Accelerating Molecular Conformational Searches Using Genetic Algorithms

Victor Gueorguiev[1], Supervisor: Prof. Michelle Kuttel

***Abstract*—Predicting the structure of protein chains and other chemical structures is an open problem in chemistry. In particular, computational chemistry aims to use methods such as genetic algorithms to find optimal conformations for molecules. In this paper, a method proposed by F. Custodio et al is investigated and replicated together with some minor modifications and more extensive comparisons made with other methods developed over recent years for the problem. It is shown that the proposed method improves upon the results where possible, and also does so using less computation for larger data sets.**

***Index Terms*—'HP Lattice' 'population crowding' 'genetic algorithm' 'conformational searches' 'energy function optimization' 'hydrophobic-hydrophilic model'**

I. Introduction

PREDICTING the shapes and structures of molecules of various kinds has been a prominent area of research in the field of chemistry and, more specifically, in that of computational chemistry. Such predictions rely on information about how the molecules behave, such as the forces between bonding atoms and fundamental properties of the atoms such as charge and atomic radius. These simple-seeming conditions can, however, give rise to extremely complex behavior which creates a problem space of all possible molecular conformations too large to search through using a brute force technique or even cleverly designed classical algorithms. In fact, for many problems of protein structure prediction, the problem has been proven to be NP-complete [1], that is, there is no polynomial time solution to the problem. As a result, other methods for searching for good local maxima (short of not finding a global maximum), i.e. the conformation with the minimum bonding energy that is also physically realizable, have been developed over the years since the field's birth.

The ability to predict the shapes of hypothesized molecules of varying type has a plethora of applications in medicine and engineering [2] [3]. In medicine for example, many drugs rely on the structure of constituent molecules to deliver drugs into cells of the patient, where the shapes of the molecules determine the type of transmission through a cell barrier is possible. In many other applications such as engineering, discovering new materials which are based off of amino and protein molecules create a crucial need to know the structure of the new material in order to have a good indication of its physical traits before investment in the material can be furthered. In order to help find such molecules of interest, models around the shapes of these molecules must be built. In most of these scenarios, molecules that have of the order of tens or hundreds of atoms prove to be a challenge to model and minimize analytically. It therefore is expedient to have smart, efficient algorithms that can find minimum conformations without searching the entire population of solutions.

Ethically, predicting protein structures might have serious consequences as it does benefits. Such techniques may produce an incorrect result which may lead to incorrect usage of proteins in the applications above, leading to negative impacts on peoples' lives.

With this context above in mind, the aim of this research project pertains to structure prediction on 3D HP molecules by using current models of molecular energy and bonding regimes to specify a domain that can be investigated and solved by a genetic algorithm. The genetic algorithm will serve the purpose of finding physically possible conformations that minimize the energy for a molecule in the domain of hydrocarbons and produce structures that are conducive with expectations on existing data. The above aim is elucidated by replicating the paper by F. Custodio et al [4], which is further described in following sections, in which their use of crowding-out in their population domain to retain genetic diversity in their population greatly benefited their search for candidate solutions. This project will aim to replicate the results of this paper, and in the process create a system that might in principle also be applied to hydrocarbons as well as other protein folds.

II. Related Works

There has been great interest in recent years in applying machine learning algorithms of various kinds and complexity to the problem of protein structure prediction with varied results and success.

In their paper, J. Cheng and A. Tegge [3] describe various areas in which machine learning algorithms such as neural networks and support vector machines have been useful in performing tasks such as successfully identifying contacts between different amino acids in a molecular chain, predicting

[1] Email: GRGVIC001@myuct.ac.za; vctr.grgv2@gmail.com;

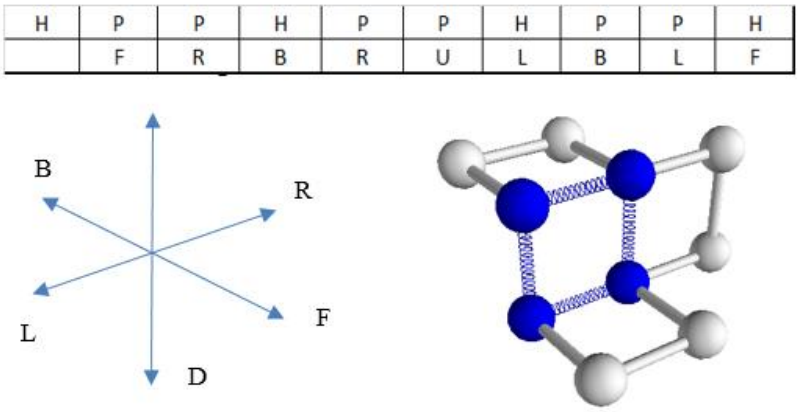

Figure 1: Example of the encoding scheme: The blue spheres are H monomers and white are the P monomers. The white connectors indicate the molecule chain and blue connectors indicate the HH contacts. Here the fitness of the molecule is 4 (4 contacts).

the expected properties of certain molecules given their configuration and constituent atoms, and, more akin to this work, predicting the conformations of three dimensional protein structures.

C.H.Q Ding et al. [5] also provide a novel approach to classification of molecules via their support vector machine implementation which allowed them to classify molecules into various categories based on different structural properties. They also showed that their method could outperform some previously applied neural networks to the problem.

Of course, the disadvantage of using neural networks and support vector machines comes in that to train these models, prior knowledge must be built into the system in the sense that some large batch of training data must be fed to the system before it can begin to behave in an according manner to predict similar cases. The basis for this project however, is to allow the model to pick best conformations based on only one piece of prior knowledge given to the system: the rules of potentials and energies of a given molecule as specified by real physical systems. Of course, other models also include some techniques in their introduction of optimal structures in their initial sample population as J. Cheng et al [3] when the subsection on using genetic algorithms on protein structure prediction is mentioned.

Some other methods which the proposed algorithm is compared against in this paper include using memetic algorithms by A. Bazzoli et al [6], particle swarm optimization by N. Mansour et al [7] and a contact interactions method employed by L. Toma et al [8]. These methods are more specific to the problem looked at in this paper, that is, the hydrophobic-hydrophilic lattice model[2]. These methods have been shown to produce good results, if not optimal solutions, to many sequences encountered in biology [9] and others that are randomly generated [10] [11] [7]. However, a disadvantage of these methods is that they require more computation to reach a desired result accuracy, thus limiting the size of the problems they can tackle. This issue will be revisited later in the paper.

[2] It makes comparison of the proposed algorithm with these methods more feasible, since they are applied to the same data sets.

III. Materials for the Genetic Algorithm

The materials and algorithm for the algorithm described below are taken from the paper by F.L. Custodio et al and others. [4] [12] [13] [14]

A. Model for the 3D HP Lattice

As mentioned by F.L. Custodio [4], creating a sequence for the genetic algorithm using Cartesian coordinates (x, y, z) will be computationally cumbersome and memory intensive. Therefore, the model used in Figure 1 is used as in their original paper. To reiterate the approach, the model stores a sequence of directions which are traversed in the 3D lattice to reach the next hydrophobic or hydrophilic monomer. The set of directions therefore has six elements: $\{F, B, L, R, U, D\}$. The sequence of HP monomers is varied only between different molecular test cases- each test case looks to find the best conformation for a given HP sequence by modifying the direction sequence using genetic operators.

B. Genetic Operators for the Genetic Algorithm

THE methods used in the genetic algorithm are described here in general and apply to the hydrophobic-hydrophilic model as a modeled sequence/string of characters.

The various operators used in the genetic algorithm were two-point crossover (2X), multi-point crossover (MPX), local move (LM), segment mutation (SMUT), exhaustive search mutation (EMUT) and loop move (LPM). These methods are described in detail and are used as given by F. L. Custodio et al [4], and explanations on many such as the 2PX and MPX are quite standard and can be found at length in M. Melanie's book [15]. More complicated operators such as LM, LPM, SMUT and EMUT are more complex and problem-specific and thus are also illustrated in Figure 2.

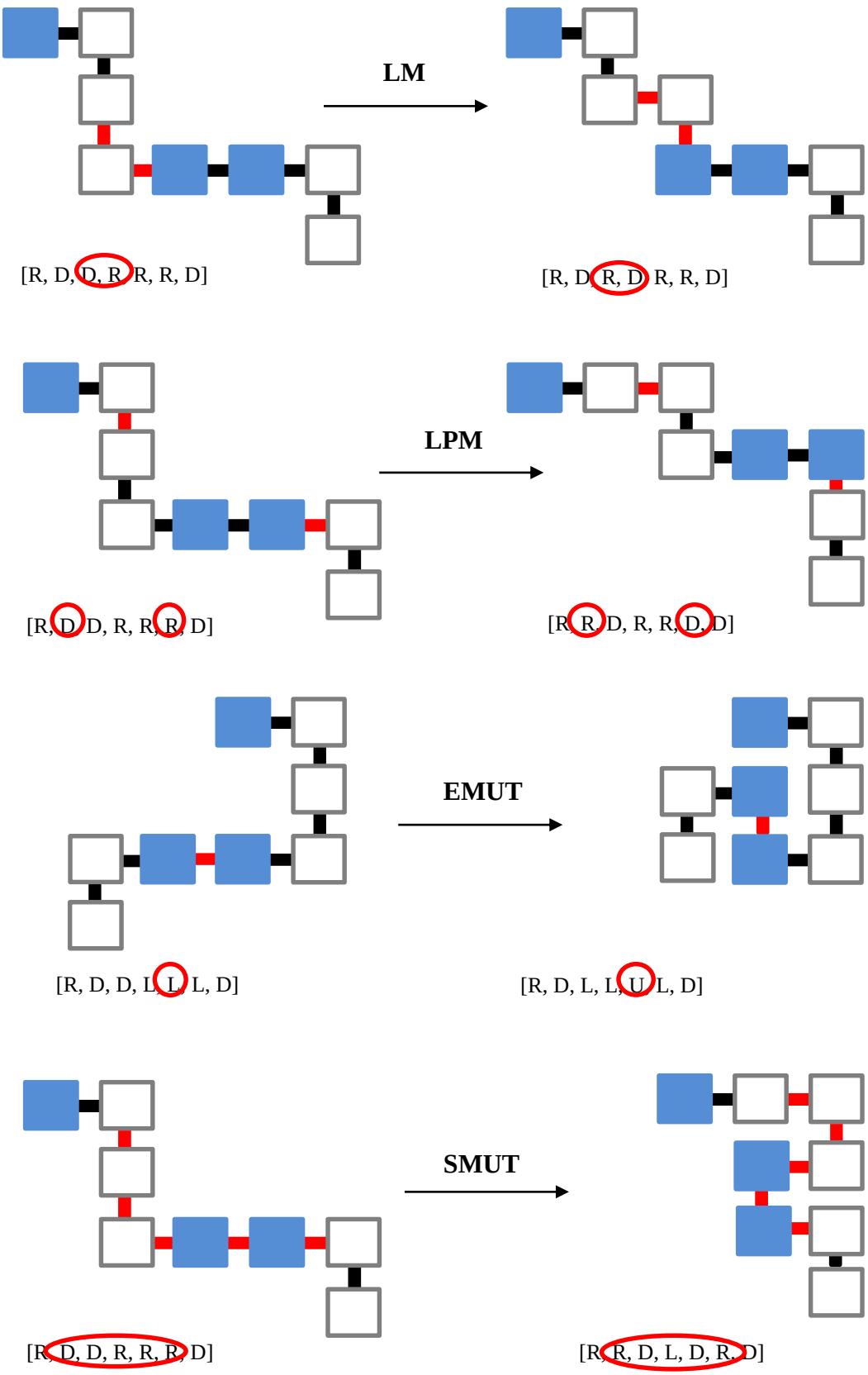

Figure 2: The randomly selected red-circled direction bases have their directions swapped for the LM and LPM cases.

For SMUT, the circled sequence of bases has their bases randomly reassigned.

For EMUT, the random change in direction is performed until the best possible improvement produced the result with an increase in fitness of one HH contact.

The 2X operator is the standard operator which conducts a two-point crossover in the parents of the next member of population. A crossover point is chosen at random and the parent sequences are split at the crossover point. At this point, the offspring are formed by combining the first split sequence from parent one with the second split sequence from parent two. The second offspring is produced by combining the alternate segments, that is, by combining the second split sequence from the first parent to the first split sequence from the second parent.

The MPX operator is a more complex and general case of operator to the 2X operator used in generating new population members. The MPX operator works in a similar process to above, by selecting crossover points in the parent sequences (this time multiple, and the exact number determined randomly between 2 and 10) and then two new offspring population members produced using alternating segments from each parent, that is, if segment x is taken from parent one, then $x + 1$ is taken from parent two. The second child is produced similarly by using the alternating segment choices from each parent.

The LM operator works by swapping the directions of two consecutive monomers on a random location in the HP sequence. For example, if one molecule moves up and its successor goes right, then the molecule will go right and its successor will go up. After this operator has acted, the collision detection methods need to be applied to ensure that this molecule is still a valid conformation.

The LPM operator works in a similar fashion, except that the molecules with swapped directions are not necessarily next to one another, but are chosen from two random locations on the monomer chain. Similarly to above, the collision detection needs to be applied to ensure that the molecule is still valid.

The SMUT operator is an operator that changes a segment of molecule (the size of the segment is determined randomly between 2 and 7) by taking each monomer in this segment and changing its direction randomly to one of the five directions. Note that the monomer does not necessarily have to change, the random choice may be the same as its original value.

The EMUT operator takes a random monomer in the HP sequence specified and changes its direction to the best possible directions among the 5 directions possible by evaluating the fitness function for each of these possibilities. In fact, only four need to be done, since the current conformation evaluation is already calculated in prior steps and stored.

With both of these SMUT and EMUT operators, collision detection also needs to be applied to ensure the molecule is still valid.

C. Validation of the HP Molecular Chain

In using the three dimensional lattice as a model for hydrophobic-hydrophobic monomer chains, where initial population members are randomly generated, it is crucial to ensure that such molecules are physically realizable entities. In this investigation, using such a simplified model, this entailed checking that, given a set of directions to navigate along the chain of monomer from the starting monomer, there was no collision anywhere along the chain which would result from two monomers occupying the same point in space as another monomer.

In reality, there is clearly no molecule that accommodates collisions among constituent monomers. To check the model for any collisions, there is a brute-force way of detecting collisions. This method [4] [8] is as follows: Start with a three dimensional Cartesian point as the origin. Then, for each given direction, add or subtract a unit from the relevant coordinate of the point and relevant direction traversed (for example, moving forward will add 1 to the x coordinate, moving backwards subtracts 1). Then for each point calculated- including the starting point- add it to a list or set of points which will be used to check against later. For each new calculated point, check whether the set of points already contains this point; if it does, then clearly this point in space is already occupied, otherwise continue checking. If the molecule is traversed without any collisions, then it is a valid conformation.

This technique is applied as part of the repair mechanism. When generating new molecules, the candidate is checked at each stage/monomer for a collision, and is then assigned a new random direction until there is no longer a collision. When a situation arises where all directions lead to collisions, then rather than removing the member from the population, the fitness of the individual is assigned to zero. In the repair mechanism, new directions are tried until the new direction produces a point in space not currently occupied. Again, if no such point exists, the conformation is invalid and discarded, with the original valid conformation kept instead.

D. Parental Selection and Replacement

Selecting parent structures from the population takes the form of a tournament described by A. Brindle [14] and L. Davis [12]. To summarize, a tournament size of four is used, and candidates for the tournament are selected randomly from the population; the tournament then proceeds by selecting one or two individuals (based on the operator being used) with a probability of being selected given by each individual's fitness as a fraction of the total fitness of the tournament or in other words each probability is given by,

$$P_i = \frac{f_i}{\sum_{n=1}^{N} f_n} \dots (2)$$

Above, we have the probability of selecting the ith individual from a population of size N.

This allows for the fittest individuals to be selected more often while still preserving diversity in the population. Some folks showed that this indeed maintained diversity in the structures of molecules exhibited in the population.

When new individuals are created from the parental population, the question of which members of the population must be replaced to maintain a fixed population size. The standard genetic algorithm approach has that the member with lowest fitness in the population be replaced assuming that the new individual has a higher fitness than the lowest [12] [11]. However, this might end up removing certain structural diversities from the population. Rather what S.W. Mahfound [13] and F.L Custodio et al [4] propose is a method by which only members closest to each other in terms of conformation are compared using their fitness, leading to either a retention or replacement. Concretely, a distance metric is used to determine the closest matching individual in the population to the new member; once found, if the new member has a better energy than the member of similar shape, it is replaced in the population. If they have equal values, then there is a fifty percent chance it is replaced. In the final case the new individual is discarded. The distance metric used to determine structurally similar individuals is the distance matrix error, given by,

$$DME = \sqrt{\sum_{i,j}^{N} \frac{p_i - q_j}{N(N-1)}} \dots (3)$$

In the above equation, summation is over the length of both the molecules, and the value p_i denotes the magnitude of the ith coordinate of H monomer site in the first chain and q_j denotes the magnitude of the jth coordinate of the H monomer site on the second chain.

This has the advantage of determining exactly which monomers are different and by what magnitude, then applying a squared error standard deviation to arrive at the result. While this is computationally expensive, it is helped along by storing pre-calculated values in the implementation of the molecule, to avoid doing too many computations for each comparison. Even using matrix operations, this step is still $O(n^2)$ and thus one of the biggest bottlenecks in the algorithm.

E. Initial population and parameters

The initial population conformations are randomly generated and then checked using the repair mechanism above, following the assignment of the monomer sequence of hydrophilic-hydrophobic arrangements. In the work done by F.L. Custodio [4], the initial population was set to 500 individuals and function evaluations limited to four million-roughly two hundred thousand generations in this scheme. In cases of comparisons with other algorithms and where optimal energies are known, this upper bound was rarely reached.

Each generation creates ten new individuals which may or may not be added to the population. This also coincides with the frequency of operator application probability adjustments discussed below.

F. Operator Application Probabilities

In the works done by, and, the advantages of using dynamic operator application probabilities is illustrated in assisting with keeping out of local optima and finding the global minimum fitness in the fitness landscape. An approach similar to those mentioned by L. Davis [12] and A. Brindle [14] is used. Whenever an operator creates an individual with better fitness than the current best in the population, that operator is rewarded with some numerical reward equal to the difference in HH contacts between the old best and current individual. The operators that created the parent of the individual are rewarded with half that amount (or a quarter to each if two parents were used in crossover rather than a single parent in mutation). This simple reward addition is given by

$$R_i = R_i + \left(f_{new_{best}} - f_{old_{best}}\right) \dots (4)$$

Above we have the reward for the ith operator. The probability is then,

$$P(i) = \frac{R_i}{\sum_{n=1}^{6} R_n} \dots (5)$$

After the creation of ten new individuals, the probabilities of operators are adjusted to new values calculated as their current overall reward as fraction of the total reward for all operators. No probability can fall below five percent so as to eliminate it from use, so in which case a simple check subtracted the shortfall from the current highest probability to keep probabilities guaranteed above five percent. The above conditions maintain that the cumulative probability sums to one as it should. The rewards for each operator are initialized at one to ensure uniformity. Finally, if an operator has not produced an optimal individual in five hundred calls to the method then a penalty of one unit is given as a negative reward, while maintaining that the operator's reward stays above one. This mechanism ensures that operators that stagnate the population after some time are penalized and then allows for other operators to be more likely tried, eventually returning to a uniform distribution if no improvements are made. This has the benefit of exploring the fitness landscape more efficiently for a global minimum.

G. Test Data Sets

The data sets used for this are ten specified sequences for molecules of length forty-eight monomers [16], and ten sequences of length sixty-four monomers randomly generated (in other words they bare no physical significance) [7]. Also used are sequences of length forty-six, fifty-eight, one hundred and three, one hundred and twenty-four and one hundred and thirty-six which are biologically inspired from the data set used by K. Dill et al [9]. Finally, the additional data set used by N. Mansour [7] includes ten sequences of length twenty-

Table 1
HP Results for Sequences of Length 48

Sequence	No. of HH Contacts						
	AGAPC (Best)	AGACP - μ (σ)	SGA	MA	ACO	CHCC	MA
48.1	**32**	29.98 (0.89)	24	**32**	**32**	**32**	**32**
48.2	**34**	31.11 (0.78)	24	**34**	**34**	**34**	**34**
48.3	32	30.41 (0.52)	23	**34**	**34**	**34**	**34**
48.4	**33**	30.93 (0.94)	24	**33**	**33**	**33**	**33**
48.5	31	29.81 (0.65)	28	**32**	**32**	**32**	**32**
48.6	**32**	29.32 (0.86)	25	**32**	**32**	**32**	**32**
48.7	**32**	28.32 (1.03)	27	31	**32**	**32**	31
48.8	**31**	28.26 (1.15)	26	**31**	**31**	**31**	**31**
48.9	**34**	30.98 (0.85)	27	33	**34**	**34**	33
48.10	**33**	30.06 (0.68)	26	**33**	**33**	**33**	**33**

Absolute value of HH contacts are shown. In bold are the optimal contacts. The values shown for AGADP are the best of 50 runs, then the average and standard deviation in brackets. The other methods are CI – Contact interactions [8]; SGA – Standard genetic algorithm; MA – Memetic algorithm [6]; ACO – Ant Colony Optimization [7]; CHCC – Constraint-based Hydrophobic-core construction [10].

Table 2
HP Results for Sequences of Length 64

Sequence	No. of HH Contacts			
	AGAPC (Best)	AGACP - μ (σ)	SGA	PSO
64.1	*30*	26.13 (0.95)	27	28
64.2	36	32.25 (1.36)	29	31
64.3	*43*	40.69 (0.56)	35	39
64.4	39	35.80 (1.85)	34	36
64.5	*38*	34.88 (1.10)	32	38
64.6	*31*	28.55 (0.86)	29	31
64.7	*27*	24.69 (1.03)	20	27
64.8	36	32.26 (1.35)	29	35
64.9	38	35.12 (1.25)	32	35
64.10	33	28.51 (0.68)	24	27

Absolute value of HH contacts are shown. In bold are the optimal contacts. The values shown for AGADP are the best of 50 runs, then the average and standard deviation in brackets. The other methods are SGA – Standard genetic algorithm; PCO – Particle Colony Optimization [7].

Table 3
Function Evaluations Spent on Sequences of Length 64

Sequence	No. of Function Evaluations		
	AGAPC	SGA	PSO
64.1	312 000	433 533	422 373
64.2	268 000	167 017	159 873
64.3	125 600	172 192	109 541
64.4	132 100	107 146	197 879
64.5	161 600	154 168	189 634
64.6	231 400	454 727	410 586
64.7	99 100	320 396	309 532
64.8	165 400	315 036	410 813
64.9	168 900	151 705	143 182
64.10	153 200	191 019	165 762

The function calls shown for AGADP are the best of 50 runs. The other methods SGA – Standard genetic algorithm; PCO – Particle Colony Optimization [7]

seven which are additional to this investigation and do not appear in the original work by F.L. Custodio et al.

Data sets are available in Appendix A and are provided by [6] [9] [16] [5] [10]

IV. Results and Discussion

A. HP Results on Test Data

In hindsight, the algorithm developed as a replica to that of F.L Custodio et al produced comparable results for the overwhelming majority of cases. As a consequence, the GAHP results of their paper are not reproduced here, but rather the reader is referred to their work [4]. The GAHP replica-algorithm is here quoted as AGADP, short for adaptive genetic algorithm with dynamic probabilities, to avoid confusion with the original method.

For each test case- i.e. the molecule used- a set of fifty runs is recorded using the same aforementioned parameters and then the best result, the average of the best results and their standard deviation is quoted for the AGADP. In the various tables, other results from different works and algorithms are quoted.

For the set of forty-eight length monomers (results in Table 1) the AGADP produced results that agree with GAHP on eight of the ten cases. In each case however, the averages between AGADP and any other method did fall within the bounds of each other's standard deviations. From an experimental stand point, this implies the two algorithms achieved close results, although AGADP did not find the global minimum on two cases. As with GAHP, it did better than SGA on all cases, and where GAHP beat out MA on two cases, the proposed algorithm matches it on number of optimal test cases. On many of the test cases the averages trailed the minimum by two or three HH contacts, and furthermore the standard deviation is low. As F.L. Custodio et al [4] point out, this reflects that near the global minimum there are ever more conformations of low energies that cause local minima traps. The graphs in Figure 3 also indicate this by the peak in number of conformations over all fifty runs just before the global minimum value for

Table 4

Function Evaluations Spent on Sequences of Length

Sequence	No. of Function Evaluations		
	AGAPC	SGA	PSO
27.1	8 970	15 854	3 158
27.2	10 560	19 965	5 771
27.3	14 560	7 991	2 667
27.4	13 300	23 525	8 556
27.5	5 650	3 561	893
27.6	9 870	14 733	12 790
27.7	20 590	23 112	17 024
27.8	2 130	889	149
27.9	2 180	5 418	1 915
27.10	6 150	5 592	2 638

The function calls shown for AGADP are the best of 50 runs. The other methods SGA – Standard genetic algorithm; PCO – Particle Colony Optimization [7]

Table 5

HP Results for Sequences of Length 27

Sequence	No. of HH Contacts			
	AGAPC (Best)	AGACP - μ (σ)	SGA	PSO
27.1	9	7.21 (0.95)	9	9
27.2	10	9.01 (0.36)	10	10
27.3	8	7.12 (0.56)	8	8
27.4	15	13.56 (0.80)	15	15
27.5	8	7.31 (0.21)	8	8
27.6	11	9.89 (0.66)	11	11
27.7	13	12.06 (0.36)	13	13
27.8	4	3.26 (0.32)	4	4
27.9	7	5.87 (0.60)	7	7
27.10	11	10.14 (0.68)	11	11

Absolute value of HH contacts are shown. In bold are the optimal contacts. The values shown for AGADP are the best of 50 runs, then the average and standard deviation in brackets. The other methods are SGA – Standard genetic algorithm; PCO – Particle Colony Optimization [7].

Table 6

HP Results for Biologically Inspired Sequences

Sequence	No. of HH Contacts		
	AGAPC (Best)	AGACP - μ (σ)	CI
46	36	33.23 (2.45)	34
58	43	38.56 (2.36)	42
103	52	47.67 (3.56)	49
124	63	56.55 (4.85)	58
136	68	60.32 (5.01)	65

Absolute value of HH contacts are shown. In bold are the optimal contacts. The values shown for AGADP are the best of 50 runs, then the average and standard deviation in brackets. The other methods are CI – Contact interactions [8];

sequences 48.2 and 103. The results for the structures for some of the sequences are shown in Figure 4.

Table 2 displays data from the sixty-four length monomer test data together with the algorithms that produced their own results. As is seen, the energies produced are of better quality than both the standard genetic algorithm and particle swarm optimization. The results also agree well with those put forward by F.L. Custodio et.al [4] in their paper, with the exception of two sequences which had two fewer HH contacts than with GAHP. As before, the average and standard deviation of best fitness values over the fifty runs is indicative of the tightly packed fitness landscape of local optima around sites of possible a global minimum and the algorithms tendency to get stuck in these. It should be noted that the results in Table 2 have not been confirmed to be optimal by any means, and there is no reason to believe yet that they are at all. Some structures are shown in Figure 5.

The results of the twenty-seven length monomer chains in Table 5 and some example structures are shown in Figure 7. Here, both the standard algorithm and the swarm optimization achieved known global optimal values for all sequences, as did the proposed algorithm AGADP. With such results, it may be worth considering where each algorithm has its merits. For an answer to this, a look at the function evaluations will prove useful later.

Finally, the biologically inspired sequences [9] are given in Table 6. Once again, the proposed algorithm outperformed the contact interaction method in finding higher numbers of HH contacts. The results the third sequence (103) are shown in Figure 6. An important pattern to pick up on in the above results is that as the size of the molecule is increased, the average deviates further from the best predicted value of the algorithm and the standard deviation grows. As mentioned for sequences of length forty-eight, this is a feature of the packed fitness space [17] near larger numbers of HH contacts where many local optima traps begin to occur. Indeed, Figure 3a and Figure 3b, for the 48.2 and 103 length monomers respectively, that as the algorithm approaches the best result, the number of unique conformations over the fifty run test with a certain fitness spikes near the best result. This maximum or spike shifts left as the sequence length increases, or in other words as the molecule complexity increases, and is seen through the marked difference in graph maxima between the graphs in Figure 3.

B. Function Evaluations during Testing

In benchmarking performance between different algorithms, the number of fitness function evaluations should be taken into account, as in the work done by F. L. Custodio et al [4]. In fact, since the AGADP performed at a near equivalent caliber, much discussion will be merely summarizing points made by those authors.

For sequences of length 64, the number of function evaluations required to arrive at the best result are recorded in Table 3. Overall, except for two sequences, the number of function evaluations on these sequences was less than those of competing algorithms, with the two discrepancies attributed to

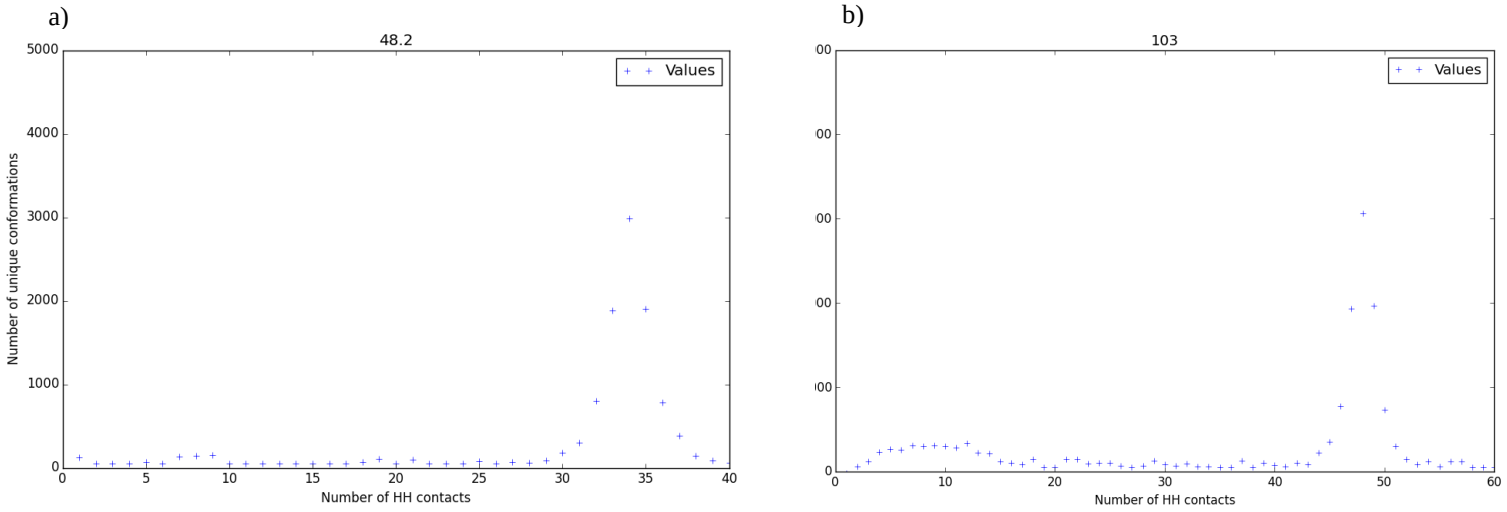

Figure 3: Graphs of number of unique conformations for sequences 48.2 (a) and 103 (b)

the complex fitness landscape mentioned above which needed to be traversed by the proposed algorithm.

However, in the original paper by F.L. Custodio, an evaluation of the algorithm's performance for twenty-seven length monomers was not performed in comparison with previous approaches. The number of fitness function evaluations for twenty-seven length monomers are stored in Table 7. Here one can see that the algorithm performs substantially worse than previously attempted algorithms which similarly found global minimum energies, requiring many more function evaluations- in some cases double- to achieve the same results. This poor performance on smaller length monomer chains might be attributed to the fact that the algorithm uses six operators together with dynamically varying probabilities, which might be an overly complex solution to a level of problem still able to be solved by simpler methods- for example, the EMUT requires four fitness evaluations but may be an unnecessary operator for small problems.

What can be learned from this is that the proposed AGADP algorithm performs poorly on smaller data sets, but outperforms other methods as the data size increases. Again, due to the large number of local optima traps [17] which develop around the global minimum as the length of the molecule increases (see Figure 3), it can be seen that this algorithm more efficiently and successfully navigates past these traps.

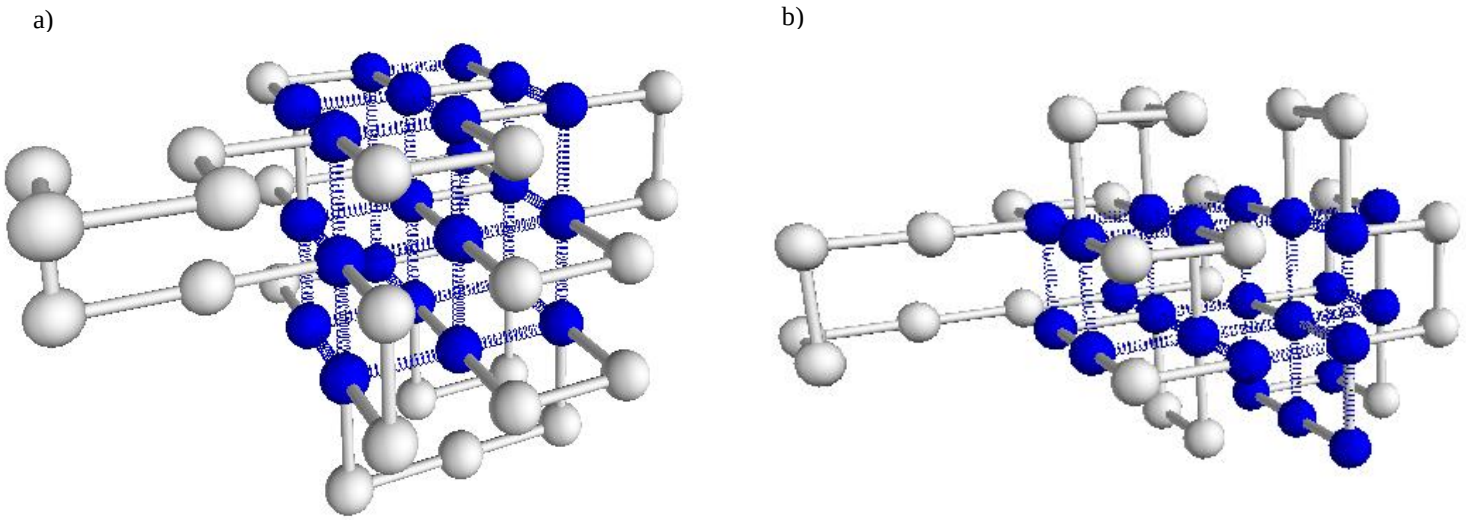

Figure 4: Example structures for sequences 48.2 (a), with 39 HH contacts, and 48.7 (b) with 32 HH.

a) b)

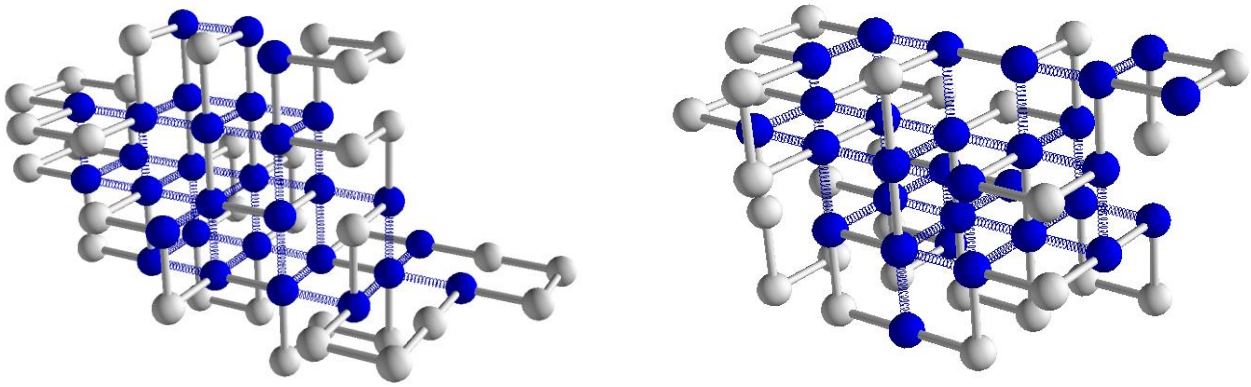

Figure 5: Example structures for sequences 64.5 (a) (38 contacts) and 64.8 (b) (36 contacts).

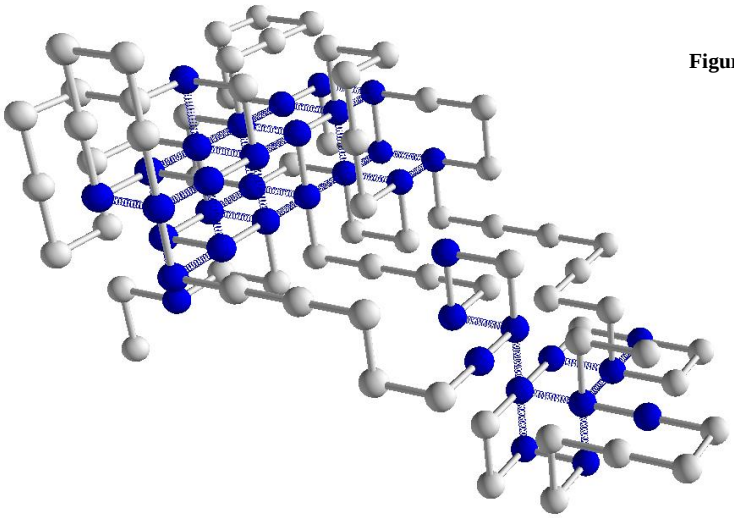

Figure 5: Example structures for sequence 103 with a di-core structure and 52 contacts

---

a) b) c)

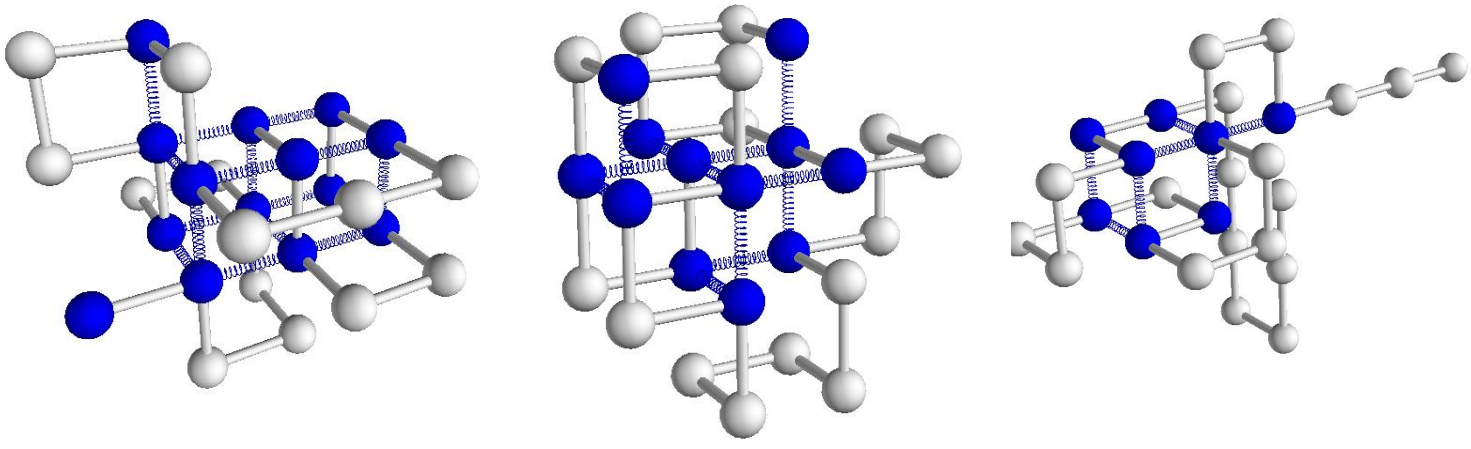

Figure 7: Example structures for sequences 27.4 (a) (15 contacts), 27.1 (b) (9 contacts) And 27.9 (c) (7 contacts)

V. Conclusions, Evaluation and Future Work

The proposed algorithm adequately replicates the results produced by F.L Custodio et.al in predicting the global minimum for a variety of data sets or in predicting the energies found for those data sets where no global minimum is known yet. It is also shown how in growing the monomer length, the complexity of the problem increases by increasing the local minima traps near the global minimum as the number of conformations with a given energy increases near such points. The AGADP neatly navigates past these as did its master clone GAHP [4]. Finally, the number of function evaluations analysis gave information that the proposed algorithm performed poorly for smaller length data sets while still finding global minimums for each sample, but began to outperform other global minimum achieving algorithms in larger length data sets.

While F.L. Custodio et.al [4] and others [1] [17] [7] have argued that number of fitness function evaluations is the crucial measure of performance when considering the time it takes to reach global minimum, it may be of benefit to switch to a notion of real-time analysis on various parts of the algorithm in boosting its performance. While function evaluations are normally the most cumbersome operations, a real-time analysis will help identify other bottlenecks for future works.

Such examples of future works will be to transform this model into a parallel genetic algorithm to execute on multi-core computer architectures. Clearly, a more detailed analysis will be needed than just the number of function evaluations in determining the performance boost that can be offered by such a change in approach. Extending this genetic algorithm to a free 3D system with more complicated potentials, for example see F.L. Custodio et al [4], is also a large possibility, as it will allow the analysis of far more diverse and complex systems which have even more real-world applicability.

VI. Acknowledgements

Many thanks to the supervisor of this project, Professor Michelle Kuttel of the University of Cape Town, for her guidance and knowledge which helped contribute to the fruition of this project.

VII. Bibliography

[1] B. Berger and T. Leighton, "Protein folding in the hydrophobic hydrophilic (HP) model is NP-complete," *Journal of Computational Biology,* vol. 5, pp. 27-40, 1995.

[2] M. Jacobson and A. Sali, "Comparative Protein Structure Modeling and its Applications to Drug Discovery," *Annual Reports in Medicinal Chemistry,* vol. 39, pp. 259-273, 2004.

[3] J. Cheng and A. Tegge, "Machine learning methods for protein structure prediction," *IEEE Reviews in Biomedical Engineering,* vol. 1, pp. 41-49, 2008.

[4] F. Custodio and L. Barbosa, "A multiple minima genetc algorithm for protein structure prediction," *Applied Soft Computing,* vol. 15, pp. 88-99, 2014.

[5] C. Ding and I. Dubchak, "Multi-class protein folding using support vector machines and neural networks," *BioInformatics,* vol. 17, no. 4, pp. 349-358, 2001.

[6] A. Bazzoli and A. G. B. Tettamanzi, "A memetic algorithm for protein structure prediction in a 3D-lattice HP model," *Applications of Evolutionary Computing, EvoWorkshops,* pp. 1-10, 2004.

[7] N. Mansour, F. Kanj and K. Hassan, "Particle Swarm Optimization Approach for the Protein Structure Prediction in the 3D HP Model," *Interdisciplinary Sciences Computer Life Sciences,* vol. 4, pp. 190-200, 2012.

[8] L. Toma and S. Toma, "Contact interactions method: A new algorithm for protein folding simulations," *Protein Science,* pp. 147-153, 1996.

[9] K. A. Dill, H. S. Chan and K. M. Fiebig, "Cooperativity in protien-folding kinetics," *Proceedings of the National Academy of Sciences, USA,* vol. 90, pp. 1942-1946, 1993.

[10] K. Yue, K. M. Fiebig, P. D. Thomas, H. S. Chan, E. I. Shaknhovich and K. A. Dill, "A test of lattice protein folding algorithms," *Proceedings of the National Academy of Sciences, USA,* pp. 325-329, 1995.

[11] R. Unger and J. Moult, "Genetic Algorithms for Protein Folding Simulations," *Journal of Molecular Biology,* vol. 231, pp. 75-81, 1993.

[12] L. Davis, Handbook of Genetic Algorithms, Boston: London International Thomson Computer Press, 1996.

[13] S. Mahfound, "Crowding and preselection revisited," *Nature II,* vol. 2, pp. 27-36, 1992.

[14] A. Brindle, "Genetic Algorithms for Function Optimization," The University of Alberta, Ottawa, 1980.

[15] M. Melanie, Introduction to Genetic Algorithms, Cambirdge, Massachusetts: The MIT Press, 1999.

[16] W. Hart, N. Krasnogor, J. Smith and D. Pelta, "Protein structure prediction with evolutionary algorithms," *Proceedings of the Genetic and Evolutionary Computation Conference,* pp. 1596-1601, 1999.

[17] S. Flores and J. Smith, "Study of fitness landscapes for the hp model of protein structure prediciton," *Proceedings of the 2003 Congress on Evolutionary Computation CEC2003,* pp. 2338-2345, 2003.