallis, Gary Lamont, and Ruth Pachter. Scaling a genetic algorithm to medium-sized peptides by detecting secondary structures with an analysis of building blocks. In Matthew Laudon and Bart Romanowicz, editors, *Proceedings of the First International Conference on Computational Nanoscience*, pages 29–32, Hilton Head, SC, March 2001.

[7] Steven R. Michaud, Jesse B. Zydallis, David M. Strong, and Gary Lamont. Load balancing search algorithms on a heterogeneous cluster of pcs. In *Proceedings of the Tenth SIAM Conference on Parallel Processing for Scientific Computing (PP01)*, Portsmouth, VA, March 2001.