Faster Evaluation of Protein Energies and Energy Gradients *

R. H. Byrd
E. Eskow
W. J. Pullan
R. B. Schnabel

CU-CS-855-98

University of Colorado at Boulder
DEPARTMENT OF COMPUTER SCIENCE

* Research supported by Air Force Office of Scientific Research Grant F49620-97-1-0164, Army Research Office Contract DAAH04-94-G-0228, and NSF grant CDA-9502956.

ANY OPINIONS, FINDINGS, AND CONCLUSIONS OR RECOMMENDATIONS EXPRESSED IN THIS PUBLICATION ARE THOSE OF THE AUTHOR(S) AND DO NOT NECESSARILY REFLECT THE VIEWS OF THE AGENCIES NAMED IN THE ACKNOWLEDGMENTS SECTION.

Faster Evaluation of Protein Energies and Energy Gradients [*]

CU-CS-855-98

R.H.Byrd
Department of Computer Science,
University of Colorado

E. Eskow
Department of Computer Science,
University of Colorado

W.J.Pullan
Department of Mathematics and Computing,
Central Queensland University

R.B.Schnabel
Department of Computer Science,
University of Colorado

March 31, 1998

Abstract

This paper describes a hybrid table lookup / exact calculation method (TLEC) for calculating both protein energies and energy gradients. The tables used are relatively small and contain the expected non-bonded and coulombic interaction energies and energy gradients between components of residues which form the protein. The protein energy is calculated using both these tables and exact computation of the actual energy when components are close enough such that their relative orientation gives a wide range in the possible energies. A similar technique is applied to the energy gradient calculations required by most local optimisers. Computational results comparing local optimisers implemented using both the complete CHARMM energy and energy gradient calculations and the TLEC method are presented. These results show that the use of TLEC results in a speedup of protein energy calculations by a factor of 6.0, gradient energy calculations by a factor of 3.1 and local optimisations by a factor of 4.5 while still giving comparable results.

Keywords: Global Optimisation, Proteins.

[*] Research supported by Air Force Office of Scientific Research Grant F49620-97-1-0164, Army Research Office Contract DAAH04-94-G-0228, and NSF grant CDA-9502956.

1 Introduction

Proteins play an essential role in almost all biological processes by acting as enzymatic catalysts, providing transport and storage mechanisms, enabling coordinated motion, mechanical support and immune protection, generation and transmission of nerve impulses and controlling growth and differentiation. All proteins consist of a sequence of amino acids, of which only twenty different forms exist. Amino acids consist of an amino group (NH_3^+), a carboxyl group (COO^-), a hydrogen atom H, and a distinctive R group (side chain) bonded to a carbon atom C_a (Figure 1).

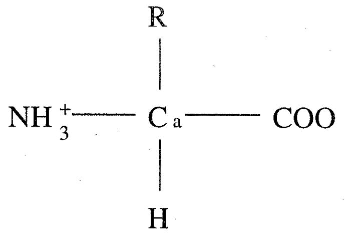

Figure 1: Generic amino acid. Twenty different amino acids exist differing only in the composition of the R side-chain.

In proteins, the carboxyl group of one amino acid is joined to the amino group of another amino acid by a peptide bond (Figure 2). As the peptide bond is a partial double-bond, there is no rotation possible around this bond and the structure of the protein may be specified in terms of the rotation about the pure single bonds between the two carbon atoms and between the carbon and nitrogen atoms.

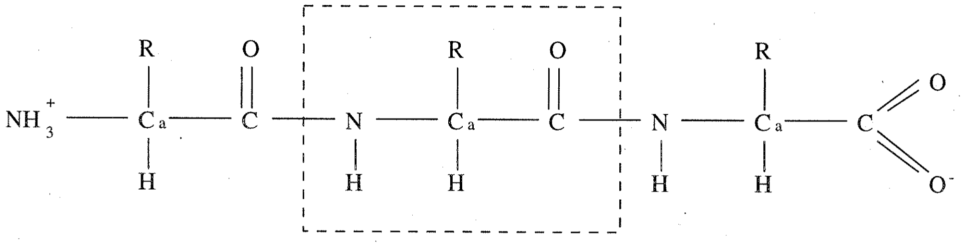

Figure 2: Protein containing 3 residues (an amino acid unit in a protein is usually referred to as a residue). For larger proteins, the residue outlined by the box is repeated a number of times.

The sequence of residues (the primary structure of the protein) determines how the protein will act as a catalyst and also appears to determine the three dimensional structure of the protein (the tertiary structure) [1]. The tertiary structure is important as it appears to be a critical determinant of the biological function of the protein. The problem of determining the tertiary structure of the protein, given its primary structure, is usually referred to as the protein folding problem and has attracted considerable interest from researchers over a number of years [2].

Currently there are two major analytical methods which have been used in attempts to solve the protein folding problem. The first, molecular dynamics, employs standard Newtonian physics to model the creation of the tertiary structure. Inherent problems with this method are the accuracy of the equations when applied at the atomic level and the timescales involved. For accurate simulation, timesteps of the order of 10^{-15} seconds are required however, for proteins, times of the order of $10^{-1} - 10^{3}$ seconds need to be simulated. Currently, times of the order of 10^{-8} have been achieved.

The alternative method of global optimisation uses the observation that the tertiary structure of the protein is nearly always that which corresponds to the global minimum of the potential energy for the protein. Thus the protein folding problem can be viewed as a global optimisation problem where the objective function to be minimised is the potential energy of the protein and the parameters of the problem are those which determine the physical structure of the protein. Inherent problems with this method are:

- developing an efficient global (as distinct from local) optimiser which is able to solve a problem of this magnitude.
- modelling the potential energy as a function of the structure of the protein. While there do exist models of varying degrees of accuracy, the more exact require considerable computational effort to calculate the protein potential energy.
- the extremely large number of local minima present on the potential energy hyper-surface. This appears to be exponentionally proportional to the degrees of freedom of the system [3] and makes locating the global minimum an extremely difficult problem.

As described in [4], a wide range of global optimisation methods have been applied to the protein folding problem. Generally they all have the common characteristic of requiring a large number of energy calculations. In addition some of these methods incorporate the use of a local optimiser which, to be efficient, requires that the gradient of the protein energy be calculated.

This paper addresses the problem of reducing the computational requirements of calculating the potential energy and energy gradient for the protein. First we describe the CHARMM model [5] of protein energy and present an overview of existing methods for reducing the computational requirements of calculating protein energy. Next the hybrid table look-up / exact calculation (TLEC) method used in this study is described and results obtained for calculating protein energies, energy gradients and local optimisations using both the CHARMM and TLEC are presented. Finally a summary and conclusion describes how TLEC may be incorporated into a global optimisation method in conjunction with the CHARMM model.

2 CHARMM Protein Energy

In the CHARMM model, the energy E of a protein is calculated as:

$$E = \sum E_s + \sum E_\theta + \sum E_\omega + \sum E_v + \sum E_{el} \tag{1}$$

where

- $E_s = \sum k_s(l - l_0)^2$ is the energy due to stretching/contraction of chemical bonds within the protein (l_0 is the natural bond length, l is the actual bond length and k_s is a constant which depends on the bond type).
- $E_\theta = \sum k_\theta(\theta - \theta_0)^2$ is the energy due to changes in bond angles away from their natural values. (θ_0 is the natural bond angle, θ is the actual bond angle and k_θ is a constant which depends on the bond type).
- $E_\omega = \sum V_s(1 + s\cos(n\omega))$ is the energy associated with rotation around the $N - C_a$ and $C_a - C$ bonds of each residue in the protein (ω is the dihedral angle which measures the amount of rotation about the bond and V_s is a constant which depends on the atoms involved in the bond).
- $E_v = \sum \epsilon((r_m/r)^{12} - 2(r_m/r)^6)$ is the Lennard-Jones energy associated with interactions between non-bonded atoms (r is the distance between the atoms while ϵ and r_m are constants which depend on the atom types).
- $E_{el} = \sum q_i q_j / D r_{ij}$ is the coulombic energy between non-bonded atoms (q is the charge associated with each atom and r is the distance between the atoms).

As an example of the computational requirements in calculating protein energy using this model, for the alanine 58–mer, the number of atoms $= 572$, bonds $= 571$, bond angles $= 1,026$, dihedral angles $= 341$, 1-4 interactions $= 1,416$ and non-bonded pairs $= 160,293$. Clearly the dominant computational element in the calculation of the protein energy and energy gradient is the computation of the Lennard-Jones and coulombic contributions for the non-bonded pairs (in fact it accounts for almost 99% of the total processor time used for both calculations).

The first simplification used by almost all global optimisation methods that have been applied to the protein folding problem is to assume that bond angles and bond lengths remain constant and the only variables are the two dihedral angles per residue. While this substantially reduces the degrees of freedom in the optimisation problem it only produces a marginal decrease ($< 1\%$) in the processor time for computing protein energies and energy gradients.

A number of global optimisation methods have been implemented which substantially reduce the non-bonded computational requirements of the energy calculation. These mainly consist of reducing the degrees of freedom in residues by using a simplified geometric representation, smoothing the potential energy hypersurface by the use of an average potential energy function and using heuristics to reduce the conformational space [3, 6, 7]. The fundamental assumption of these methods is that the basic folded structure of a protein is relatively insensitive to the fine details of atomic interactions and an average field interaction potential should be sufficient to account for the overall folding of a protein.

3 TLEC Protein Energy and Gradient Calculations

This study takes a different approach to those described above and investigates the feasibility of implementing a table look-up mechanism to reduce the computational requirements in calculating the protein energy and energy gradients. The overall goal of our study is to provide fast energy and energy gradient calculations that are sufficiently accurate so that a local optimiser using these calculations will find structures that are close to those found by a local optimiser using the exact energy and gradient calculations.

As a first step, rather than calculating the energy by summing over all atom pairs, a summation at the residue level was investigated. If this were possible, then for an alanine n-mer, the computational requirements of calculating energy and energy gradients would be reduced by a factor of approximately 100. Clearly the interaction energy between two residues depends both on the distance that they are apart and also their relative orientation. A measure of the relative orientation can be determined by, for example, measuring some number of distances between selected points on the two residues. These distances could then be used as indices into a multi-dimensional energy look-up table. However, even if only just three distances are measured, given the resolution required in the distance measurement, the size of the energy look-up table for residue–residue interaction is of the order of 4 Mega-bytes for each possible pairing of residues. As there are 20 different residues, up to 400 such energy tables may be required (totalling 1.6 Giga-bytes in size). In addition, there is a requirement for gradient tables of similar size. Clearly this approach is not feasible with current technology (while possibly all energy tables could be memory resident, they would certainly not be cache resident).

As a variation on the use of residue–residue energy look-up tables, a method of only using them at distances greater than some lower bound was investigated. As shown in Figure 3 for alanine (the protein used in this study was alanine 58–mer), if the centre of masses of the two residues are separated by more than 12.8 Å then the range of possible interaction energies is less than unity. This suggests that using an energy look-up table, when the distance between the residue centre of masses is greater than 12.8 Å, will only have a minimal effect on the final accuracy of the calculated protein energy. The energy look-up table is indexed only by the distance between the centres of mass of the interacting residues, and contains the average sampled energy for each distance (resolution 0.1 Å). This gives a maximum error of ±0.5 for each residue–residue interaction and, as there is a significant number of randomly orientated residue pairs for which the energy look-up table will be used for during each protein energy calculation, the error in the final protein energy should be minimal.

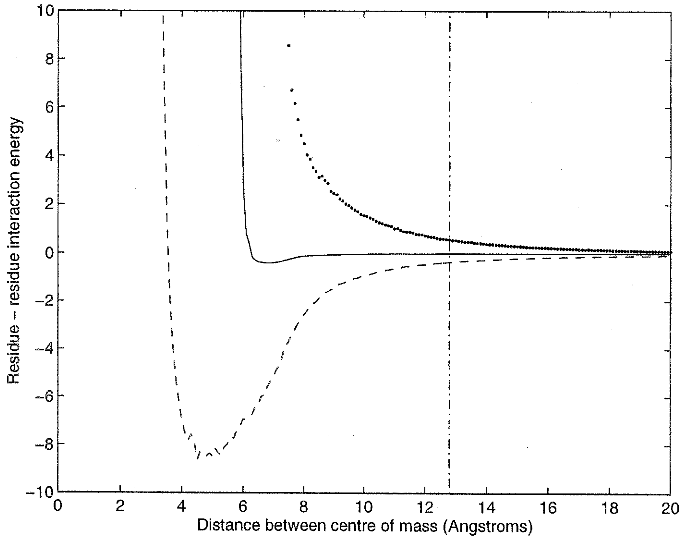

Figure 3: Randomly sampled residue interaction energies as a function of the distance between the residue centres of mass. The curves show the average, maximum and minimum energies found over a large number of random samples of component orientations. The lower bound of the residue energy look-up table is shown by the '-.' line.

The next logical step is to consider each residue as consisting of a number of smaller components and obtain the interaction energy of two residues by summing the interaction energies of these individual components. As for summation at the residue level, there is a distance below which an exact calculation must be performed (because of the range of energies which the relative component orientations can produce). However it is reasonable to expect that this distance will be less than the 12.8 Å of the residue–residue energy look-up table. This is the basis of the TLEC method used in this study and is now described in more detail:

In TLEC the calculation of protein energy is subdivided into two categories:

- **Chain Energy** - the total energy arising from elements that are physically close in the protein chain.
- **Interaction Energy** - the total energy arising from elements that not physically close in the protein chain (but which may be close in space).

The chain energy is identical to that described in (1) above with the basic exception that the Lennard-Jones and coulombic portions are only summed over atom pairs that are within the same or adjacent residues. However, for simplicity, all interactions involving atoms within the first and last residue were also calculated as part of the chain energy.

The interaction energy is calculated for each residue pairing not included in the chain energy calculation as follows:

- If the distance between the centres of mass of the two residues is greater than 12.8 Å then the residue–residue energy look-up table is used.
- If the distance between the centres of mass of the two residues is less than 12.8 Å each residue is considered to consist of three components, the $N + H$, $C_a + R + H$ and $C + O$ groupings (referred to as NH, CRH and CO in the remainder of this paper).
  - If the distance between the centres of mass of the two components is within the lower and upper bounds of the component energy look-up table then it is used. These energy look-up tables, indexed by the distance between the centre of mass of the two components, give the expected potential energy arising from the interaction of all atoms of each component. The expected potential energy contained in the energy tables is obtained by sampling, over a large number of random orientations, the potential energy between components.

- If the distance between the centre of mass of the two components is less than the energy table lower bound then an exact calculation is performed to obtain the interaction energy for the two components.
- If the distance between the centre of mass of the two components is greater than the energy table upper bound then the component interaction is ignored.

In this study, the component energy table lower bounds were defined as that distance where the difference between the maximum and minimum energies obtained during the sampling process exceeds unity (Table 1). From this table it is clear that these energy tables can be used at a distance considerably less than the 12.8 Å cut-off distance for the residue–residue look-up energy table.

For the alanine 58–mer protein, the number of TLEC energy tables required is seven and each energy table is approximately 1.5 Kilo-bytes in size. For a more typical protein, nine energy tables per possible unique residue pairing are required giving a maximum possible size of (400 * 10 * 1.5) Kilo-bytes = 6 Mega-bytes for the energy tables (as compared to the 1.6 Giga-bytes for the multi-dimensional energy table described above).

Interacting components	Energy table lower bound (Å)
$NH - NH$	6.5
$NH - CRH$	6.1
$NH - CO$	8.8
$CRH - CRH$	5.8
$CO - CRH$	8.1
$CO - CO$	8.5

Table 1: Component energy look-up table lower bounds. Each lower bound is the distance between the centre of masses of the components at which the possible variation in interaction energy exceeds unity.

Figure 4 shows the distribution of energy as a function of the distance between the centre of mass for each of the components of alanine.

Gradient tables were obtained by least-squares fitting a polynomial of degree eight to each of the average curves shown in Figure 4. This polynomial was then differentiated and the resulting equation used to generate tables containing the contribution to the energy gradient of each component to component pairing. The lower and upper bounds for the gradient tables were the same as those used for the corresponding energy table.

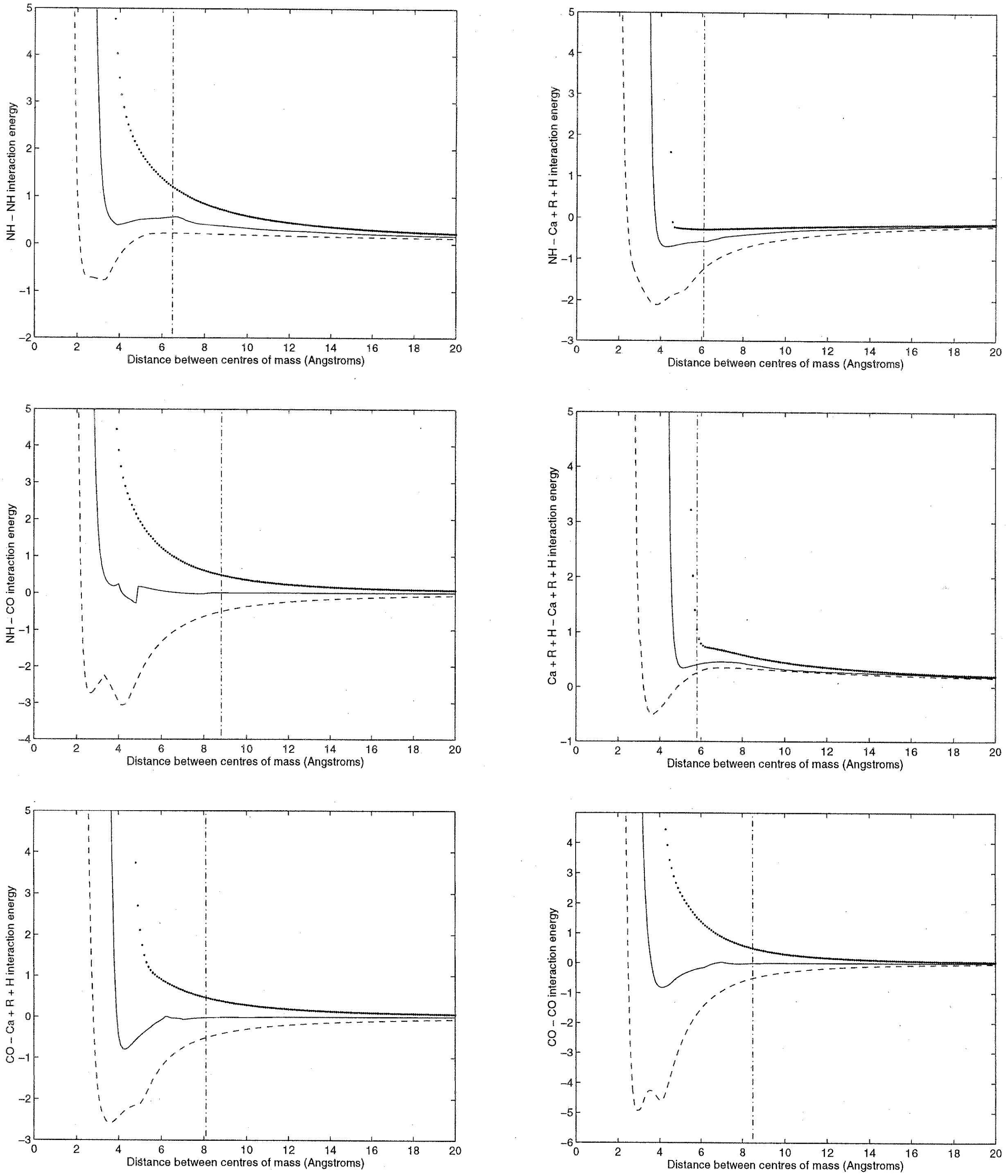

Figure 4: Randomly sampled residue component interaction energies as a function of the distance between the component centres of mass. The curves show the average, maximum and minimum energies found over a large number of random samples of component orientations. In each plot the lower bound of the corresponding energy look-up table is shown by the '-.' line.

4 Computational Results

Using alanine 58−mer as the test case and 1000 sampling runs, results were obtained by comparing CHARMM and TLEC energies and processor times taken for protein energy calculations, gradient calculations and final protein energy obtained using a local optimiser. The local optimiser used was the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm contained in [8]. The variables used to specify the structure of the protein were the dihedral angles (i.e. all bond lengths and bond angles were assumed to remain constant). In both CHARMM and TLEC, calculation of protein energies and energy gradients required calculation of the cartesian coordinates of all atoms and, in the case of the energy gradients, transformation back to gradients with respect to these dihedral angles.

For the protein energy and energy gradient calculations, two types of alanine 58−mer configurations were used. The first was simply a configuration generated by randomly generating dihedral angles in the range $0 \ldots 2\pi$. The second type of configuration used was obtained by taking a randomly generated configuration and performing a local optimisation on it.

4.1 Energy Calculations

The results obtained are summarised in Table 2 where the average processor time required for a CHARMM energy calculation is 5.96 times that of a TLEC energy calculation (for a randomly generated protein). For proteins that were randomly generated and then locally optimised the corresponding ratio is 6.32. In both experiments, less than 10% of component−component interactions needed to be calculated exactly.

Protein	CHARMM		TLEC	
Configuration	Mean	STD	Mean	STD
Random	0.4020	0.0007	0.0674	0.0094
Optimised	0.4017	0.0007	0.0635	0.0040

Table 2: Mean and standard deviation of processor times for CHARMM and TLEC energy calculations. The random protein configurations were as randomly generated while the optimised protein configurations were initially randomly generated and then locally optimised.

As expected there is very little variation in the processor times taken for the CHARMM energy evaluation while there is more variation in the TLEC energy calculation. This is a direct result of the variation in distances between components in different configurations allowing more or less use of the table look-up mechanism.

Figure 5 shows the correlation between the CHARMM and TLEC energies for alanine 58−mer configurations that were initially randomly generated and then locally optimised.

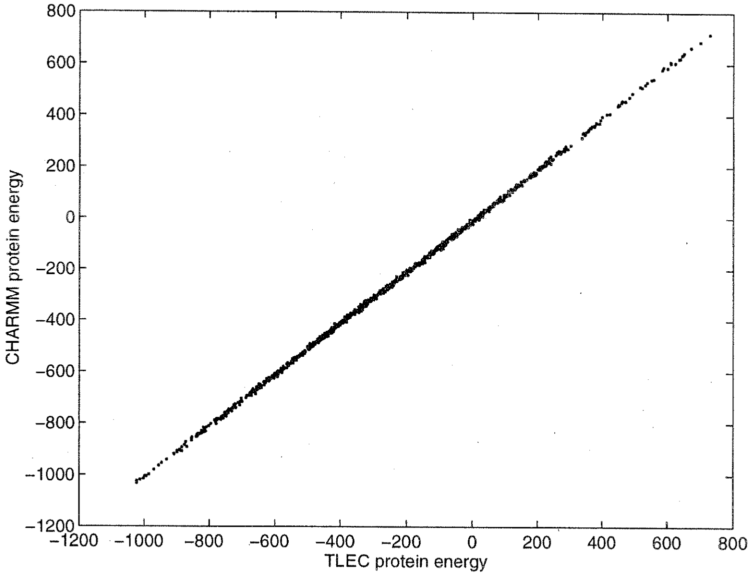

Figure 5: Correlation between TLEC and CHARMM energies for randomly generated then locally optimised alanine 58−mer configurations. The correlation coefficient is 0.9999.

4.2 Gradient Calculations

The results are summarised in Table 3 where the average processor time required for a CHARMM energy gradient calculation is 3.18 times that of a TLEC energy gradient calculation (for a randomly generated protein). For proteins that were randomly generated and then locally optimised the corresponding ratio is 3.14. This ratio is somewhat lower than that obtained for the energy calculations because the additional coordinate transformations involved in the gradient calculation must be performed in both CHARMM and TLEC.

Protein	CHARMM		TLEC	
Configuration	Mean	STD	Mean	STD
Random	0.6737	0.0008	0.2118	0.0139
Optimised	0.6564	0.0010	0.2092	0.0064

Table 3: Mean and standard deviation of processor times for CHARMM and TLEC energy gradient calculations. The random protein configurations were as randomly generated while the optimised protein configurations were initially randomly generated and then locally optimised.

4.3 Local Optimisations

Two experiments were performed to evaluate the use of the BFGS optimiser using CHARMM energies and gradients as compared to TLEC energies and gradients. The first experiment started from a randomly generated alanine 58−mer (almost always with a high energy) while the second was done starting with a very low energy alanine 58−mer which had been modified by randomly changing two consecutive dihedral angles. This second experiment simulates the effect of a random local change to a low energy protein commonly used by global optimisers.

For the first experiment, Figure 6 shows the correlation between the final CHARMM and TLEC BFGS optimised energies. The mean of the TLEC final energies (after recalculating using CHARMM) is -255.10 while the mean of the CHARMM final energies is -263.67. Figure 6 also shows the correlation between the processor times required for CHARMM and TLEC BFGS optimisations. The mean of the TLEC processor time is 66.73 seconds with a standard deviation of 20.22 seconds. For CHARMM, the mean BFGS processor time is 306.23 seconds with a standard deviation of 106.04 seconds. On average a CHARMM BFGS optimisation requires 4.6 times more processor time than that required for a TLEC BFGS optimisation.

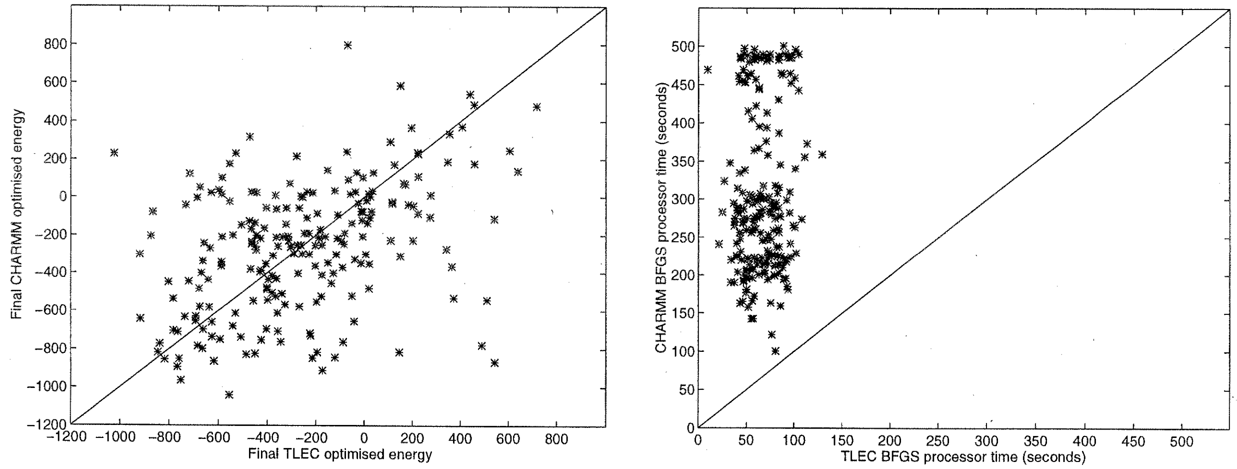

Figure 6: Correlation between TLEC and CHARMM final BFGS optimised energies and processor times starting from a randomly generated alanine 58−mer configuration.

For the second experiment, Figure 7 shows the correlation between the final CHARMM and TLEC BFGS optimised energies. The mean of the TLEC final energies (after recalculating in CHARMM) is -1,500.10 while the mean of the CHARMM final energy is -1,514.40.

Figure 7 shows the correlation between the processor times required for CHARMM and TLEC

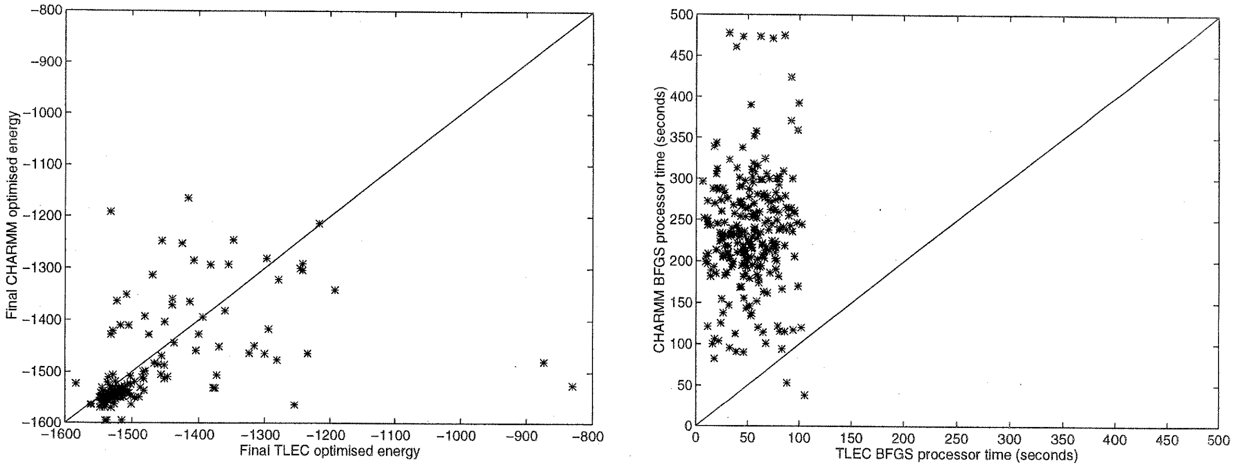

Figure 7: Correlation between TLEC and CHARMM final BFGS optimised energies and processor times starting from an optimal alanine 58−mer configuration with 2 random changes to consecutive dihedral angles.

BFGS optimisations. The mean of the TLEC processor time is 52.27 seconds with a standard deviation of 23.48 seconds. For CHARMM, the mean BFGS processor time was 234.24 seconds with a standard deviation of 70.55 seconds. On average a CHARMM BFGS optimisation requires 4.5 times more processor time than that required for a TLEC BFGS optimisation.

4.4 Effect of Chain Energy Calculations

In the experiments described above, only the interactions between adjacent residues were included in the exact chain energy calculation. Clearly the chain energy calculation can be extended by including all interactions where residues are separated by less than some 'pairing depth' (and removing these calculations from the interaction energy calculation of TLEC).

The following measurements (using alanine 58−mer with 100 sampling runs) were performed to determine the effect of the pairing depth:

- average processor time to calculate the energy and energy gradient for randomly generated configurations (Figure 8).
- average processor time and average final energy for locally optimising randomly generated configurations (Figure 9).
- average processor time and average final energy for locally optimising an optimal configuration for which 2 consecutive dihedral angles had been randomly modified (Figure 10).

As can be seen from these figures, the extra overhead in the calculation of the chain energy is relatively small. The improvement in the accuracy of the TLEC calculation is shown in Table 4 where the average difference in protein energy calculated using both TLEC and CHARMM are tabulated.

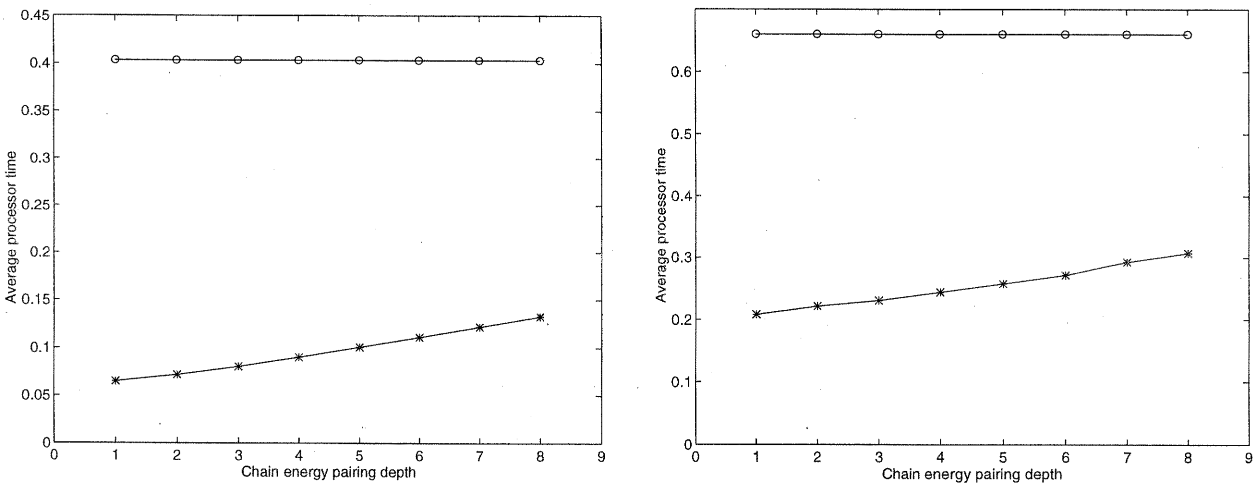

Figure 8: Average processor time to calculate the energy and energy gradient for randomly generated configurations of alanine 58−mer ('*' = TLEC, 'o' = CHARMM).

Chain Energy Pairing Depth	Average Difference in TLEC & CHARMM Energies
1	16.6772
2	12.9308
3	11.7581
4	8.7261
5	6.7235
6	5.4367
7	4.5866
8	3.8850

Table 4: The average difference in protein energy as calculated by both TLEC and CHARMM.

These results suggest that in some cases it may more appropriate to use a chain energy pairing depth greater than unit. In fact early tests show that to obtain the alpha-helix structure for alanine n−mer, a chain pairing depth of four is required.

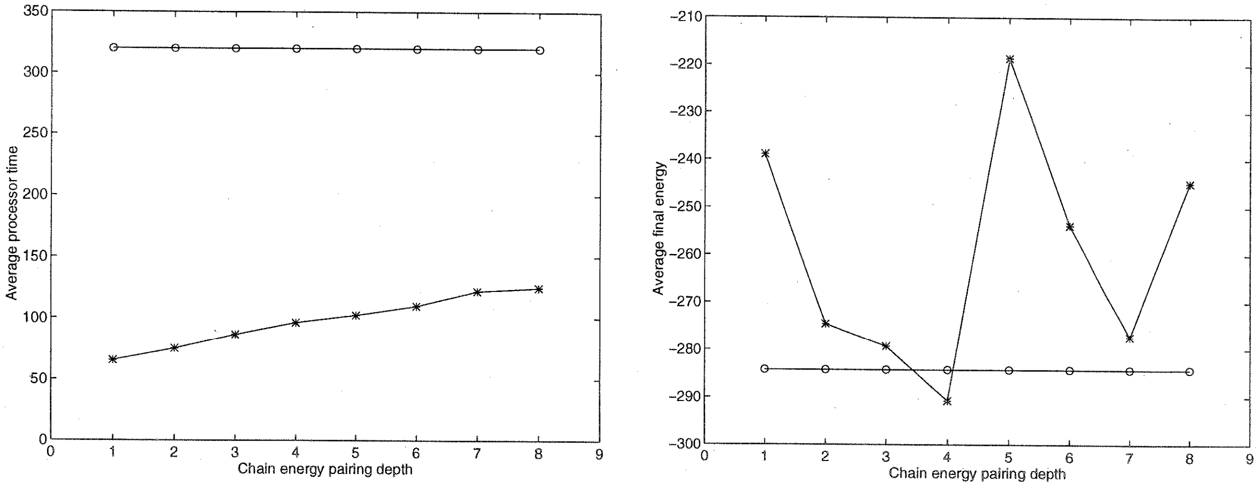

Figure 9: Average processor time and average final energy for locally optimising randomly generated configurations of alanine 58−mer ('*' = TLEC, 'o' = CHARMM).

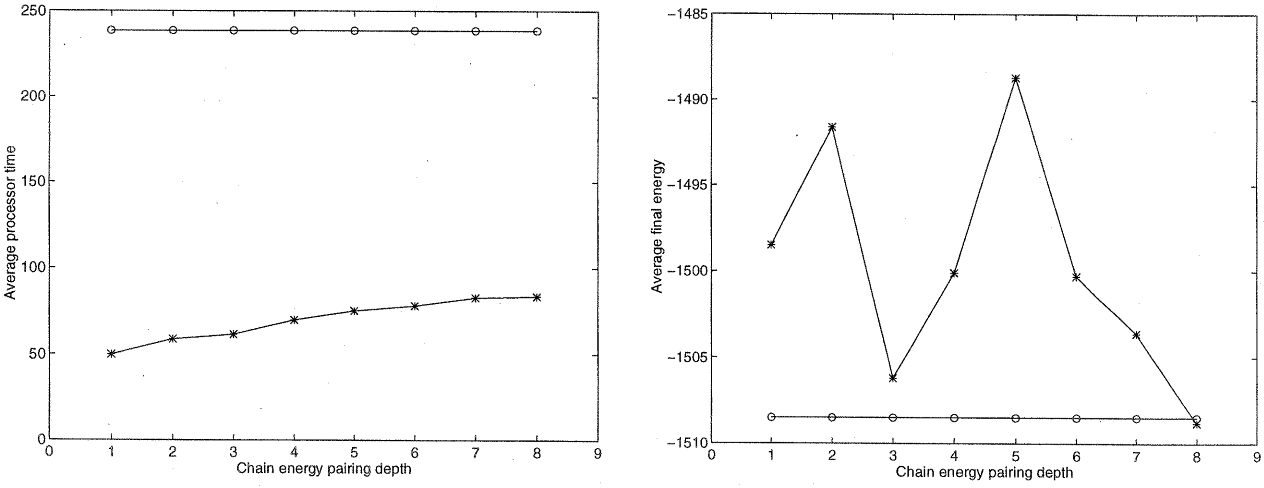

Figure 10: Average processor time and average final energy for locally optimising an optimal configuration of alanine 58−mer for which 2 consecutive dihedral angles had been randomly modified ('*' = TLEC, 'o' = CHARMM).

5 Conclusion

This study demonstrates that it is possible to quickly calculate protein energies and energy gradients with sufficient accuracy for optimisation purposes using a combination of table look-ups and partial exact calculation. The speedup factors are significant and suggest that the method could usefully be employed within a global optimisation method to be obtain low energy protein structures which could subsequently be optimised using a complete exact calculation of protein energy and energy gradient. Work is currently in progress in incorporating TLEC within a genetic algorithm.

References

[1] C.B. Anfinsen, Principles That Govern The Folding of Protein Chains, *Science* 181 223-230 (1973).

[2] F.M. Richards, The Protein Folding Problem, *Scientific American*, January, 54-63 (1991).

[3] S. Sun, Reduced Representation Approach to Protein Tertiary Structure Prediction: Statistical Potential and Simulated Annealing, *J. Theor. Biol.* 172 13-32 (1995).

[4] P.M. Pardalos, D. Shalloway, G. Xue, Optimisation Methods for Computing Global Minima of Nonconvex Potential Energy Functions, *Journal of Global Optimization* 4 2 117-133 (1994).

[5] Bernard R. Brooks, Robert E. Bruccoleri, Barry D. Olafson, David J. States, S. Swaminathan, Martin Karplus, CHARMM: A Program for Macromolecular Energy, Minimization and Dynamics Calculations, *Journal of Computational Chemistry*, 4 187-217 (1983).

[6] M. Levitt, A Simplified Representation of Protein Conformation for Rapid Simulation of Protein Folding, *Journal of Molecular Biology*, 104 59-107 (1976).

[7] K.A. Dill, Dominant forces in protein folding, *Biochemistry* 29 7133-7155 (1990).

[8] William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery, *Numerical Recipes in C*, Cambridge University Press (1992).